To appear in MFCS 2000. Available online from
http://www.cl.cam.ac.uk/∼ljw1004

Explicit Fusions

Philippa Gardner and Lucian Wischik*

June 2000

Abstract

We introduce *explicit fusions* of names. To 'fuse' two names is to declare that they may be used interchangeably. An explicit fusion is one that can exist in parallel with some other process, allowing us to ask for instance how a process might behave in a context where $x = y$. We present the π_F-calculus, a simple process calculus with explicit fusions. It is similar in many respects to the fusion calculus but has a simple local reaction relation. We give embeddings of the π-calculus and the fusion calculus. We provide a bisimulation congruence for the π_F-calculus and compare it with hyper-equivalence in the fusion calculus.

1 Introduction

We introduce *explicit fusions* of names. To 'fuse' two names is to declare that they may be used interchangeably. An *explicit* fusion is one that can exist in parallel with some other process. For example, we can use the explicit fusion $\langle x{=}y\rangle$ to ask how a process might behave in a context where the addresses x and y are equal.

In this paper we focus on one particular application of explicit fusions. We introduce the π_F-calculus, which incorporates these fusions. It is similar to the π-calculus in that it has input and output processes which react together. It differs from the π-calculus in how they react. In a π-reaction, names are sent by the output process to replace abstracted names in the input process; this replacement is represented with a substitution. In contrast a π_F-reaction is directionless and *fuses* names; this is recorded with an explicit fusion.

The π_F-calculus is similar in many respects to the fusion calculus of Parrow and Victor [10, 13], and to the chi-calculus of Fu [1]. These calculi also have a directionless reaction which fuses names. The difference is in how the name-fusions have effect. In the fusion calculus, fusions occur implicitly within the reaction relation and their effect is immediate. In the π_F-calculus, fusions are explicitly recorded and their effect may be delayed. A consequence of this is that π_F-reaction is a simple local reaction between input and output processes.

Explicit fusions can be used to analyse, in smaller steps, reactions that occur in existing process calculi. We give embedding results for the π-calculus and the fusion calculus. These embeddings show that explicit fusions are expressive enough to describe both name-substitution in the π-reaction, and the fusions that occur in the fusion reaction. We are currently exploring an embedding of the λ-calculus in the π_F-calculus [14]. Intriguingly, explicit fusions allow for an embedding which is purely

*Computing Laboratory, University of Cambridge. Gardner is supported by an EPSRC Advanced Fellowship, Wischik by an EPSRC Studentship. Philippa.Gardner@cl.cam.ac.uk, ljw1004@cam.ac.uk.

compositional, in contrast with the analogous embeddings in the π-calculus and fusion calculus.

We provide a bisimulation congruence for the π_F-calculus, which is automatically closed with respect to substitution. We compare it with hyper-equivalence in the fusion calculus [10] and open bisimulation in the π-calculus [12].

2 The π_F-calculus

To illustrate the key features of the π_F-calculus, we contrast it to the fusion calculus. Both calculi have symmetric input and output processes. They have no abstraction operator. Instead, they interpret the π-calculus abstraction $(x)P$ with the concretion $(\nu x)\langle x\rangle P$. A π_F-reaction is

$$z.\langle x\rangle P \mid \overline{z}.\langle y\rangle Q \mid R \ \searrow_{\pi_F}\ \langle x{=}y\rangle \mid P \mid Q \mid R.$$

The reaction in this example is a local one between the input and output processes. However the effect of the resulting fusion $\langle x{=}y\rangle$ is global in scope: x and y can be used interchangeably throughout the entire process, including R. To limit the scope of the fusion, we use restriction. For example, restricting x in the above expression we obtain

$$(\nu x)(\langle x{=}y\rangle|P|Q|R) \ \equiv\ P\{y/x\} \mid Q\{y/x\} \mid R\{y/x\}.$$

Thus, using just explicit fusions and restriction, we can derive a name-substitution operator which behaves like the standard capture-avoiding substitution.

The corresponding reaction in the fusion calculus *requires* that either x or y be restricted: for instance,

$$(\nu x)\big(z.\langle x\rangle P \mid \overline{z}.\langle y\rangle Q \mid R\big) \ \searrow_{fu}\ P\{y/x\} \mid Q\{y/x\} \mid R\{y/x\}.$$

The x and y are implicitly fused during the reaction. If we had restricted y rather than x, then the substitution would have been $\{x/y\}$. The full polyadic reaction rule, using many $\vec{x}$s and $\vec{y}$s, is more complicated.

We assume an infinite set of names ranged over by $u, \ldots z$, and write $\vec{z}$ for a sequence of names and $|\vec{z}|$ for its length.

Definition 2.1 *The set $\mathcal{P}_{\pi_F}$ of* processes *of the π_F-calculus is defined by the grammar*

$$P \quad ::= \quad \mathbf{nil} \;\big|\; P|P \;\big|\; (\nu x)P \;\big|\; \langle x\rangle \;\big|\; \langle x{=}y\rangle \;\big|\; x.P \;\big|\; \overline{x}.P$$

We call the process $\langle x\rangle$ a datum, *and the process $\langle x{=}y\rangle$ a* fusion.

We say that a datum is at the *top-level* if it is not contained within an input or output process. The *arity* of a process is the number of top-level datums in it. We write $P : m$ to declare that P has arity m. More general arities are also possible, such as typing information similar to the sorting discipline for the π-calculus [8]. For simplicity, we consider in this paper only that fragment of the π_F-calculus without replication or summation. Replication is considered elsewhere [14].

Datums are primitive processes, with the process $\langle\vec{y}\rangle \mid P$ corresponding to the conventional concretion $\langle\vec{y}\rangle P$. The choice between datums and concretions does not affect the results in this paper. Our choice to use datums is motivated in [2, 14], where we

Standard axioms for $|$ and **nil**:

$$P|\mathbf{nil} \equiv P \qquad (P|Q)|R \equiv P|(Q|R) \qquad P|Q \equiv Q|P \text{ if } P:0$$

Standard scope axioms:

$$(\nu x)(P|Q) \equiv (\nu x)P|Q \quad \text{if } x \notin fn(Q) \qquad (\nu x)(\nu y)P \equiv (\nu y)(\nu x)P$$
$$(\nu x)(P|Q) \equiv P|(\nu x)Q \quad \text{if } x \notin fn(P)$$

Fusion axioms:

$$\langle x{=}x\rangle \equiv \mathbf{nil} \qquad \langle x{=}y\rangle \mid x.P \equiv \langle x{=}y\rangle \mid y.P \qquad \langle x{=}y\rangle \mid \langle x\rangle \equiv \langle x{=}y\rangle \mid \langle y\rangle$$
$$(\nu x)\langle x{=}y\rangle \equiv \mathbf{nil} \qquad \langle x{=}y\rangle \mid \overline{x}.P \equiv \langle x{=}y\rangle \mid \overline{y}.P \qquad \langle x{=}y\rangle \mid z.P \equiv \langle x{=}y\rangle \mid z.(\langle x{=}y\rangle|P)$$
$$\langle x{=}y\rangle \equiv \langle y{=}x\rangle \qquad \langle x{=}y\rangle \mid \langle x{=}z\rangle \equiv \langle x{=}y\rangle \mid \langle y{=}z\rangle \qquad \langle x{=}y\rangle \mid \overline{z}.P \equiv \langle x{=}y\rangle \mid \overline{z}.(\langle x{=}y\rangle|P)$$

Figure 1: The structural congruence between π_F-process, written $\equiv$, is the smallest equivalence relation satisfying these axioms and closed with respect to contexts

represent variables of the λ-calculus by datums to obtain a direct translation of the λ-calculus into the π_F-calculus.

The definitions of *free* and *bound* names are standard. The *restriction* operator $(\nu x)P$ binds x; x is free in $\langle x\rangle$, $x.P$, $\overline{x}.P$ and in fusions involving x. We write $fn(P)$ to denote the set of free names in P. We use the following abbreviations: $(\nu\vec{x})P \stackrel{\text{def}}{=} (\nu x_1)\ldots(\nu x_n)P$, $\langle\vec{x}\rangle \stackrel{\text{def}}{=} \langle x\rangle_1|\ldots|\langle x\rangle_n$ and $\langle\vec{x}{=}\vec{y}\rangle \stackrel{\text{def}}{=} \langle x_1{=}y_1\rangle|\ldots|\langle x_n{=}y_n\rangle$.

Definition 2.2 *The* structural congruence *between processes, written $\equiv$, is the smallest congruence satisfying the axioms given in Figure 1, and closed with respect to the contexts $\_|\_$, $(\nu x)\_$, $x.\_$ and $\overline{x}.\_$.*

The side-condition on the commutativity of parallel composition allows for processes of arity 0 to be reordered, but not arbitrary processes. For instance,

$$x.P \mid \overline{x}.Q \;\equiv\; \overline{x}.Q \mid x.P \qquad \text{but} \qquad \langle x\rangle|\langle y\rangle|P \not\equiv \langle y\rangle|\langle x\rangle|P.$$

This is essentially the same as in the conventional π-calculus, where processes can be reordered but the names in the concretion $\langle xy\rangle P$ cannot.

The fusion axioms require further explanation. Our intuition is that $\langle x{=}y\rangle$ is an equivalence relation which declares that two names can be used interchangeably. The fusion $\langle x{=}x\rangle$ is congruent to the nil process. So too is $(\nu x)\langle x{=}y\rangle$, since the bound name x is unused. The final six fusion axioms describe small-step substitution, allowing us to deduce $\langle x{=}y\rangle|P \equiv \langle x{=}y\rangle|P\{y/x\}$ and α-conversion. For example,

	$(\nu x)(\overline{x}.\mathbf{nil})$	
$\equiv$	$(\nu x)(\nu y)\big(\langle x{=}y\rangle \mid \overline{x}.\mathbf{nil}\big)$	create fresh bound name y as an alias for x
$\equiv$	$(\nu x)(\nu y)\big(\langle x{=}y\rangle \mid \overline{y}.\mathbf{nil}\big)$	substitute y for x
$\equiv$	$(\nu y)(\overline{y}.\mathbf{nil})$	remove the now-unused bound name x

Honda investigates a simple process framework with equalities on names that are probably the most like our fusion axioms [5]: the axioms are different but the spirit of the equalities is similar. Honda and Yoshida have also introduced π-processes called *equators* [6]. In the asynchronous π-calculus they simulate the effect of explicit fusions; but they do not generalise to the synchronous case [7].

With the structural congruence we can factor out the datums and fusions. In particular, every π_F-process is structurally congruent to one in the *standard form*

$$\langle\vec{u}{=}\vec{v}\rangle \mid (\nu\vec{x})(\langle\vec{y}\rangle \mid P),$$

where the $\vec{x}$s are distinct and contained in the $\vec{y}$s, and P contains no datums or fusions in its top level. We call $\langle\vec{u}{=}\vec{v}\rangle|(\nu\vec{x})(\langle\vec{y}\rangle| \_)$ the *interface* of the process. It is unique in the sense that, given two congruent standard forms

$$\langle\vec{u}_1{=}\vec{v}_1\rangle \mid (\nu\vec{x}_1)(\langle\vec{y}_1\rangle|P_1) \;\equiv\; \langle\vec{u}_2{=}\vec{v}_2\rangle \mid (\nu\vec{x}_2)(\langle\vec{y}_2\rangle|P_2),$$

the fusions $\langle\vec{u}_1{=}\vec{v}_1\rangle$ and $\langle\vec{u}_2{=}\vec{v}_2\rangle$ denote the same equivalence relation on names, $|\vec{x}_1| = |\vec{x}_2|$, and the datums $\vec{y}_1, \vec{y}_2$ are identical and the processes P_1, P_2 structurally congruent up to the name-equivalence and α-conversion of the $\vec{x}$s. We write $E(P)$ for the name-equivalence. It can be inductively defined on the structure of processes, or more simply characterised by $(x, y) \in E(P)$ iff $P \equiv P|\langle x{=}y\rangle$.

We define a symmetric *connection* operator @ between processes of the same arity, which connects them through their interfaces. The effect of the connection $P@Q$ is to fuse together the top-level names in P and Q. If P and Q have standard forms $\langle\vec{u}_1{=}\vec{v}_1\rangle|(\nu\vec{x}_1)(\langle\vec{y}_1\rangle|P_1)$ and $\langle\vec{u}_2{=}\vec{v}_2\rangle|(\nu\vec{x}_2)(\langle\vec{y}_2\rangle|P_2)$ respectively, then

$$P@Q \stackrel{\text{def}}{=} \langle\vec{u}_1\vec{u}_2{=}\vec{v}_1\vec{v}_2\rangle \mid (\nu\vec{x}_1\vec{x}_2)(\langle\vec{y}_1{=}\vec{y}_2\rangle|P_1|P_2),$$

renaming if necessary to avoid name clashes. Because interfaces are unique, the connection operator is well-defined up to structural congruence.

Definition 2.3 *The reaction relation between processes, written* $\searrow$, *is the smallest relation closed with respect to* $\_|\_$, $(\nu x)\_$ *and* $\_ \equiv \_$, *which satisfies*

$$z.P \mid \overline{z}.Q \searrow P@Q.$$

3 Embedding the π-calculus and the Fusion Calculus

The π_F-calculus naturally embeds the π-calculus, the π_I-calculus [11] and the fusion calculus. For the embeddings we consider the fragment of the calculus without summation or replication. The interesting part in the translations concerns the abstractions and concretions:

$$\begin{array}{rcll} (\vec{x})P & \stackrel{*}{\longmapsto} & (\nu\vec{x})(\langle\vec{x}\rangle|P^*) & \textit{Abstraction} \\ (\nu\vec{x})\langle\vec{z}\rangle P & \stackrel{*}{\longmapsto} & (\nu\vec{x})(\langle\vec{z}\rangle|P^*) & \textit{Concretion} \end{array}$$

For example, the π-reaction $z.(x)P \mid \overline{z}.\langle y\rangle Q \searrow_\pi P\{y/x\}|Q$ corresponds to the π_F-reaction

$$\begin{array}{lll} z.(\nu x)(\langle x\rangle|P^*) \mid \overline{z}.(\langle y\rangle|Q^*) & & \\ \quad \searrow_{\pi_F} & (\nu x)(\langle x\rangle|P^*) \;@\; (\langle y\rangle|Q^*) & \\ \quad \equiv & (\nu x)(\langle x{=}y\rangle \mid P^* \mid Q^*) & \text{renaming if necessary} \\ \quad \equiv & (\nu x)(\langle x{=}y\rangle \mid P^*\{y/x\} \mid Q^*) & \text{substituting } y \text{ for } x \\ \quad \equiv & P^*\{y/x\} \mid Q^* & \text{removing unused bound } x \end{array}$$

There is a key difference between the (straightforward) embeddings of the π- and π_I-calculi, and the embedding of the fusion calculus. For the π-calculus, reaction of a π_F-process in the image of $(\_)^*$ necessarily results in a process congruent to one in the image. Even though the reaction temporarily results in a fusion $\langle x{=}y\rangle$, one of those fused names must have arisen from an abstraction $(x)Q$ and so the fusion can be factored away. The same is not true with the fusion calculus. For example,

$$z.(\langle x\rangle|P^*) \mid \overline{z}.(\langle y\rangle|Q^*) \searrow_{\pi_F} \langle x{=}y\rangle \mid P^* \mid Q^*.$$

The process on the left is in the image of the fusion calculus under $(\_)^*$, but the one on the right has an unbounded explicit fusion and so is not. Essentially, because the fusion calculus has unbound input and output processes and yet lacks explicit fusions, it can only allow those reactions that satisfy certain restriction properties on names (given at the end of this section). We do obtain an embedding result in the sense that, by restricting x or y we obtain a π_F-reaction which corresponds to a valid fusion reaction. This embedding result is as strong as can be expected: the fusion reaction requires that a side-condition on restricted names be satisfied; the π_F-reaction does not.

Embedding the π-calculus

We define a translation $(\_)^*$ from π-processes to π_F-processes. We also define a reverse translation $(\_)^o$ and prove embedding results. (The embedding of the π_I-calculus is similar.) Following [8], the set $\mathcal{P}_\pi$ of π-processes is generated by the grammar

$$\begin{array}{lcll} P & ::= & \mathbf{nil} \mid P|P \mid (\nu x)P \mid z.A \mid \overline{z}.C & \textit{Processes} \\ A & ::= & (\vec{x})P & \textit{Abstractions} \\ C & ::= & (\nu\vec{x})\langle\vec{y}\rangle P & \textit{Concretions} \end{array}$$

where the $\vec{x}$s are distinct and, in the concretion, contained in the $\vec{y}$s. The structural congruence on processes and the reaction relation are standard. In order to define the reverse translation $(\_)^o$, we identify the π-*image* in $\mathcal{P}_{\pi_F}$:

$$\begin{array}{lcll} P & ::= & \mathbf{nil} \mid P|P \mid (\nu x)P \mid A & \textit{Processes} \\ A & ::= & z.(\nu\vec{x})(\langle\vec{x}\rangle|P) \mid \overline{z}.(\nu\vec{x})(\langle\vec{y}\rangle|P) & \textit{Input / Output Processes} \end{array}$$

Definition 3.1 *The translation $(\_)^* : \mathcal{P}_\pi \rightarrow \mathcal{P}_{\pi_F}$ is defined inductively by*

$$\begin{array}{rclrcl} (\mathbf{nil})^* & = & \mathbf{nil} & (z.(\vec{x})P)^* & = & z.(\nu\vec{x})(\langle\vec{x}\rangle|P^*) \\ (P|Q)^* & = & P^*|Q^* & (\overline{z}.(\nu\vec{x})\langle\vec{y}\rangle P)^* & = & \overline{z}.(\nu\vec{x})(\langle\vec{y}\rangle|P^*) \\ \big((\nu x)P\big)^* & = & (\nu x)P^* & & & \end{array}$$

The translation $(\_)^o$: π-image $\rightarrow \mathcal{P}_\pi$ is the reverse of this.

Theorem 3.2 *The translations $(\_)^* : \mathcal{P}_\pi \rightarrow \mathcal{P}_{\pi_F}$ and $(\_)^o$: π-image$\rightarrow \mathcal{P}_\pi$ are mutually inverse, preserve the structural congruence, and strongly preserve the reaction relation:*

$$\begin{array}{rlrll} P \in \mathcal{P}_\pi & \textit{and} & P \searrow_\pi Q & \textit{implies} & P^* \searrow_{\pi_F} Q^* \\ P \in \pi\textit{-image} & \textit{and} & P \searrow_{\pi_F} Q & \textit{implies} & P^o \searrow_\pi R \textit{ and } R^* \equiv_{\pi_F} Q \end{array}$$

Embedding the Fusion Calculus

The set of fusion processes $\mathcal{P}_{fu}$ is generated by the grammar

$$P ::= \textbf{nil} \mid P|P \mid (\nu x)P \mid z\vec{x}.P \mid \overline{z}\vec{x}.P.$$

Its structural congruence is standard, and its reaction relation is generated by the rule

$$(\nu\vec{u})(z\vec{x}.P \mid \overline{z}\vec{y}.Q \mid R) \searrow P\sigma \mid Q\sigma \mid R\sigma,$$

where $\text{ran}(\sigma), \text{dom}(\sigma) \subseteq \{\vec{x}, \vec{y}\}$ and $\vec{u} = \text{dom}(\sigma)\backslash\text{ran}(\sigma)$ and $\sigma(v) = \sigma(w)$ if and only if $(v, w) \in E(\langle\vec{x}{=}\vec{y}\rangle)$. The side-conditions describe a natural concept. Consider the equivalence relation generated from the equalities $\vec{x}{=}\vec{y}$. The side-conditions ensure that, for each equivalence class, every element is mapped by σ to a single free witness.

The *fusion-image* of the fusion calculus in the π_F-calculus is similar to that of the π-calculus, but with input and output processes given by

$$A \quad ::= \quad z.(\langle\vec{y}\rangle|P) \ \mid\ \overline{z}.(\langle\vec{y}\rangle|P) \qquad \textit{Input / Output Processes}$$

Translations between the fusion calculus and the fusion-image are straightforward.

Theorem 3.3 *The translations* $(\_)^* : \mathcal{P}_{fu} \to \mathcal{P}_{\pi_F}$ *and* $(\_)^o : \mathcal{P}_{\pi_F} \to \mathcal{P}_{fu}$ *are mutually inverse and preserve structural congruence as in Theorem 3.2. They also preserve reaction in the sense that*

$$P \in \mathcal{P}_{fu} \textit{ and } P \searrow_{fu} Q \ \textit{ implies } \ P^* \searrow_{\pi_F} Q^*$$
$$P \in \textit{fusion-image and } P \searrow_{\pi_F} Q \ \textit{ implies } \ \exists\vec{u}.\ (\nu\vec{u})P \searrow_{fu} R \textit{ and } R^* \equiv_{\pi_F} (\nu\vec{u})Q$$

As discussed, reaction of a process in the fusion-image does not necessarily result in a process also in the fusion-image. Note that the restricted names $\vec{u}$ are precisely those needed to satisfy the side-conditions on reaction in the fusion calculus.

4 Bisimulation for the π_F-calculus

We define a bisimulation relation for the π_F-calculus using a labelled transition system (LTS) in the standard way. The LTS consists of the usual CCS labels $\overline{x}$, x and τ, accompanied by a definition of bisimulation which incorporates fusions:

$$P\mathcal{S}Q : 0 \text{ implies for all } x, y, \text{ if } \langle x{=}y\rangle|P \xrightarrow{\alpha} P_1 \text{ then } \langle x{=}y\rangle|Q \xrightarrow{\alpha} Q_1 \text{ and } P_1\mathcal{S}Q_1.$$

We call this bisimulation the *open bisimulation*, by analogy with open bisimulation for the π-calculus.

In this definition of open bisimulation, labelled transitions are analysed with respect to all possible fusion contexts $\_ \mid \langle x{=}y\rangle$. In fact, we do not need to consider all such contexts. Instead we introduce *fusion transitions*, generated by the axiom

$$x.P \mid \overline{y}.Q \xrightarrow{?x=y} P@Q.$$

The label $?x{=}y$ declares that the process can react in the presence of an explicit fusion $\langle x{=}y\rangle$. Fusion transitions allow us to define bisimulation without having to quantify over fusion contexts. However, the label also declares additional information about the structure of the process. If $P \xrightarrow{?x=y} Q$, then we infer that P must contain input and

$$x.P \xrightarrow{x} P \qquad \overline{x}.P \xrightarrow{\overline{x}} P$$

$$x.P \mid \overline{y}.Q \xrightarrow{?x=y} P@Q \qquad x.P \mid \overline{x}.Q \xrightarrow{\tau} P@Q$$

$$\frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \quad \frac{P \xrightarrow{\alpha} P'}{Q|P \xrightarrow{\alpha} Q|P'} \quad \frac{P \xrightarrow{\alpha} P', \quad x \notin \alpha}{(\nu x)P \xrightarrow{\alpha} (\nu x)Q} \quad \frac{P' \equiv P \xrightarrow{\alpha} Q \equiv Q'}{P' \xrightarrow{\alpha} Q'}$$

Figure 2: *Quotiented* labelled transition system. We do not distinguish between $?x{=}y$ and $?y{=}x$. The final rule closes the LTS with respect to the structural congruence

output processes on unbounded channels x and y. In order to define a bisimulation relation which equals the open bisimulation, we remove this additional information:

$P\mathcal{S}Q$:0 and $P \xrightarrow{?x=y}$ implies either $Q \xrightarrow{?x=y} Q_1$ or $Q \xrightarrow{\tau} Q_1$, and $\langle x{=}y\rangle|P_1\mathcal{S}\langle x{=}y\rangle|Q_1$

The resulting bisimulation equals open bisimulation. A consequence of adding fusion transitions is that we can use standard techniques to prove congruence.

We give two labelled transition systems for the π_F-calculus: a *quotiented LTS* in which we explicitly close the labelled transitions with respect to the structural congruence, and a *structured LTS* in which the labelled transitions are defined according to the structure of processes. These LTSs are equivalent; the quotiented LTS is simpler to understand, and the structured LTS is easier to use. We define corresponding bisimulation relations and prove that they are the same. Finally we use the structured LTS to prove that bisimulation is a congruence.

The Quotiented LTS

The quotiented LTS is given in Figure 2. Notice that the structural congruence rule allows fusions to affect the labels on transitions: for example, the process $\langle x{=}y\rangle|\overline{x}.P$ can undergo the transition $\xrightarrow{\overline{y}}$ as well as $\xrightarrow{\overline{x}}$, because it is structurally congruent to $\langle x{=}y\rangle \mid \overline{y}.P$. We have defined transitions for arbitrary processes instead of just processes of arity 0. This requires two rules for parallel composition, since parallel composition does not commute in the presence of datums.

Proposition 4.1 *$P \searrow Q$ iff $P \xrightarrow{\tau} Q$.*

We now define the bisimulation relation. Our basic intuition is that two processes are bisimilar if and only if they have the same interface and, in all contexts of the form $\_@\langle\vec{y}\rangle$, if one process can do a labelled transition then so can the other. In fact we do not need to consider all such contexts. Instead it is enough to factor out the top-level datums and analyse the labelled transitions for just the processes of arity 0.

Definition 4.2 (Fusion bisimulation) *A symmetric relation $\mathcal{S}$ is a* fusion bisimulation *iff whenever $P\mathcal{S}Q$ then*

1. *$P, Q : m > 0$ implies P and Q have standard forms $\langle\vec{u}{=}\vec{v}\rangle|(\nu\vec{x})(\langle\vec{y}\rangle|P_1)$ and $\langle\vec{u}{=}\vec{v}\rangle|(\nu\vec{x})(\langle\vec{y}\rangle|Q_1)$ respectively and $\langle\vec{u}{=}\vec{v}\rangle|P_1 \ \mathcal{S} \ \langle\vec{u}{=}\vec{v}\rangle|Q_1$;*
2. *$P, Q : 0$ implies they have standard forms $\langle\vec{u}{=}\vec{v}\rangle|P_1$ and $\langle\vec{u}{=}\vec{v}\rangle|Q_1$, and*

$$x.P \xrightarrow{x}_s P \qquad \overline{x}.P \xrightarrow{\overline{x}}_s P \qquad \frac{P \xrightarrow{\alpha_1}_s Q \quad \alpha_1 =_{E(P)} \alpha_2}{P \xrightarrow{\alpha_2}_s Q}\,*$$

$$\frac{P \xrightarrow{x}_s P' \quad Q \xrightarrow{\overline{y}}_s Q'}{P|Q \xrightarrow{?x=y}_s P'@Q'} \qquad \frac{P \xrightarrow{\overline{x}}_s P' \quad Q \xrightarrow{y}_s Q'}{P|Q \xrightarrow{?x=y}_s P'@Q'} \qquad \frac{P \xrightarrow{?x=x}_s Q}{P \xrightarrow{\tau}_s Q}$$

$$\frac{P \xrightarrow{\alpha}_s P'}{P|Q \xrightarrow{\alpha}_s P'|Q} \qquad \frac{P \xrightarrow{\alpha}_s P'}{Q|P \xrightarrow{\alpha}_s Q|P'} \qquad \frac{P \xrightarrow{\alpha}_s Q, \quad x \notin \alpha}{(\nu x)P \xrightarrow{\alpha}_s (\nu x)Q}$$

* We write $\alpha =_{E(P)} \beta$ if α, β are identical up to $E(P)$

Figure 3: *Structured* labelled transition system. This LTS does not include a rule involving the structural congruence. Recall that $E(P)$ is the equivalence relation on names generated by P. A simple characterisation is given by $(x, y) \in E(P)$ if and only if $P \equiv P|\langle x{=}y \rangle$

(a) *if* $P \xrightarrow{\alpha} P'$ *where* α *is* x, $\overline{x}$ *or* τ, *then* $Q \xrightarrow{\alpha} Q'$ *and* $P' \mathcal{S} Q'$

(b) *if* $P \xrightarrow{?x=y} P'$ *then either* $Q \xrightarrow{?x=y} Q'$ *or* $Q \xrightarrow{\tau} Q'$, *and* $\langle x{=}y \rangle | P' \mathcal{S} \langle x{=}y \rangle | Q'$;

3. *similarly for* Q.

Two processes P and Q are *fusion bisimilar*, written $P \sim Q$, if and only if there exists a fusion bisimulation between them. The relation $\sim$ is the largest fusion bisimulation.

Another bisimulation worth exploring is the standard strong bisimulation, which requires that fusion transitions match exactly. This bisimulation is a congruence and contained in the fusion bisimulation. We do not know whether the containment is strict. This question relates to an open problem for the π-calculus without replication or summation, of whether strong bisimulation is closed with respect to substitution.

The Structured LTS

Our goal is to show that the fusion bisimulation in Definition 4.2 is a congruence. However, although the quotiented LTS of Figure 2 is simple due to the presence of the structural congruence rule, the same rule is a problem for proofs. We therefore introduce a *structured* LTS, in which the structural congruence rule is replaced. This structured LTS is ultimately used in Theorem 4.3 to prove that bisimulation is a congruence. The power of the structured LTS is that we can analyse the transition $P \xrightarrow{\alpha}_s Q$ by looking at the structure of P and the label α.

The structured LTS is given in Figure 3. Note the first fusion rule. It allows us to deduce for example that $\langle x{=}y \rangle \mid x.P$ can undergo the transition $\xrightarrow{y}_s$ as well as $\xrightarrow{x}_s$.

We write $\sim_s$ for the bisimulation generated by the structured LTS, defined in the same way as for the quotiented LTS in Definition 4.2.

Theorem 4.3

1. $P \sim_s Q$ *implies* $C[P] \sim_s C[Q]$.

2. $\sim = \sim_s$

From Theorem 4.3 we deduce the main result of this section: that the fusion bisimulation $\sim$ for the quotiented LTS is a congruence.

Towards Full Abstraction for the Fusion Calculus

We believe that hyper-equivalence for the fusion calculus [10] corresponds to open bisimulation for its embedding in the π_F-calculus. The following examples illustrate labelled transitions in the fusion calculus on the left, and the corresponding transitions the π_F-calculus on the right:

$$\begin{array}{rcl@{\qquad}rcl} \overline{u}x.P & \xrightarrow{\overline{u}x}_{fu} & P & \overline{u}.(\langle x\rangle|P^*) & \xrightarrow{\overline{u}}_{\pi_F} & \langle x\rangle|P^* \\ (\nu x)\overline{u}x.P & \xrightarrow{(x)\overline{u}x}_{fu} & P & (\nu x)\overline{u}.(\langle x\rangle|P^*) & \xrightarrow{\overline{u}}_{\pi_F} & (\nu x)(\langle x\rangle|P^*) \\ \overline{u}x.P \mid uy.Q & \xrightarrow{x=y}_{fu} & P \mid Q & \overline{u}.(\langle x\rangle|P^*) \mid u.(\langle y\rangle|Q^*) & \xrightarrow{\tau}_{\pi_F} & \langle x{=}y\rangle \mid P^* \mid Q^* \end{array}$$

First consider the transitions for the fusion calculus. The labels $\overline{u}x$ and $(\nu x)\overline{u}x$ are standard. The label $x{=}y$ states that a fusion has occurred as a consequence of a reaction. Notice that it is not the same as the label $?x{=}y$ in the π_F-calculus, which states that an external fusion must be present for reaction to occur. Now compare the transitions of the fusion calculus with those of the π_F-calculus. The additional information conveyed by a fusion calculus label, is conveyed in the π_F-calculus by the interface of the resulting process.

Victor and Parrow show that hyper-equivalence does not correspond to open bisimulation for the π-calculus [10]. The same result holds for the π_F-calculus with replication. The difference is illustrated by the process $(\nu x)(\overline{u}.(\langle xy\rangle|P))$. In the π-calculus the names x and y can never be substituted for equal names. In the π_F-calculus they can, using the context $\_ \mid u.(\langle zz\rangle)$.

5 Conclusions

Several calculi with name-fusions have recently been proposed. These include the fusion calculus [10], the related chi calculus [1] and the π_I-calculus [11]. In all these calculi the fusions occur *implicitly* in the reaction relation. With the π_F-calculus we have introduced *explicit* fusions. Explicit fusions are processes which can exist in parallel with other processes. They are at least as expressive as implicit fusions. The effect of explicit fusions is described by the structural congruence, not by the reaction relation. The simplicity of the π_F-calculus follows directly from its use of explicit fusions.

We have given embedding results for the π-calculus and the fusion calculus in the π_F-calculus. The embedding for the fusion calculus is weaker than that for the π-calculus. This is to be expected. The π_F-reaction is a local reaction between input and output processes, whose result contains explicit fusions. In contrast, reaction in the fusion calculus has the side-condition that certain names be restricted. The effect of this is to permit only those reactions which do not result in explicit fusions. This is why explicit fusions are not used (or needed) in the fusion calculus.

We have presented a bisimulation congruence for the π_F-calculus. We believe that hyper-equivalence for the fusion calculus is the same as the bisimulation arising from its embedding in the π_F-calculus.

Ongoing Research

Our work on explicit fusions originally arose from a study of process frameworks. We have developed a framework based on the structural congruence studied here [4, 2]. It is related to the action calculus framework of Milner [9, 3]. Explicit fusions allow us to work in a process algebra style, rather than the categorical style used for action calculi.

We are currently exploring an embedding of the λ-calculus in the π_F-calculus. Explicit fusions allow for a translation that is purely compositional, unlike the analogous translations into the π-calculus and fusion calculus. It remains further work to relate behavioural congruence for the λ-calculus with the bisimulation arising from its embedding in the π_F-calculus.

Acknowledgements

We thank Peter Sewell, Robin Milner and the anonymous referees. Gardner is supported by an EPSRC Advanced Fellowship, and Wischik by an EPSRC Studentship.

References

[1] Y. Fu. Open bisimulations on chi processes. In *CONCUR*, LNCS 1664. Springer, 1999.

[2] P. Gardner. From process calculi to process frameworks. In *CONCUR*, 2000. To appear.

[3] P. Gardner and L. Wischik. Symmetric action calculi (abstract). Manuscript online.

[4] P. Gardner and L. Wischik. A process framework based on the π_F calculus. In *EXPRESS*, volume 27. Elsevier Science Publishers, 1999.

[5] K. Honda. Elementary structures in process theory (1): sets with renaming. *Mathematical Structures in Computer Science*. To appear.

[6] K. Honda and N. Yoshida. On reduction-based process semantics. In *Foundations of Software Technology and Theoretical Computer Science*, LNCS 761. Springer, 1993.

[7] M. Merro. On equators in asynchronous name-passing calculi without matching. In *EXPRESS*, volume 27. Elsevier Science Publishers, 1999.

[8] R. Milner. *Communicating and mobile systems: the pi calculus*. CUP, 1999.

[9] Robin Milner. Calculi for interaction. *Acta Informatica*, 33(8), 1996.

[10] J. Parrow and B. Victor. The fusion calculus: expressiveness and symmetry in mobile processes. In *LICS*. IEEE, Computer Society Press, 1998.

[11] D. Sangiorgi. Pi-calculus, internal mobility and agent-passing calculi. *Theoretical Computer Science*, 167(2), 1996.

[12] D. Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33, 1996.

[13] B. Victor and J. Parrow. Concurrent constraints in the fusion calculus. In *ICALP*, LNCS 1443. Springer, 1998.

[14] L. Wischik. 2001. Ph.D. thesis. In preparation.