Chi Calculus with Mismatch*

Yuxi Fu** Zhenrong Yang

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

Abstract. The theory of chi processes with the mismatch operator is studied. Two open congruence relations are investigated. These are weak early open congruence and weak late open congruence. Complete systems are given for both congruence relations. These systems use some new tau laws. The results of this paper correct some popular mistakes in literature.

1 Introduction

In recent years several publications have focused on a class of new calculi of mobile processes. These models include χ-calculus ([1–4]), update calculus ([7]) and fusion calculus ([8, 10]). In a uniform terminology they are respectively χ-calculus, asymmetric χ-calculus and polyadic χ-calculus. The χ-calculus has its motivations from proof theory. In process algebraic model of classical proofs there has been no application of mismatch operator. The χ-calculus studied so far contains no mismatch operator. On the other hand the update and fusion calculi have their motivations 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 have the mismatch combinator. Strong bisimulation congruence has been investigated for each of the three models. It is basically the open congruence. A fundamental difference between χ-like calculi and π-like calculi ([5]) is that all names in the former are subject to update whereas local 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 [1] 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 localization. This is the property that led Fu to introduce L-congruences ([2]). These congruence relations form a lattice under inclusion order. It has been demonstrated

* *CONCUR 2000*, Lecture Notes in Computer Science 1877, 596-610, 2000.

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

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 asymmetric χ-calculus ([4]). Complete systems have been discovered for L-congruences on both finite χ-processes and finite asymmetric χ-processes ([4]). 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 ([8]). 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 a third party can detect a difference between the two processes. Usually the third party is also a process. Now to observe a process is to communicate with it. In process calculi the communication happens if the observer and the observee are composed via a parallel operator. It follows that process equivalences must be closed under 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 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 $(x)ax.[x{\neq}y]\tau.P+(x)ax.P \xrightarrow{a(x)} P$ can be simulated by $(x)ax.[x{\neq}y]\tau.P \xrightarrow{a(x)}\xrightarrow{\tau} P$. However

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

for $\overline{a}y|((x)ax.[x{\neq}y]\tau.P+(x)ax.P) \xrightarrow{\tau} \mathbf{0}|P[y/x]$ can not be matched up by any transitions from $\overline{a}y|(x)ax.[x{\neq}y]\tau.P$. For similar reason

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

but

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

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

In the above counter examples the mismatch operator plays a crucial part. 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 problem caused by the mismatch combinator is mostly operational. A well known fact is that transitions are not stable under name instantiations, which render 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 early and late semantics.

The early/late dichotomy is well known in the semantic theory of π-calculus. 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. This has led them to realize the problem with the weak hyperequivalence.

In this paper we study early and late open congruences for χ-calculus with the mismatch operator. The main contributions of this paper are as follows:

– We point out that there is an early/late discrepancy in the open semantics of calculi of mobile processes with the mismatch combinator.
– We provide a correct treatment of open semantics for χ-calculus with mismatch. Our definitions of early and late congruences suggest immediately how to generalize them to fusion calculus.
– We propose two new tau laws to handle free prefix operator and one to deal with update prefix combinator. These tau laws rectify the mistaken tau laws in [8]. We point out that the new tau laws for free prefix operator subsume the corresponding laws for bound prefix operator.
– We give complete axiomatic systems for both early open congruence and late open congruence. These are the first completeness result for weak congruences on χ-like processes with the mismatch operator.

The structure of the paper is as follows: Section 2 summarizes some background material on χ-calculus. Section 3 defines two weak open congruences. Section 4 proves completeness theorem. Some comments are made in the final section.

2 The 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 restricted names. The input prefix operator $a(x)$ introduces the dummy name x to be instantiated by an action induced by the prefix operator. On the other hand the localization operator (y) forces the name

y to be local, which will never be instantiated. Semantically these two restricted names are very different. The χ-calculus can be seen as obtained from the π-calculus by unifying the two classes of restricted names. 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 $\mathcal{C}$ for the set of $\chi^{\neq}$-processes defined by the following grammar:

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

where $\alpha \in \mathcal{N} \cup \overline{\mathcal{N}}$. Here $\mathcal{N}$ is the set of names ranged over by small case letters. The set $\{\overline{x} \mid x \in \mathcal{N}\}$ of conames is denoted by $\overline{\mathcal{N}}$. 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 local. A name is global in P if it is not local in P. The global names, the local names and the names of P, as well as the notations $gn(P)$, $ln(P)$ and $n(P)$, are used in their standard meanings. In sequel we will use the functions $gn(\_)$, $ln(\_)$ and $n(\_)$ without explanation. We adopt the α-convention widely used in the literature on process algebra.

The following labeled transition system defines the operational semantics:

Sequentialization

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

Composition

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

Communication

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

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

Localization

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

Condition

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

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 ν ranges over the set $\{\alpha(x), \alpha[x] \mid \alpha \in \mathcal{N} \cup \overline{\mathcal{N}}, x \in \mathcal{N}\} \cup \{\tau\}$ of actions and the letter λ over the set $\{\alpha(x), \alpha[x], [y/x] \mid \alpha \in \mathcal{N} \cup \overline{\mathcal{N}}, x, y \in \mathcal{N}\} \cup \{\tau\}$ of labels. The symbols $\alpha(x), \alpha[x], [y/x]$ represent restricted action, free action and update action respectively. The x in $\alpha(x)$ is local.

A substitution is a function from $\mathcal{N}$ to $\mathcal{N}$. 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 $P\sigma$. Suppose Y is a finite set $\{y_1, \ldots, y_n\}$ of names. The notation $[y \notin Y]P$ stands 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 to stand for sequences of match and mismatch combinators concatenated one after another, μ for a sequence of match operators, and δ for a sequence of mismatch operators. Consequently we write ψP, μP and δP. When the length of ψ (μ, δ) is zero, ψP (μP, δP) is just P. The notation $\phi \Rightarrow \psi$ says that ϕ logically implies ψ and $\phi \Leftrightarrow \psi$ that ϕ and ψ are logically equivalent. A substitution σ agrees with ψ, and ψ agrees with σ, when $\psi \Rightarrow x{=}y$ if and only if $\sigma(x){=}\sigma(y)$. The notations $\Longrightarrow$ and $\stackrel{\hat{\lambda}}{\Longrightarrow}$ are used in their standard meanings. A sequence $x_1, \ldots, x_n$ of names will be abbreviated to $\tilde{x}$.

The following lemma is useful in later proofs.

Lemma 1. *If $P\sigma \xrightarrow{\lambda} P_1$ then there exists some P' such that $P_1 \equiv P'\sigma$.*

Notice that a substitution σ may disable an action of P. So one can not conclude $P\sigma \xrightarrow{\lambda\sigma} P'\sigma$ from $P \xrightarrow{\lambda} P'$. But the above lemma tells us that we may write $P\sigma \xrightarrow{\lambda} P'\sigma$ once we know that $P\sigma$ can induce a λ transition.

3 Open Bisimilarities

In our view the most natural equivalence for mobile processes is the open equivalence introduced by Sangiorgi ([9]). Technically the open equivalence is a bisimulation equivalence closed under substitution of names. Philosophically the open approach assumes that the environments are dynamic in the sense that they are shrinking, expanding and changing all the time. After the simulation of 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 simple minded definition of weak open bisimulation 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 $Q \stackrel{\widehat{\lambda}}{\Longrightarrow} Q'\mathcal{R}P'$ for some Q'.

As it turns out this is a bad definition 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. To correct the above definition, one should adopt the approach that name instantiations should take place in the earliest possible occasion. This brings us to the familiar early and late frameworks.

Definition 2. *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 $P \stackrel{\tau}{\longrightarrow} P'$ then Q' exists such that $Q \Longrightarrow Q'\mathcal{R}P'$.
(ii) If $P \stackrel{[y/x]}{\longrightarrow} P'$ then Q' exists such that $Q \stackrel{[y/x]}{\Longrightarrow} Q'\mathcal{R}P'$.
(iii) If $P \stackrel{\alpha[x]}{\longrightarrow} P'$ then for every y some Q', Q'' exist such that $Q \Longrightarrow\stackrel{\alpha[x]}{\longrightarrow} Q''$ and $Q''[y/x] \Longrightarrow Q'\mathcal{R}P'[y/x]$.
(iv) If $P \stackrel{\alpha(x)}{\longrightarrow} P'$ then for every y some Q', Q'' exist such that $Q \Longrightarrow\stackrel{\alpha(x)}{\longrightarrow} Q''$ and $Q''[y/x] \Longrightarrow Q'\mathcal{R}P'[y/x]$.
The early open bisimilarity $\approx_o^e$ is the largest early open bisimulation.

The clause (iv) is easy to understand. Its counterpart for weak bisimilarity of π-calculus is familiar. The clause (iii) calls for some explanation. In $\chi^{\neq}$-calculus free actions can also incur name updates in suitable contexts. Suppose $P \stackrel{\alpha[x]}{\longrightarrow} P''$. Then $(x)(P|\overline{\alpha}[y].Q) \stackrel{\tau}{\longrightarrow} P''[y/x]|Q[y/x]$. Even if $P'' \Longrightarrow P'$, one does not necessarily have $P''[y/x] \Longrightarrow P'[y/x]$. Had we replace clause (iii) by

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

then we would have obtained a relation to which the second counter example in the introduction applies. 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 3. *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 $P \stackrel{\tau}{\longrightarrow} P'$ then Q' exists such that $Q \Longrightarrow Q'\mathcal{R}P'$.
(i) If $P \stackrel{[y/x]}{\longrightarrow} P'$ then Q' exists such that $Q \stackrel{[y/x]}{\Longrightarrow} Q'\mathcal{R}P'$.
(iii) If $P \stackrel{\alpha[x]}{\longrightarrow} P'$ then Q'' exists such that $Q \Longrightarrow\stackrel{\alpha[x]}{\longrightarrow} Q''$ and for every y some Q' exists such that $Q''[y/x] \Longrightarrow Q'\mathcal{R}P'[y/x]$.
(iv) If $P \stackrel{\alpha(x)}{\longrightarrow} P'$ then Q'' exists such that $Q \Longrightarrow\stackrel{\alpha(x)}{\longrightarrow} Q''$ and for every y some Q' exists such that $Q''[y/x] \Longrightarrow Q'\mathcal{R}P'[y/x]$.
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: $a[x].[x{=}y]\tau.P{+}a[x].[x{\neq}y]\tau.P \approx_o^e a[x].[x{=}y]\tau.P{+}a[x].[x{\neq}y]\tau.P{+}a[x].P$ but not $a[x].[x{=}y]\tau.P{+}a[x].[x{\neq}y]\tau.P \approx_o^l a[x].[x{=}y]\tau.P{+}a[x].[x{\neq}y]\tau.P{+}a[x].P$.

The lesson we have learned is that we should always check if an observational equivalence is closed under parallel composition. The next lemma makes sure that this is indeed true for the two open bisimilarities.

Lemma 4. *Both $\approx_o^e$ and $\approx_o^l$ are closed under localization and composition.*

Both open bisimilarities are also closed under the prefix and match combinators. But neither is closed under the summation operator or the mismatch operator. For instance $[x{\neq}y]P \approx_o^e [x{\neq}y]\tau.P$ does not hold in general. To obtain the largest congruence contained in early (late) open bisimilarity we follow the standard approach: We say that 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 a tau action of $P\sigma$ must be matched up by a non-empty sequence of tau actions from $Q\sigma$ and vice versa. Clearly $\simeq_o^e$ is a congruence. Similarly we can define $\simeq_o^l$.

4 Axiomatic System

In [2] completeness theorems are proved for L-bisimilarities on χ-processes without 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 [2] should be modified. The modification is done by incorporating ideas from [6]. In this section, we give the complete axiomatic systems for early and late open congruences using the modified approach. First we need to define two induced prefix operators, tau and update prefixes, as follows:

$$[y|x].P \stackrel{\text{def}}{=} (a)(\overline{a}[y]|a[x].P)$$
$$\tau.P \stackrel{\text{def}}{=} (b)[b|b].P$$

where a, b are fresh. The following are some further auxiliary definitions.

Definition 5. *Let V be a finite set of names. We say that ψ is complete on V if $n(\psi) \subseteq V$ and for each pair x, y of names in V it holds that either $\psi \Rightarrow x{=}y$ or $\psi \Rightarrow x \neq y$.*

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

Definition 7. *A substitution σ is induced by ψ if it agrees with ψ and $\sigma\sigma = \sigma$.*

Let AS denote the system consisting of the rules and laws in Appendix A plus the following expansion law:

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

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

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

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

We write $AS \vdash P = Q$ to indicate that the equality $P = Q$ can be inferred from AS. Some important derived laws of AS are given in Appendix A.

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

Definition 8. *A process P is in normal form on $V \supseteq fn(P)$ if P is of the form $\sum_{i\in I_1} \phi_i\alpha_i[x_i].P_i + \sum_{i\in I_2} \phi_i(x)\alpha_i[x].P_i + \sum_{i\in I_3} \phi_i[z_i|y_i].P_i$ such that x does not appear in P, ϕ_i is complete on V for each $i \in I_1 \cup I_2 \cup I_3$, P_i is in normal form on V for $i \in I_1 \cup I_3$ and is in normal form on $V \cup \{x\}$ for $i \in I_2$. Here I_1, I_2 and I_3 are pairwise disjoint finite indexing sets.*

Notice that if P is in normal form and σ is a substitution then $P\sigma$ is in normal form.

The depth of a process measures the maximal length of nested prefixes in the process. The structural definition goes as follows: (i) $d(\mathbf{0}) = 0$; (ii) $d(\alpha[x].P) = 1{+}d(P)$; (iii) $d(P|Q) = d(P){+}d(Q)$; (iv) $d((x)P) = d(P)$; (v) $d([x{=}y]P) = d(P)$; (vi) $d(P{+}Q) = max\{d(P), d(Q)\}$.

Lemma 9. *For a process P and a finite set V of names such that $fn(P) \subseteq V$ there is a normal form Q on V such that $d(Q) \leq d(P)$ and $AS \vdash Q = P$.*

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 of the two weak open congruences. The tau laws used in this paper are given in Figure 1. Some derived tau laws are listed in Figure 2. In what follows, we will write AS_o^l for $AS \cup \{T1, T2, T3a, T3b, T3d, T4\}$ and AS_o^e for $AS \cup \{T1, T2, T3a, T3c, T3d, T4\}$.

The next lemma discusses some relationship among the tau laws.

Lemma 10. *(i) $AS_o^e \vdash TD5$. (ii) $AS_o^l \vdash TD6$. (iii) $AS \cup \{T3c\} \vdash T3b$.*

Proof. (i) By T3c and C2, we get:

$$\begin{aligned} AS_o^e \vdash \Sigma(a(x), P, Q, \delta) &= (x)(\Sigma(a[x], P, Q, \delta){+}[x{\notin}n(\delta)]\delta a[x].Q) \\ &= \Sigma(a(x), P, Q, \delta){+}(x)[x{\notin}n(\delta)]\delta a[x].Q \\ &\overset{LD3}{=} \Sigma(a(x), P, Q, \delta){+}(x)\delta a[x].Q \\ &\overset{LD2}{=} \Sigma(a(x), P, Q, \delta){+}\delta(x)a[x].Q \end{aligned}$$

The proofs of (ii) and (iii) are omitted. □

T1	$\alpha[x].\tau.P = \alpha[x].P$		
T2	$P+\tau.P = \tau.P$		
T3a	$\alpha[x].(P+\tau.Q) = \alpha[x].(P+\tau.Q)+\alpha[x].Q$		
T3b	$\alpha[x].(P+\delta\tau.Q) = \alpha[x].(P+\delta\tau.Q)+[x\notin n(\delta)]\delta\alpha[x].Q$	$x \notin n(\delta)$	
T3c	$\Sigma(\alpha[x], P, Q, \delta) = \Sigma(\alpha[x], P, Q, \delta)+[x\notin n(\delta)]\delta\alpha[x].Q$	$x \notin n(\delta)$	
T3d	$[y\vert x].(P+\delta\tau.Q) = [y\vert x].(P+\delta\tau.Q)+\psi\delta[y\vert x].Q$		
T4	$\tau.P = \tau.(P+\psi\tau.P)$		

In T3d, 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 $\psi \Rightarrow [x\neq u][x\neq v][y\neq u][y\neq v]$.
In T3c, $\Sigma(\alpha[x], P, Q, \delta)$ is $\sum_{y\in Y} \alpha[x].(P_y+\delta[x=y]\tau.Q)+\alpha[x].(P+\delta[x\notin Y]\tau.Q)$.

Fig. 1. The Tau Laws

TD1	$[x\vert y].\tau.P = [x\vert y].P$	
TD2	$\tau.\tau.P = \tau.P$	
TD3	$[x\vert y].(P+\tau.Q) = [x\vert y].(P+\tau.Q)+[x\vert y].Q$	
TD4	$\tau.(P+\tau.Q) = \tau.(P+\tau.Q)+\tau.Q$	
TD5	$(x)\alpha[x].(P+\delta\tau.Q) = (x)\alpha[x].(P+\delta\tau.Q)+\delta(x)\alpha[x].Q$	$x \notin n(\delta)$
TD6	$\Sigma(\alpha(x), P, Q, \delta) = \Sigma(\alpha(x), P, Q, \delta)+\delta(x)\alpha[x].Q$	$x \notin n(\delta)$

In TD6, $\Sigma(\alpha(x), P, Q, \delta)$ is $\sum_{y\in Y}(x)\alpha[x].(P_y+\delta[x=y]\tau.Q)+(x)\alpha[x].(P+\delta[x\notin Y]\tau.Q)$.

Fig. 2. The Derived Tau Laws

To establish the completeness theorem, some properties of AS and the open bisimilarities must be established first. The next three lemmas describe these properties.

Lemma 11. *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 $Q\sigma \stackrel{\tau}{\Longrightarrow} Q'$ then $AS \cup \{T1, T2, T3a\} \vdash Q = Q+\phi\tau.Q'$.
(ii) If $Q\sigma \Longrightarrow\xrightarrow{\alpha[x]} Q'$ then $AS \cup \{T1, T2, T3a\} \vdash Q = Q+\phi\alpha[x].Q'$.
(iii) If $Q\sigma \Longrightarrow\xrightarrow{\alpha(x)} Q'$ then $AS \cup \{T1, T2, T3a\} \vdash Q = Q+\phi(x)\alpha[x].Q'$.
(iv) If $Q\sigma \stackrel{[y/x]}{\Longrightarrow} Q'$ then $AS \cup \{T1, T2, T3a, T3d\} \vdash Q = Q+\phi[y|x].Q'$.

Proof. (iv) If $Q\sigma \xrightarrow{[y/x]} Q'$ then $AS \cup \{T1, T2, T3a, T3d\} \vdash Q = Q+\phi[y|x].Q'$. Suppose for example $Q\sigma \stackrel{\tau}{\Longrightarrow} Q_1\sigma \xrightarrow{[y/x]} Q_2 \stackrel{\tau}{\Longrightarrow} Q'$. It is easy to see that $Q_2 \equiv Q_2'\sigma[y/x]$. Let ψ be a complete condition on $fn(Q_2')$ that induces $\sigma[y/x]$. Suppose $\psi \Leftrightarrow \mu\delta$. Clearly $\mu\sigma[y/x]$ is true. Therefore

$$\begin{aligned}
AS \cup \{T1, T2, T3a, T3d\} \vdash Q &= Q+\phi[y|x].Q_2 \\
&= Q+\phi[y|x].Q_2'\sigma[y/x] \\
&= Q+\phi[y|x].Q_2' \\
&= Q+\phi[y|x].(Q_2'+\psi\tau.Q')
\end{aligned}$$

$$
\begin{aligned}
&= Q+\phi[y|x].(Q_2'+\mu\delta\tau.Q') \\
&= Q+\phi[y|x].(Q_2'+\delta\tau.Q') \\
&= Q+\phi([y|x].(Q_2'+\delta\tau.Q')+\theta\delta\tau.Q') \\
&= Q+\phi\theta\delta[y|x].Q'
\end{aligned}
$$

in which θ is defined as in $T3d$. First of all it is easy to see that $\phi \Rightarrow \delta$ for ϕ is complete on V. So $\phi\delta \Leftrightarrow \phi$. Second we need to explain how to construct θ. It should be constructed in such a way that $\phi\theta \Leftrightarrow \phi$. Suppose $\delta \Rightarrow [u{\neq}v]$. There are five possibilities:

- $\phi \Rightarrow x{=}u$. Then we let θ contain $[x{=}u][y{\neq}v]$. If $\phi \Rightarrow y{=}v$ then σ is induced by $[x{=}u][y{=}v]$. It follows that $\sigma[y/x]$ is induced by $[u{=}v]$, which is impossible. Hence $\phi \Rightarrow y{\neq}v$.
- $\phi \Rightarrow y{=}u$ or $\phi \Rightarrow x{=}v$ or $\phi \Rightarrow y{=}v$. These cases are similar to previous case.
- $\phi \Rightarrow [x{\neq}u][x{\neq}v][y{\neq}u][y{\neq}v]$. Simply let θ contain $[x{\neq}u][x{\neq}v][y{\neq}u][y{\neq}v]$.

It is clear from the construction that $\phi \Rightarrow \theta$. Therefore $AS{\cup}\{T1,T2,T3a,T3d\} \vdash Q = Q+\phi\theta\delta[y|x].Q' = Q+\phi[y|x].Q'$. □

Lemma 12. *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*

$$
\begin{aligned}
&Q\sigma \Longrightarrow \xrightarrow{\alpha(x)} Q_1'\sigma,\ Q_1'\sigma[y_1/x] \Longrightarrow Q_1, \\
&Q\sigma \Longrightarrow \xrightarrow{\alpha(x)} Q_2'\sigma,\ Q_2'\sigma[y_2/x] \Longrightarrow Q_2, \\
&\vdots \\
&Q\sigma \Longrightarrow \xrightarrow{\alpha(x)} Q_k'\sigma,\ Q_k'\sigma[y_k/x] \Longrightarrow Q_k, \\
&Q\sigma \Longrightarrow \xrightarrow{\alpha(x)} Q_{k+1}'\sigma \Longrightarrow Q_{k+1}
\end{aligned}
$$

then the following properties hold:

1. $AS \cup \{T1,T2,T3a\} \vdash Q = Q + \psi\sum_{j=1}^{k}(x)\alpha[x].(\tau.Q_j' + \psi[x{=}y_j]\tau.Q_j) + \psi(x)\alpha[x].(\tau.Q_{k+1}' + \psi[x{\notin}V]\tau.Q_{k+1}))$.
2. *If* $Q_1' \equiv Q',\dots,Q_{k+1}' \equiv Q'$, *then* $Q + \psi(x)\alpha[x].(\tau.Q' + \psi\sum_{j=1}^{k}[x{=}y_j]\tau.Q_j + \phi[x{\notin}V]\tau.Q_{k+1})$ *is provably equal to* Q *in* $AS \cup \{T1,T2,T3a\}$.

The proof of the above lemma is similar to that of Lemma 11.

Lemma 13. *In $\chi^{\neq}$-calculus the following properties hold:*
(i) If $P \approx_o^e Q$ then $AS_o^e \vdash \tau.P = \tau.Q$.
(ii) If $P \approx_o^l Q$ then $AS_o^l \vdash \tau.P = \tau.Q$.

Proof. By Lemma 9 we may assume that P and Q are in normal form on $V = fn(P|Q) = \{y_1,y_2,\dots,y_k\}$. Let P be

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

and Q be

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

We prove this lemma by induction on the depth of $P|Q$.

(i) Suppose $\phi_i\pi_i.P_i$ is a summand of P and σ is induced by ϕ_i. There are several cases:

- $\pi_i\sigma$ is an update action $[y|x]$. It follows from $P\approx_o^e Q$ that $Q\sigma \stackrel{[y/x]}{\Longrightarrow} Q'\approx_o^e P_i[y/x]\sigma$. By induction we have $AS_o^e\vdash Q'=P_i\sigma[y/x]$. By (iv) of Lemma 11

$$\begin{aligned}AS_o^e\vdash Q &= Q+\phi_i[y|x].Q'\\ &= Q+\phi_i[y|x].P_i[y/x]\sigma\\ &= Q+\phi_i[y|x].P_i\sigma\\ &= Q+\phi_i\pi_i\sigma.P_i\sigma\\ &= Q+\phi_i\pi_i.P_i\end{aligned}$$

- $\pi_i\sigma$ is a restricted action $\alpha(x)$. Since $P\approx_o^e Q$ one has the following cases:
  - For each $l\in\{1,\dots,k\}$, Q'_{i_l} and Q_{i_l} exist such that $Q\sigma\Longrightarrow\stackrel{\alpha(x)}{\longrightarrow}Q'_{i_l}\sigma$ and $Q'_{i_l}\sigma[y_l/x]\Longrightarrow Q_{i_l}\approx_o^e P_i\sigma[y_l/x]$.
  - $Q'_{i_{k+1}}$ and $Q_{i_{k+1}}$ exist such that $Q\sigma\Longrightarrow\stackrel{\alpha(x)}{\longrightarrow}Q'_{i_{k+1}}\sigma\Longrightarrow Q_{i_{k+1}}\approx_o^e P_i\sigma$.

  By Lemma 12

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

- $\pi_i\sigma$ is a free action $\alpha[x]$. Using the fact $P\approx_o^e Q$ one has the following cases:
  - For each $l\in\{1,\dots,k\}$, Q'_{i_l} and Q_{i_l} exist such that $Q\sigma\Longrightarrow\stackrel{\alpha[x]}{\longrightarrow}Q'_{i_l}\sigma$ and $Q'_{i_l}\sigma[y_l/x]\Longrightarrow Q_{i_l}\approx_o^e P_i\sigma[y_l/x]$.
  - $Q'_{i_{k+1}}$ and $Q_{i_{k+1}}$ exist such that $Q\sigma\Longrightarrow\stackrel{\alpha[x]}{\longrightarrow}Q'_{i_{k+1}}\sigma\Longrightarrow Q_{i_{k+1}}\approx_o^e P_i\sigma$.

  Since ϕ_i is complete on V, it groups the elements of V into several disjoint classes. Assume that these classes are $[x],[a_1],\dots,[a_r]$. Let $\phi_i^{=}$ be the sequence of match operators induced by the equivalence classes $[a_1],\dots,[a_r]$. Let $\phi_i^{=x}$ be the sequence of match operators induced by the equivalence class $[x]$. Let $\phi_i^{\neq}$ be the sequence of mismatch combinators constructed as follows: For $1 \leq p,q \leq r$ and $a \in [a_p], b \in [a_q]$, $a{\neq}b$ is in $\phi_i^{\neq}$. And let $\phi_i^{\neq x}$ be the sequence of mismatch combinators constructed as follows: For $a \in [a_1] \cup \ldots \cup [a_r]$, $a{\neq}x$ is in $\phi_i^{\neq x}$. It is clear that $\phi_i \Leftrightarrow \phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_i^{\neq x}$. Now V can be divided into two subsets: $V^{=x} \stackrel{\text{def}}{=} \{y \mid y \in V, \phi_i \Rightarrow y{=}x\}$; and $V^{\neq x} \stackrel{\text{def}}{=} \{y \mid y \in V, \phi_i \Rightarrow y{\neq}x\} = [a_1] \cup \ldots \cup [a_r]$. Clearly $\phi_i^{\neq x} \Leftrightarrow [x{\notin}V^{\neq x}]$.

- If $y_l \in V^{\neq x}$ then we define $\phi_{i\backslash[y_l]}^{\neq x}$ as follows: For $a \in ([a_1] \cup \ldots \cup [a_r]) \setminus [y_l]$, $a{\neq}x$ is in $\phi_{i\backslash[y_l]}^{\neq x}$. It is easy to see that $\phi_{i\backslash[y_l]}^{\neq x} \Leftrightarrow [x{\notin}(V^{\neq x} \setminus [y_l])]$. Now $\phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_{i\backslash[y_l]}^{\neq x}[x{=}y_l]$ is complete on V and induces $\sigma[y_l/x]$. By Lemma 11

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

- It is clear that $\phi_i^{=}\phi_i^{=x}\phi_i^{\neq}\phi_i^{\neq x}$ is complete on V and induces σ. One has by Lemma 11 that

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

Now

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

– $\pi_i\sigma$ is a tau action. If the tau action is matched by $Q\sigma \stackrel{\tau}{\Longrightarrow} Q'$ 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 $AS^e_o \vdash P{+}Q = Q{+}\Sigma_{i\in I'}\phi_i\tau.Q$ for some $I' \subseteq I$. So by T4 we get $AS^e_o \vdash \tau.(P+Q) = \tau.(Q+\Sigma_{i\in I'}\phi_i\tau.Q) = \tau.Q$. Symmetrically, we can prove $AS^e_o \vdash \tau.(P{+}Q) = \tau.P$. Hence $AS^e_o \vdash \tau.P = \tau.Q$.

(ii) The proof is similar to that for $\approx^e_o$. We consider only one case:

- $\pi_i\sigma$ is a restricted action $\alpha(x)$. It follows from $P \approx^l_o Q$ that some Q' exists such that the following holds:
  - For each $l \in \{1,\ldots,k\}$, Q', Q_{i_l} exists such that $Q\sigma \Longrightarrow \xrightarrow{\alpha(x)} Q'\sigma$ and $Q'\sigma[y_l/x] \Longrightarrow Q_{i_l} \approx^l_o P_i\sigma[y_l/x]$.
  - $Q_{i_{k+1}}$ exists such that $Q\sigma \Longrightarrow \xrightarrow{\alpha(x)} Q'\sigma \Longrightarrow Q_{i_{k+1}} \approx^l_o P_i\sigma$.

  By (ii) of Lemma 12 we get

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

Then by a similar argument as in (i), we get $AS^l_o \vdash \tau.P = \tau.Q$. □

Theorem 14. *In $\chi^{\neq}$-calculus the following completeness results hold:*
(i) If $P \simeq^e_o Q$ then $AS^e_o \vdash P = Q$.
(ii) If $P \simeq^l_o Q$ then $AS^l_o \vdash P = Q$.

Proof. By Lemma 11 and Lemma 13, one can prove the theorem in very much the same way as the proof of Lemma 13 is done. □

5 Historical Remark

The first author of this paper has been working on χ-calculus for some years. His attention had always been on the version of χ without 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 for the π-calculus with the mismatch combinator. Their investigation showed that the simple-minded definition of open bisimilarity is not closed under parallel composition. It is then a small step to realize the problem of the weak hyperequivalence.

It has come a long way to settle down on the axiom T4. The first solution, proposed by the first author in an early version of [4], is the following rule:

$$
\frac{P{+}\Sigma_{i\in I}\psi_i\tau.P = Q{+}\Sigma_{j\in J}\psi_j\tau.Q}{\tau.P = \tau.Q}
$$

The premises of the rule is an equational formalization of $P \approx Q$. In the final version of [4] he observed that the rule is equivalent to the following law:

$$\tau.P = \tau.(P + \Sigma_{i\in I}\psi_i\tau.P)$$

Later on he realized that the above equality can be simplified to T4.

In [8] two tau laws are proposed for fusion calculus. Using the notations of [8] they can be written as follows:

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

$$\iota.(P + \tilde{M}\rho.Q) = \iota.(P + \tilde{M}\rho.Q) + \tilde{M}\iota \wedge \rho.Q \text{ if } \forall x, y.(\tilde{M} \Rightarrow x \neq y) \Rightarrow \neg(x\iota y) \quad (2)$$

where α is a communication action, ι and ρ are fusion actions, and $\tilde{M}$ is a sequence of match/mismatch operators. Neither (1) nor (2) is valid. The counterexample to (1) is given in the introduction. The problem with (2) is that it is not closed under substitution. The following is an instance of (2) 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

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

This equality should not hold. Our T3b is the correction of (1) while our T3d is a special case of the correct version of (2). In order for (2) to be valid, the side condition has to be internalized as it were.

References

1. Y. Fu. A Proof Theoretical Approach to Communications. *ICALP'97*, Lecture Notes in Computer Science 1256, Springer, 325–335, 1997.
2. Y. Fu. Bisimulation Lattice of Chi Processes. *ASIAN'98*, Lecture Notes in Computer Science 1538, Springer, 245–262, 1998.
3. Y. Fu. Variations on Mobile Processes. *Theoretical Computer Science*, **221**: 327-368, 1999.
4. Y. Fu. Open Bisimulations of Chi Processes. *CONCUR'99*, Lecture Notes in Computer Science 1664, Springer, 304-319, 1999.
5. R. Milner, J. Parrow, D. Walker. A Calculus of Mobile Processes. *Information and Computation*, **100**: 1-40 (Part I), 41-77 (Part II), Academic Press.
6. J. Parrow, D. Sangiorgi. Algebraic Theories for Name-Passing Calculi. *Journal of Information and Computation*, **120**: 174-197, 1995.
7. J. Parrow, B. Victor. The Update Calculus. *AMAST '97*, Lecture Notes in Computer Science 1119, Springer, 389–405, 1997.
8. J. Parrow, B. Victor. The Tau-Laws of Fusion. *CONCUR '98*, Lecture Notes in Computer Science 1466, Springer, 99–114, 1998.
9. D. Sangiorgi. A Theory of Bisimulation for π-Calculus. *CONCUR 93*, Lecture Notes in Computer Science 715, Springer, 1993.
10. B. Victor, J. Parrow. Concurrent Constraints in the Fusion Calculus. *ICALP '98*, Lecture Notes in Computer Science 1443, Springer, 455–469, 1998.

A Axiomatic System for the Strong Open Congruence

E1	$P = P$	
E2	$P = Q$	if $Q = P$
E3	$P = R$	if $P = Q$ and $Q = R$
C1	$\alpha[x].P = \alpha[x].Q$	if $P = Q$
C2	$(x)P = (x)Q$	if $P = Q$
C3a	$[x{=}y]P = [x{=}y]Q$	if $P = Q$
C3b	$[x{\neq}y]P = [x{\neq}y]Q$	if $P = Q$
C4	$P{+}R = Q{+}R$	if $P = Q$
C5	$P_0 \vert P_1 = Q_0 \vert Q_1$	if $P_0 = Q_0$ and $P_1 = Q_1$
L1	$(x)\mathbf{0} = \mathbf{0}$	
L2	$(x)\alpha[y].P = \mathbf{0}$	$x \in \{\alpha, \overline{\alpha}\}$
L3	$(x)\alpha[y].P = \alpha[y].(x)P$	$x \notin \{y, \alpha, \overline{\alpha}\}$
L4	$(x)(y)P = (y)(x)P$	
L5	$(x)[y{=}z]P = [y{=}z](x)P$	$x \notin \{y, z\}$
L6	$(x)[x{=}y]P = \mathbf{0}$	$x{\neq}y$
L7	$(x)(P{+}Q) = (x)P{+}(x)Q$	
L8	$(x)[y \vert z].P = [y \vert z].(x)P$	$x \notin \{y, z\}$
L9	$(x)[y \vert x].P = \tau.P[y/x]$	$y \neq x$
L10	$(x)[x \vert x].P = \tau.(x)P$	
M1	$\phi P = \psi P$	if $\phi \Leftrightarrow \psi$
M2	$[x{=}y]P = [x{=}y]P[y/x]$	
M3a	$[x{=}y](P{+}Q) = [x{=}y]P{+}[x{=}y]Q$	
M3b	$[x{\neq}y](P{+}Q) = [x{\neq}y]P{+}[x{\neq}y]Q$	
M4	$P = [x{=}y]P{+}[x{\neq}y]P$	
M5	$[x{\neq}x]P = \mathbf{0}$	
S1	$P{+}\mathbf{0} = P$	
S2	$P{+}Q = Q{+}P$	
S3	$P{+}(Q{+}R) = (P{+}Q){+}R$	
S4	$P{+}P = P$	
U1	$[y \vert x].P = [x \vert y].P$	
U2	$[y \vert x].P = [y \vert x].[x{=}y]P$	
U3	$[x \vert x].P = \tau.P$	
LD1	$(x)[x \vert x].P = [y \vert y].(x)P$	by U3 and L8
LD2	$(x)[y{\neq}z]P = [y{\neq}z](x)P$	by L5, L7 and M4
LD3	$(x)[x{\neq}y]P = (x)P$	by L6, L7 and M4
MD1	$[x{=}y].\mathbf{0} = \mathbf{0}$	by S1, S4 and M4
MD2	$[x{=}x].P = P$	by M1
MD3	$\phi P = \phi(P\sigma)$ where σ agrees with ϕ	by M2
SD1	$\phi P{+}P = P$	by S-rules and M4
UD1	$[y \vert x].P = [y \vert x].P[y/x]$	by U2 and M2