# Solos in Concert

Cosimo Laneve[1] and Björn Victor[2]

[1] Dept. of Computer Science, University of Bologna, Italy.
[2] Dept. of Computer Systems, Uppsala University, Sweden.

**Abstract.** We present a calculus of mobile processes without prefix or summation, and using two different encodings we show that it can express both action prefix and guarded summation. One encoding gives a strong correspondence but uses a match operator; the other yields a slightly weaker correspondence but uses no additional operators.

## 1 Introduction

The fusion calculus was introduced by Parrow and Victor [15, 20, 16] as a simplification and extension of the $\pi$-calculus [6]. The simplification is easy to see: there is only one binding operator where $\pi$ has two; input and output are completely symmetric which they are not in $\pi$; and it has only one sensible bisimulation congruence where $\pi$ has three. The extension is that the effects of communication need not be local to the recipient process. Furthermore the fusion calculus contains the $\pi$-calculus as a proper subcalculus and thus inherits all its expressive power.

In recent years the *asynchronous* $\pi$-calculus [3, 5] has gained interest; here the asymmetry between input and output is further increased by dropping continuations from output prefixes. The resulting calculus has significant expressive power, and is also motivated by practical implementation in distributed systems.

In the fusion calculus it would be unfortunate to break the symmetry between input and output in order to develop an asynchronous subcalculus. Indeed, in this paper we show that continuations may be removed both from inputs and outputs – hereafter called *solos* – without loss of expressive power. More precisely, the *fusion calculus of solos*, where prefixes are replaced by solos and summation is removed, is expressive enough to encode prefixes and guarded summation.

We give two different encodings. One preserves strong barbed bisimulation but uses a match operator; the other yields weak barbed bisimulation but uses no additional operators. Both preserve divergence.

A key of our encodings in this paper is the use of *catalyst agents* of the type $(z)u\,zz$, which inputs the same name twice. If some agent in parallel composition with $(z)u\,zz$ sends any two names on $u$, they will be fused and made indistinguishable everywhere. In one of our encodings such a fusion effect enables an agent guarded by a match construction testing two names for equality; in the other it removes the restriction of a private communication channel, thereby enabling the communication. The fusion effect performed by the reduction relation of the fusion calculus plays a crucial role in these encodings – the same machinery cannot be used in the $\pi$-calculus.

Another important factor is that the calculus is polyadic: an input or output can carry arbitrarily many objects. In the strictly monadic calculus we strongly conjecture that the expressiveness of prefixes is strictly greater than that of solos.

*Related work:* Parrow shows in [13] that in the $\pi$-calculus without match, any agent can be encoded as a *concert of trios*, i.e., a parallel composition of possibly replicated prefixes $\alpha_1 . \alpha_2 . \alpha_3$ up to weak open equivalence [18]. He also shows that *duos*, i.e., prefixes nested to depth 2, are not sufficient. In this paper we show that for the fusion calculus, it suffices with *solos*, i.e., we do not need prefixes at all.

Nestmann and Pierce show in [10] that input-guarded choice $\sum_i u_i(\widetilde{x_i}) . P_i$ can be encoded in the asynchronous $\pi$-calculus without choice, up to coupled bisimulation [14], and in [9] Nestmann gives encodings of separate and mixed guarded choice. While these encodings involve increasingly complex communication protocols, our encodings of separate choice are simpler and work up to stronger equivalences.

In [11] Palamidessi shows that there can not exist an encoding of mixed choice into the asynchronous $\pi$-calculus which is *uniform* and preserves a *reasonable* semantics. A *uniform* encoding is compositional, meaning that $[\![P \mid Q]\!] = [\![P]\!] \mid [\![Q]\!]$, and respects injective renaming of free names. A *reasonable* semantics distinguishes two processes $P$ and $Q$ if the actions in some computation of $P$ are different from those in any computation of $Q$; in particular it is sensitive to divergence. Nestmann [9] argues that these criteria are too strong for practical purposes, and that by allowing a top-level context (but keeping the inner encoding compositional), or relaxing reasonableness, many practically motivated encodings turn out to be "good". Indeed even more theoretically motivated encodings often use top-level contexts, including our second encoding in this paper; our first encoding does not need this, but is in fact uniform and reasonable in Palamidessi's sense.

Yoshida in [21] presents separation results between subcalculi of the asynchronous $\pi$-calculus (without match and choice) by means of concurrent combinators, and shows the non-existence of encodings between such subcalculi. Here a concept of *standard encoding* is used, which means that the encoding is homomorphic, respects injective substitutions, and preserves weak observations and reductions. In addition to this the encodings are required to be *message-preserving*, i.e., $[\![\overline{u}\, x]\!] \approx \overline{u}\, x$. While our first encoding is standard by this definition, neither one is message-preserving. Yoshida works with monadic calculi, where the requirement may be quite sensible, but in a polyadic setting this requirement seems very strong, especially when only considering encoded contexts. Yoshida also proves that the *reflexive $\pi$-calculus*, a variant of the $\pi$-calculus similar to the monadic fusion calculus of solos, does not contain a synchronizer agent such as $a(x) . b(y) . \overline{c}\, y$, and is therefore less expressive than the monadic asynchronous $\pi$-calculus. In this paper we show that in the polyadic fusion calculus of solos such an agent can indeed be encoded, although our encodings are not message-preserving, but rather adds a "protocol header" to the data being sent and received.

*Acknowledgement:* We thank Uwe Nestmann and Joachim Parrow for valuable comments and remarks.

## 2 The Fusion Calculus of Solos

In this section we first present a subcalculus of the fusion calculus where we use *solos* of the form $u\,\widetilde{x}$ in place of general prefixes of the form $u\,\widetilde{x}\,.\,P$, and leave out summation. We present the syntax and semantics, review the barbed equivalences and congruences, and introduce a barbed expansion preorder.

### 2.1 Syntax and Semantics

We assume an infinite set $\mathcal{N}$ of *names* ranged over by $u, v, \ldots, z$. Names represent communication channels, which are also the values transmitted. We write $\widetilde{x}$ for a (possibly empty) finite sequence $x_1 \cdots x_n$ of names.

**Definition 1.** *The* solos*, ranged over by $\alpha$, and the* agents*, ranged over by $P, Q, \ldots$, are defined by*

$$
\begin{array}{rll@{\qquad\qquad}rl}
\alpha ::= & u\,\widetilde{x} & \text{(Input)} & P ::= \mathbf{0} & \text{(Inaction)}\\
 & \overline{u}\,\widetilde{x} & \text{(Output)} & \alpha & \text{(Solo)}\\
 & & & Q \mid R & \text{(Composition)}\\
 & & & (x)Q & \text{(Scope)}\\
 & & & [x = y]Q & \text{(Match)}
\end{array}
$$

In solos, the names $\widetilde{x}$ are the *objects* of the solo, and the name $u$ is the *subject*. We write $a$ to stand for either $u$ or $\overline{u}$, thus $a\widetilde{x}$ is the general form of a solo.

A Composition allows two solos to interact. The Scope $(x)Q$ limits the scope of $x$ to $Q$; the name $x$ is said to be *bound* in $(x)P$. We write $(\widetilde{x})P$ for $(x_1) \cdots (x_n)P$, $n \geq 0$. The *free names* in $P$, denoted $\mathrm{fn}(P)$, are the names in $P$ with a non-bound occurrence. As usual we will not distinguish between alpha-variants of agents, i.e., agents differing only in the choice of bound names.

A Match $[x = y]Q$ acts like $Q$ if $x$ and $y$ are the same name. We use $M, N$ to stand for a match operator, and write "match sequence" for a sequence of match operators, ranged over by $\widetilde{M}$, $\widetilde{N}$. We also write $\widetilde{M} \Leftrightarrow \widetilde{N}$ if the conjunction of all matches in $\widetilde{M}$ logically implies all elements in $\widetilde{N}$ and vice versa. We write $\mathrm{n}(M)$ for the names occurring in $M$.

### 2.2 Semantics

In the style of the Chemical Abstract Machine [1], we now define a structural congruence which equates all agents we will never want to distinguish for any semantic reason, and then use this when giving the operational semantics.

**Definition 2.** *The* structural congruence*, $\equiv$, between agents is the least congruence satisfying the abelian monoid laws for Composition (associativity, commutativity and* $\mathbf{0}$ *as identity), and the scope laws*

$$(x)\mathbf{0} \equiv \mathbf{0}, \quad (x)(y)P \equiv (y)(x)P, \quad [x=x]P \equiv P$$
$$(x)MP \equiv M(x)P, \quad \textit{if } x \notin \mathrm{n}(M)$$
$$P \mid (z)Q \equiv (z)(P \mid Q), \quad \textit{if } z \notin \mathrm{fn}(P)$$

We write $\{\widetilde{x} = \widetilde{y}\}$ for the smallest total equivalence relation on $\mathcal{N}$ relating each $x_i$ with $y_i$, and say that a substitution $\sigma$ *agrees with* the equivalence $\varphi$ if $\forall x, y : x\,\varphi\,y \Leftrightarrow \sigma(x) = \sigma(y)$.

The reduction relation of the fusion calculus is the least relation satisfying the rules in Table 1, where structurally equivalent agents are considered the same. Here and in the following we use $\mathrm{dom}(\sigma) = \{u : \sigma(u) \neq u\}$ and $\mathrm{ran}(\sigma) = \{\sigma(u) : \sigma(u) \neq u\}$.

$$(\widetilde{z})\Big(\widetilde{M} u\, \widetilde{x} \mid \widetilde{N} \overline{u}\, \widetilde{y} \mid R\Big) \longrightarrow R\sigma$$
$$\text{if } |\widetilde{x}| = |\widetilde{y}|, \widetilde{M} \Leftrightarrow \widetilde{N} \Leftrightarrow \mathsf{true},$$
$$\sigma \text{ agrees with } \{\widetilde{x} = \widetilde{y}\}, \mathrm{ran}(\sigma) \cap \widetilde{z} = \emptyset,$$
$$\text{and } \mathrm{dom}(\sigma) = \widetilde{z}$$

$$\frac{P \longrightarrow P'}{P \mid Q \longrightarrow P' \mid Q} \qquad \frac{P \longrightarrow P'}{(x)P \longrightarrow (x)P'}$$

$$\frac{P \equiv Q \quad Q \longrightarrow Q' \quad Q' \equiv P'}{P \longrightarrow P'}$$

**Table 1.** Reduction rules for the fusion calculus of solos.

For examples and motivations we refer the reader to [15, 20, 16].

### 2.3 Equivalence and Preorder

We will use the standard idea of *barbed bisimulation* developed by Milner and Sangiorgi [7] in the setting of CCS, further investigated in a $\pi$-calculus setting by Sangiorgi [17], and later used in many other calculi as an intuitive observational equivalence. The idea is that two agents are considered equivalent if their reductions match and they are indistinguishable under global observations.

**Definition 3.** *The* observation relation *is the least relation satisfying the rules below.*

| | |
|---|---|
| $x\,\widetilde{y} \downarrow x$ | $[x=x]P \downarrow y$ *if* $P \downarrow y$ |
| $\overline{x}\,\widetilde{y} \downarrow x$ | $(P \mid Q) \downarrow x$ *if* $P \downarrow x$ *or* $Q \downarrow x$ |
| | $(x)P \downarrow y$ *if* $P \downarrow y$ *and* $x \neq y$ |

**Definition 4.** *A* strong barbed bisimulation *is a symmetric binary relation $\mathcal{S}$ between agents such that $P\,\mathcal{S}\,Q$ implies:*

1. *If $P \longrightarrow P'$ then $Q \longrightarrow Q'$ and $P'\,\mathcal{S}\,Q'$.*
2. *If $P \downarrow x$ for some $x$, then $Q \downarrow x$.*

*$P$ is strong barbed bisimilar to $Q$, written $P \overset{\cdot}{\sim} Q$, if $P \mathcal{S} Q$ for some strong barbed bisimulation $\mathcal{S}$.*

To define the weak barbed bisimulation and congruence, we change $Q \longrightarrow Q'$ to $Q \longrightarrow^* Q'$ and $Q \downarrow x$ to $Q \longrightarrow^* \downarrow x$ (written $Q \Downarrow x$) in Definition 4.

**Definition 5.** *A* weak barbed bisimulation *is a symmetric binary relation $\mathcal{S}$ between agents such that $P \mathcal{S} Q$ implies:*

1. *If $P \longrightarrow P'$ then $Q \longrightarrow^* Q'$ and $P' \mathcal{S} Q'$.*
2. *If $P \downarrow x$ for some $x$, then $Q \Downarrow x$.*

*$P$ is weak barbed bisimilar to $Q$, written $P \overset{\cdot}{\approx} Q$, if $P \mathcal{S} Q$ for some weak barbed bisimulation $\mathcal{S}$.*

We will also make use of an expansion relation [8], an asymmetric form of weak barbed bisimulation where $P \lesssim Q$ means that $P \overset{\cdot}{\approx} Q$ in a way such that $P$ does no more reductions than $Q$. In the following we write $P \hat{\longrightarrow} P'$ if $P \longrightarrow P'$ or $P \equiv P'$.

**Definition 6.** *A* weak barbed expansion *is a binary relation $\mathcal{S}$ between agents such that $P \mathcal{S} Q$ implies:*

1. *If $P \longrightarrow P'$ then $Q \longrightarrow^+ Q'$ and $P' \mathcal{S} Q'$.*
2. *If $Q \longrightarrow Q'$ then $P \hat{\longrightarrow} P'$ and $P' \mathcal{S} Q'$.*
3. *If $P \downarrow x$ for some $x$ then $Q \Downarrow x$.*
4. *If $Q \downarrow x$ for some $x$ then $P \downarrow x$.*

*$Q$* expands *$P$, written $P \lesssim Q$, if $P \mathcal{S} Q$ for some weak barbed expansion $\mathcal{S}$. We often write $Q \gtrsim P$ instead of $P \lesssim Q$.*

## 3 Encodings of Prefixes

In this section and the following we display the expressiveness of the fusion calculus of solos by means of encodings. We first encode the general prefix operator of the fusion calculus using solos. Two such encodings are presented: one using match operators, resulting in a strong operational correspondence with the encoded terms; and one using only solos, scope and parallel composition, yielding a weaker correspondence.

We now add the prefix operator to the calculus by allowing processes of the form $\alpha \,.\, P$. We add two observation rules:

$$x\,\tilde{y}\,.\,P \downarrow x \qquad\qquad \overline{x}\,\tilde{y}\,.\,P \downarrow x$$

and the following reduction rule:

$$(\tilde{z})\big(\widetilde{M} u\,\tilde{x}\,.\,P \mid \tilde{N}\overline{u}\,\tilde{y}\,.\,Q \mid R\big) \longrightarrow (P \mid Q \mid R)\sigma$$

if $|\tilde{x}| = |\tilde{y}|$, $\widetilde{M} \Leftrightarrow \tilde{N} \Leftrightarrow \mathtt{true}$, $\sigma$ agrees with $\{\tilde{x} = \tilde{y}\}$, $\mathrm{ran}(\sigma) \cap \tilde{z} = \emptyset$ and $\mathrm{dom}(\sigma) = \tilde{z}$. We call the resulting calculus the *fusion calculus with prefix*, $f_{\mathrm{pre}}$. In this calculus we regard a solo $a\tilde{x}$ as shorthand for the prefix $a\tilde{x}\,.\,\mathbf{0}$.

$$\begin{aligned}
[\![u\,\widetilde{x}\,.\,P]\!] &\stackrel{def}{=} (zw)(u\,\widetilde{x}zww \mid [z=w][\![P]\!]) \\
[\![\overline{u}\,\widetilde{x}\,.\,P]\!] &\stackrel{def}{=} (zw)(\overline{u}\,\widetilde{x}wwz \mid [z=w][\![P]\!]) \\
[\![[x=y]P]\!] &\stackrel{def}{=} [x=y][\![P]\!] \\
[\![P \mid Q]\!] &\stackrel{def}{=} [\![P]\!] \mid [\![Q]\!] \\
[\![(x)P]\!] &\stackrel{def}{=} (x)[\![P]\!]
\end{aligned}$$

**Table 2.** Encoding of prefixes using match. $z$ and $w$ are fresh.

### 3.1 Encoding using Match

The encoding of prefixes using match, shown in Table 2, utilizes the fusion power of two interacting solos. The encoding of an input prefix $u\,\widetilde{x}\,.\,P$ creates two fresh names $z$ and $w$. The continuation $P$ of the prefix is guarded by a match operator checking for equality between $z$ and $w$; being fresh, these are initially different from each other, so $P$ cannot reduce. The input prefix action $u\,\widetilde{x}$ is encoded by a solo with the same subject and polarity, but with three additional objects $zww$ appended to $\widetilde{x}$. An output prefix $\overline{u}\,\widetilde{y}\,.\,Q$ is encoded symmetrically, but with the order of the additional objects changed to $wwz$. When such input and output solos interact, the result is a fusion of the names $z$ and $w$ on each side of the interaction, thus triggering the continuations $P$ and $Q$. Increasing polyadicity of names to encode the temporal ordering of prefixes was also used by Parrow in [12].

To get acquainted with the encoding of Table 2 we detail the interactions of the encoding of two parallel agents:

$$\begin{aligned}
&[\![(x)(u\,x\,.\,P \mid \overline{u}\,y\,.\,Q)]\!] \\
&\stackrel{def}{=} (x)(\ (zw)(u\,xzww \mid [z=w][\![P]\!]) \\
&\qquad\quad \mid (zw)(\overline{u}\,ywwz \mid [z=w][\![Q]\!])) \\
&\equiv\ (xz_1z_2w_1w_2)(\ u\,xz_1w_1w_1 \mid [z_1=w_1][\![P]\!] \\
&\qquad\qquad\qquad\quad \mid \overline{u}\,yw_2w_2z_2 \mid [z_2=w_2][\![Q]\!]) \\
&\longrightarrow (z_1)(([z_1=w_1][\![P]\!] \mid [z_2=w_2][\![Q]\!])\{y/x, z_1/w_1, z_1/w_2, z_1/z_2\}) \\
&\equiv\ ([\![P \mid Q]\!])\{y/x\} \\
&=\ [\![(P \mid Q)\{y/x\}]\!]
\end{aligned}$$

which corresponds exactly to the result of the encoded prefix agents interacting.

This operational correspondence is formalized in the following lemmas:

**Lemma 7.** *For $P$ an agent of $f_{\mathrm{pre}}$,*

1. *if $P \equiv P'$ then $[\![P]\!] \equiv [\![P']\!]$;*
2. *if $P \longrightarrow P'$ then $[\![P]\!] \longrightarrow [\![P']\!]$;*
3. *if $P \downarrow x$ then $[\![P]\!] \downarrow x$.*

**Lemma 8.** *For $P$ an agent of $f_{\mathrm{pre}}$,*

1. *if $[\![P]\!] \longrightarrow Q$, then $P \longrightarrow P'$ such that $Q \equiv [\![P']\!]$;*

*2. if* $[\![P]\!] \downarrow x$ *for some* $x$*, then* $P \downarrow x$.

We have full abstraction up to barbed bisimulation:

**Theorem 9.** *For* $P, Q$ *two agents of* $f_{\mathrm{pre}}$*,* $P \dot{\sim} Q$ *iff* $[\![P]\!] \dot{\sim} [\![Q]\!]$.

*Proof. By Lemmas 7 and 8, showing that* $\{([\![P]\!], [\![Q]\!]) : P \dot{\sim} Q\}$ *and* $\{(P, Q) : [\![P]\!] \dot{\sim} [\![Q]\!]\}$ *are barbed bisimulations.*

### 3.2 Encoding without Match

While the encoding above has very pleasant properties, it may for reasons of minimality be desirable to cope without the match operator. A new encoding is presented in Table 3.

$$
\begin{aligned}
U_v &\equiv (z)v\,zzv \\
[\![u\,\widetilde{x}\,.\,P]\!]_v &\stackrel{def}{=} (wv')(w\,\widetilde{x}vv' \mid [\![P]\!]_{v'} \mid (y)(\overline{v}\,uwy \mid U_y)) \\
[\![\overline{u}\,\widetilde{x}\,.\,P]\!]_v &\stackrel{def}{=} (wv')(\overline{w}\,\widetilde{x}v'v \mid [\![P]\!]_{v'} \mid (y)(\overline{v}\,uwy \mid U_y)) \\
[\![[x=y]P]\!]_v &\stackrel{def}{=} [x=y][\![P]\!]_v \\
[\![(x)P]\!]_v &\stackrel{def}{=} (x)[\![P]\!]_v \\
[\![P \mid Q]\!]_v &\stackrel{def}{=} [\![P]\!]_v \mid [\![Q]\!]_v \\
[\!(P)\!] &\stackrel{def}{=} (v)([\![P]\!]_v \mid U_v)
\end{aligned}
$$

**Table 3.** Encoding of prefixes without using match. $v, w$ and $v'$ are fresh.

In Table 2 the subject of a prefix is encoded through a solo with the same subject. In Table 3 the subject of a prefix is instead encoded by a fresh scoped name $w$. Consequently it has no reactions. Instead the whole encoding has a parameter $v$, and an interaction over this parameter can fuse the fresh name to the original subject, removing the scope of $w$ and eventually enabling a reaction. In order to achieve this we introduce *catalyst agents* $U_v \equiv (z)v\,zzv$ and we add an initial catalyst in the top level encoding $[\!(\cdot)\!]$.

An example illustrating the encoding follows:

$$
\begin{aligned}
&[\!((x)(u\,x\,.\,P \mid \overline{u}\,y\,.\,Q))\!] \\
&\stackrel{def}{=} (vx)\big(\; (wv')(w\,xvv' \mid [\![P]\!]_{v'} \mid (y)(\overline{v}\,uwy \mid U_y)) && (1)\\
&\qquad\qquad \mid (wv')(\overline{w}\,yv'v \mid [\![Q]\!]_{v'} \mid (y)(\overline{v}\,uwy \mid U_y)) \\
&\qquad\qquad \mid (z)v\,zzv\big) \\
&\longrightarrow (vxwv_1'v_2')(\;\; u\,xvv_1' \mid [\![P]\!]_{v_1'} \mid (z)v\,zzv && (2)\\
&\qquad\qquad\qquad \mid \overline{w}\,yv_2'v \mid [\![Q]\!]_{v_2'} \mid (y)(\overline{v}\,uwy \mid U_y)) \\
&\longrightarrow (vxv_1'v_2')(u\,xvv_1' \mid [\![P]\!]_{v_1'} \mid \overline{u}\,yv_2'v \mid [\![Q]\!]_{v_2'} \mid U_v) && (3)\\
&\longrightarrow (v)([\![P]\!]_v \mid [\![Q]\!]_v \mid U_v)\{y/x\}
\end{aligned}
$$

$$
\begin{aligned}
&\stackrel{def}{=} (v)(\llbracket P \mid Q \rrbracket_v \{y/x\} \mid U_v) \\
&= \llparenthesis (P \mid Q)\{y/x\} \rrparenthesis \qquad (4)
\end{aligned}
$$

Initially the solos corresponding to the prefix actions can not interact, since their subjects are locally scoped. Expanding the definitions we can see at (1) that the catalyst $(z)v\,zzv$ can interact with one of the terms $\overline{v}\,uwy$, thereby changing the subject of the prefix solo, removing the scope. The initial catalyst is consumed when it interacts with the term $\overline{v}\,uwy$, but this interaction also changes the subterm $U_y$ into a new catalyst $U_v$. This can be used at (2) to remove the scope of the other prefix solo, enabling the interaction between the two prefixes at (3) and producing another new catalyst. Observe that as a side effect of this latter interaction, the continuations $\llbracket P \rrbracket_{v_1'}$ and $\llbracket Q \rrbracket_{v_2'}$ are now enabled since $v_1'$ and $v_2'$ are fused with $v$. We end up with the desired result (4).

The operational and observational correspondences of the encoding are expressed by the following lemmas.

**Lemma 10.** *For $P$ an agent of $f_{\mathrm{pre}}$,*

1. *$P \equiv Q$ implies $\llparenthesis P \rrparenthesis \equiv \llparenthesis Q \rrparenthesis$;*
2. *$P \longrightarrow Q$ implies $\llparenthesis P \rrparenthesis \longrightarrow^{+} \llparenthesis Q \rrparenthesis$;*
3. *$P \downarrow x$ implies $\llparenthesis P \rrparenthesis \Downarrow x$.*

**Lemma 11.** *For $P$ an agent of $f_{\mathrm{pre}}$,*

1. *if $\llparenthesis P \rrparenthesis \longrightarrow Q$, then $P \hat{\longrightarrow} P'$ such that $Q \gtrsim \llparenthesis P' \rrparenthesis$;*
2. *if $\llparenthesis P \rrparenthesis \longrightarrow^{*} Q$, then $P \longrightarrow^{*} P'$ such that $Q \gtrsim \llparenthesis P' \rrparenthesis$;*
3. *if $\llparenthesis P \rrparenthesis \hat{\longrightarrow} \downarrow x$ for some $x$, then $P \downarrow x$;*
4. *if $\llparenthesis P \rrparenthesis \Downarrow x$ for some $x$, then $P \Downarrow x$.*

*Proof.* *1. By induction on the depth of the derivation.*
*2. By induction on the number of reductions in the left hand side, using (1).*
*3. By induction on the depth of the derivation.*
*4. By induction on the number of reductions in the left hand side, using (3).*

The results of the previous encoding are weakened because of the weaker form of operational correspondence.

**Theorem 12.** *For $P, Q$ two agents of $f_{\mathrm{pre}}$, $P \dot{\approx} Q$ iff $\llparenthesis P \rrparenthesis \dot{\approx} \llparenthesis Q \rrparenthesis$.*

*Proof.* *Using the fact that $\{(P,Q) : P \gtrsim\dot{\approx}\lesssim Q\}$ and $\{(P,Q) : P \lesssim\dot{\approx}\gtrsim Q\}$ are both weak barbed bisimulations, it is easy to show that $\{(\llparenthesis P \rrparenthesis, \llparenthesis Q \rrparenthesis) : P \dot{\approx} Q\}$ and $\{(P,Q) : \llparenthesis P \rrparenthesis \dot{\approx} \llparenthesis Q \rrparenthesis\}$ are weak barbed bisimulations. We then use Lemmas 10 and 11 to complete the proof.*

### 3.3 Replication and Recursion

We can add replication to the $f_{\mathrm{pre}}$ calculus by the structural law $!\,P \equiv P \mid !\,P$. Extending our first encoding using match (Table 2) with $[\![!\,P]\!] =! [\![P]\!]$, Lemmas 7, 8 and Theorem 9 still hold. We can further strengthen these results by proving preservation of divergence properties.

**Theorem 13.** *For $P$ an agent of $f_{\mathrm{pre}}$ with replication, $P$ diverges iff $[\![P]\!]$ diverges.*

Extending the second encoding (Table 3) in the same way does not preserve divergence, since $[\!(!\,u\,.\,\mathbf{0})\!]$ could reduce indefinitely even though the original term can not reduce. However, if we replace replication with guarded recursion, introduced by the structural law $A(\widetilde{y}) \equiv P\{\widetilde{y}/\widetilde{x}\}$ if $A(\widetilde{x}) \stackrel{def}{=} P$ and $|\widetilde{x}| = |\widetilde{y}|$, and the requirement that process variables only occur under a prefix, then Lemmas 10 and 11 and Theorem 12 still hold, and furthermore:

**Theorem 14.** *For $P$ an agent of $f_{\mathrm{pre}}$ with guarded recursion, $P$ diverges iff $[\!(P)\!]$ diverges.*

### 3.4 Other translations

There is a simpler translation which requires that channels carry just one message more instead of two, with the added cost of a further book-keeping interaction for every original interaction. We only show the translation rules for prefixes since the others are the same:

$$
\begin{aligned}
[\![u\,\widetilde{x}\,.\,P]\!]_v &\stackrel{def}{=} (wz)(w\,\widetilde{x}z \mid (v')(z\,v'v \mid [\![P]\!]_{v'}) \mid (y)(\overline{v}\,uwy \mid U_y))\\
[\![\overline{u}\,\widetilde{x}\,.\,P]\!]_v &\stackrel{def}{=} (wz)(\overline{w}\,\widetilde{x}z \mid (v')(\overline{z}\,vv' \mid [\![P]\!]_{v'}) \mid (y)(\overline{v}\,uwy \mid U_y))
\end{aligned}
$$

Once replication has been added, we can also get rid of the catalyst agents inside the encodings of prefixes of $[\![\,\cdot\,]\!]_v$. Instead we use a replicated catalyst at top level. Catalysts now need only two objects:

$$
\begin{aligned}
[\![u\,\widetilde{x}\,.\,P]\!]_v &\stackrel{def}{=} (wv')(w\,\widetilde{x}vv' \mid [\![P]\!]_{v'} \mid \overline{v}\,uw)\\
[\![\overline{u}\,\widetilde{x}\,.\,P]\!]_v &\stackrel{def}{=} (wv')(\overline{w}\,\widetilde{x}v'v \mid [\![P]\!]_{v'} \mid \overline{v}\,uw)\\
[\!(P)\!] &\stackrel{def}{=} (v)([\![P]\!]_v \mid !\,(z)v\,zz)
\end{aligned}
$$

We can also get rid of the initial book-keeping reductions that enable unguarded prefixes in the encoding of Table 3, as well as the initial catalyst agent of $[\!(\cdot)\!]$, by considering the following auxiliary function (again we only illustrate the rules for prefixes):

$$
\begin{aligned}
(\!(u\,\widetilde{x}\,.\,P)\!) &\stackrel{def}{=} (z)(u\,\widetilde{x}z \mid (vv')(z\,vv' \mid [\![P]\!]_{v'} \mid v\,zzv))\\
(\!(\overline{u}\,\widetilde{x}\,.\,P)\!) &\stackrel{def}{=} (z)(\overline{u}\,\widetilde{x}z \mid (v)(\overline{z}\,vv \mid [\![P]\!]_v))
\end{aligned}
$$

## 4 Encoding Choice

In this section we present encodings of the choice operator; $P + Q$ allows $P$ or $Q$ to take part in reduction, and discards the other branch when doing so. Here we add the mismatch operator $[x \neq y]P$, which can act like $P$ if $x$ and $y$ are different. We extend the ranges of $M, N, \widetilde{M}$ and $\widetilde{N}$ and the definition of $\widetilde{M} \Leftrightarrow \widetilde{N}$ appropriately, add the reduction rule

$$\frac{P \longrightarrow P',\ x \neq y}{[x \neq y]P \longrightarrow P'}$$

and the observation rule $[x \neq z]P \downarrow y$ if $P \downarrow y$ and $x \neq z$. We also extend our previous encodings homomorphically for the mismatch operator.

Restricting the general choice operator $P + Q$ to guarded choice, $\sum_I \alpha_i \,.\, P_i$, and further requiring that all $\alpha_i$ in a guarded choice have the same polarity (all inputs or all outputs), we can extend the encoding of Table 2 by replacing the encoding of prefixes by the following, where $z$ and $w$ are fresh:

$$\left[\!\!\left[ \sum_I u_i\, \widetilde{x_i} \,.\, P_i \right]\!\!\right] \stackrel{def}{=} (zw) \prod_I [z \neq w](u_i\, \widetilde{x}_i zww \mid [z = w][\![P_i]\!])$$
$$\left[\!\!\left[ \sum_I \overline{u_i}\, \widetilde{x_i} \,.\, P_i \right]\!\!\right] \stackrel{def}{=} (zw) \prod_I [z \neq w](\overline{u_i}\, \widetilde{x}_i wwz \mid [z = w][\![P_i]\!])$$

The mismatch operator is used in a "test-and-set" construction which tests two names $z$ and $w$ for inequality, and if they are not equal, atomically makes them so. Only one branch of the choice can succeed in doing this. The interaction between an input and an output prefix now not only enables the continuations of the prefixes, but also *disables* the other branches of the choice.

Lemmas 7, 8 and Theorems 9 and 13 still hold for this extended encoding (Lemma 7(2) and 8(1) respectively weakened from $\longrightarrow$ and $\equiv$ to $\longrightarrow\dot{\sim}$ and $\dot{\sim}$ in their consequences).

The encoding of Table 3 can also be extended in a similar way, replacing the encodings of prefixes ($v$ and $v'$ are fresh):

$$\left[\!\!\left[ \sum_I u_i\, \widetilde{x_i} \,.\, P_i \right]\!\!\right]_v \stackrel{def}{=} (v') \prod_I [v \neq v'](w)(w\, \widetilde{x}_i vv' \mid [\![P_i]\!]_{v'} \mid (y)(\overline{v}\, u_i wy \mid U_y))$$
$$\left[\!\!\left[ \sum_I \overline{u_i}\, \widetilde{x_i} \,.\, P_i \right]\!\!\right]_v \stackrel{def}{=} (v') \prod_I [v \neq v'](w)(\overline{w}\, \widetilde{x}_i v' v \mid [\![P_i]\!]_{v'} \mid (y)(\overline{v}\, u_i wy \mid U_y))$$

Lemmas 10 and 11 and Theorem 12 as well as Theorem 14 hold also for this encoding (Lemma 10(2) weakened from $\longrightarrow$ to $\longrightarrow\dot{\approx}$ in the consequence).

## References


[1] G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96:217–248, 1992.

[2] M. Boreale. On the expressiveness of internal mobility in name-passing calculi. *Theoretical Computer Science*, 195(2):205–226, 1998.

[3] G. Boudol. Asynchrony and the $\pi$-calculus (note). Rapport de Recherche 1702, INRIA Sofia-Antipolis, May 1992.

[4] R. Cleaveland, editor. *Proc. of CONCUR '92*, volume 630 of *LNCS*. Springer, 1992.

[5] K. Honda and M. Tokoro. On asynchronous communication semantics. In M. Tokoro, O. Nierstrasz, and P. Wegner, editors, *Object-Based Concurrent Computing 1991*, volume 612 of *LNCS*, pages 21–51. Springer, 1992.

[6] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, part I/II. *Journal of Information and Computation*, 100:1–77, Sept. 1992.

[7] R. Milner and D. Sangiorgi. Barbed bisimulation. In W. Kuich, editor, *Proc. of ICALP '92*, volume 623 of *LNCS*, pages 685–695. Springer, 1992.

[8] R. Milner and D. Sangiorgi. The problem of "weak bisimulation up-to". In Cleaveland [4].

[9] U. Nestmann. What is a 'good' encoding of guarded choice? In C. Palamidessi and J. Parrow, editors, *Proc. of EXPRESS '97*, volume 7 of *ENTCS*. Elsevier Science Publishers, 1997. Revised version accepted (1998) for *Journal of Information and Computation*.

[10] U. Nestmann and B. C. Pierce. Decoding choice encodings. In U. Montanari and V. Sassone, editors, *Proc. of CONCUR '96*, volume 1119 of *LNCS*, pages 179–194. Springer, 1996.

[11] C. Palamidessi. Comparing the expressive power of the synchronous and the asynchronous $\pi$-calculus. In *Proc. of POPL '97*, pages 256–265. ACM, Jan. 1997.

[12] J. Parrow. Interaction diagrams. *Nordic Journal of Computing*, 2:407–443, 1995.

[13] J. Parrow. Trios in concert. In G. Plotkin, C. Stirling, and M. Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*, 1998. To appear.

[14] J. Parrow and P. Sjödin. Multiway synchronization verified with coupled bisimulation. In Cleaveland [4], pages 518–533.

[15] J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *Proc. of LICS '98*, 1998.

[16] J. Parrow and B. Victor. The tau-laws of fusion. In Sangiorgi and de Simone [19], pages 99–114.

[17] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms*. PhD thesis, LFCS, University of Edinburgh, 1993.

[18] D. Sangiorgi. A theory of bisimulation for the $\pi$-calculus. *Acta Informatica*, 33:69–97, 1996.

[19] D. Sangiorgi and R. de Simone, editors. *Proc. of CONCUR '98*, volume 1466 of *LNCS*. Springer, Sept. 1998.

[20] B. Victor and J. Parrow. Concurrent constraints in the fusion calculus. In K. G. Larsen, S. Skyum, and G. Winskel, editors, *Proc. of ICALP '98*, volume 1443 of *LNCS*, pages 455–469. Springer, July 1998.

[21] N. Yoshida. Minimality and separation results on asynchronous mobile processes: Representability theorem by concurrent combinators. Technical Report 5/98, School of Cognitive and Computing Sciences, University of Sussex, UK, Sept. 1998. An extended abstract appeared in *Proc. of CONCUR'98*, LNCS 1466.