Explicit Fusions

Lucian Wischik[1] and Philippa Gardner[2]

[1] University of Bologna, Italy. lu@wischik.com
[2] Imperial College, London. pg@doc.ic.ac.uk

Abstract. We introduce *explicit fusions* of names. An explicit fusion is a process that exists concurrently with the rest of the system and enables two names to be used interchangeably. Explicit fusions provide a small-step account of reaction in process calculi such as the pi calculus and the fusion calculus. In this respect they are similar to the explicit substitutions of Abadi, Cardelli and Curien, which do the same for the lambda calculus. In this paper, we give a technical foundation for explicit fusions. We present the pi-F calculus, a simple process calculus with explicit fusions, and define a strong bisimulation congruence. We study the embeddings of the fusion calculus and the pi calculus. The former is fully abstract with respect to bisimulation.

1 Overview

We introduce *explicit fusions* of names. To 'fuse' two names is to declare that they may be used interchangeably. An explicit fusion is a process that exists concurrently with the rest of the system and enables the interchange. We start by outlining three uses for explicit fusions.

(1) It is important to be able to tell whether two pieces of code have the same effect in all contexts – if they do, then they can be substituted for each other. Typically we require a piece of code to function correctly even in a context where two pointers happen to point to the same object. See for example Figure 1. A context with two co-referring pointers can be represented by an explicit fusion, in the sense that it allows either pointer to be used, interchangeably, at any time.

(2) Substitution, although conceptually simple, can be difficult to implement. In the lambda calculus, Abadi and Cardelli [1] found it useful to implement the substitutive effect of β-reduction with a series of smaller steps involving *explicit substitutions*. In this paper we use explicit fusions to give a small-step account of reaction in concurrent calculi – in particular, of reaction in the pi calculus and in the fusion calculus [23]. For the fusion calculus, explicit fusions allow us to define a local small-step reaction between a single input and output atom. Without explicit fusions, reaction in the fusion calculus requires that the entire scope of the fusion's effect first be taken into account. Choosing an appropriately small step is particularly important for a distributed calculus: if an operation is complex it might fail part way through, leaving the system in a half-way state; but if all operations are small enough to be performed atomically, then there will be no half-way states.

```
fun swap x y = (
  x := !x xor !y;
  y := !x xor !y;
  x := !x xor !y );
```

Fig. 1: This ML implementation of 'swap' is intended to swap its arguments, using a bitwise exclusive-or operator to save space, but it fails in a context where x and y point to the same thing. Debugging is left as an (easy) exercise for the reader.

(3) In a distributed system an object might move from one location to another, and yet we still need to send it messages. If we at least know the previous location of the object, we can use a solution that is *distributed* and *asynchronous*: send a message and, if the object had moved, then the message gets forwarded. Abstracting away from details of implementation, the overall effect is a fusion of the object's previous and current location – in the sense that we can refer to either, interchangeably. This 'forwarding' technique is used for instance in a recent implementation of the Ambient Calculus [5]. It may also be appropriate for internet over mobile phones: when a phone moves to a new location, other parties still know its previous location. (A different *centralised synchronous* solution, used for instance in CORBA and treated theoretically by Sewell and Unyapoth [27], is to have a central naming service to mediate all object migration.) A related situation is when multiple parties wish to interact over the same channel even when they are physically remote. Explicit fusions are used for this in a distributed implementation of the pi calculus [8] and in the ongoing 'Highwire' project at Microsoft.

This paper is a full version of an earlier article by the authors [11]. Some of the material also appeared in the doctoral dissertation of one of the authors [31].

A Process Calculus With Explicit Fusions

This paper develops a simple process calculus, the pi-F calculus, which has explicit fusions. Key results are embeddings of the pi calculus and the fusion calculus into the pi-F calculus. These show that explicit fusions are expressive enough to describe both the name-substitution that occurs in pi reaction, and the fusions that occur in fusion reaction. To set the pi-F calculus in place we first give a brief survey of related calculi.

The reader is assumed to be familiar with the pi calculus. It consists of concurrent processes which may perform outputs or inputs. If one process wishes to output on a channel, and another wishes to input on the same channel, then they can react together and transmit some names as part of that reaction:

$$\overline{u}.\langle y\rangle P \mid u.(x)Q \ \searrow_{\pi}\ P \mid Q\{y/x\}.$$

This reaction has an asymmetry not present in CCS: input binds a name, but output does not. Sangiorgi challenged this asymmetry by introducing the private pi calculus [24], in which both input and output bind their names. Reaction between input and output is no longer a directed 'send' of names; instead, we might say it is a bound symmetric fusion of the names. He concludes that reaction between bound names accounts for much of the expressivity of the pi calculus.

The other way to make the pi calculus symmetric, used in this paper as well as in the fusion calculus of Victor and Parrow [23,28] and the chi calculus of Fu [6], is to make both input and output non-binding. In addition to the goal of symmetry, the fusion calculus was also motivated by a desire to express concurrent constraints, and the chi calculus by the similarity between cut-elimination and reaction. A surprising result about non-binding input and output is due to Laneve and Victor [17]: their purely asynchronous 'solos' calculus (which has no continuations after input *or* output) is fully as expressive as a synchronous calculus. This adds weight to the suggestion that non-binding input and output are fundamental.

Given a reaction between non-binding input and output, one must chose how to write the result. In this paper we use explicit fusions:

$$\overline{u}.\langle y\rangle P \mid u.\langle x\rangle Q \mid R \quad \searrow_F \quad x{=}y \mid P \mid Q \mid R.$$

The reaction in this example is a local one between the input and output processes. But the effect of the resulting fusion $x{=}y$ is global in scope: x and y can be used interchangeably throughout the entire process, including R. (We account for this interchange effect through *structural congruence* $\equiv_F$ rather than a reaction step $\searrow$.) To limit the scope of the fusion, we use restriction. For example, restricting x in the above expression we obtain

$$(\nu x)(x{=}y \mid P \mid Q \mid R) \quad \equiv_F \quad P\{y/x\} \mid Q\{y/x\} \mid R\{y/x\}.$$

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

Neither the fusion calculus nor the chi calculus have explicit fusions in their syntax. They therefore cannot include the above reaction, and instead require that that we find an enclosing restriction of either x or y: for instance,

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

The x and y are fused during the reaction, but the restricted fusion is immediately turned into a substitution. If we had restricted y rather than x, then the substitution would have been $\{x/y\}$. The full polyadic reaction rule, using many $\widetilde{x}$s and $\widetilde{y}$s, is more complicated; it is given in Definition 14. Note that the reaction here is not a local one between output and input, but instead requires a global search for enclosing restrictions. (See the conclusions for further discussion on this point.)

Other Related Work

Honda and Yoshida have highlighted certain pi processes called *equators* [14]. These simulate the effect of explicit fusions in the asynchronous pi calculus [18], but they do not generalise to the synchronous calculus. Honda also investigates a simple process framework [15] with equalities on names that are probably the most like our fusion axioms; the axioms are different but the spirit of the equalities is similar.

The ρ-calculus [22] is a concurrent-constraint calculus incorporating pi calculus processes with name-equality constraints. Victor and Parrow [28] have shown how to encode the ρ-calculus into the fusion calculus. In fact, its concurrent constraints are closer in spirit to explicit fusions in the pi-F calculus than to the fusions implicit in fusion reaction.

Plan of Paper

In the first half of this paper we introduce the pi-F calculus. Section 2 gives its syntax and reaction relation. Section 3 gives its labelled transition semantics and a strong bisimulation congruence. In the second half of the paper we compare it to existing calculi. Section 4 gives embedding results for the fusion calculus with respect to the strong congruence; Section 5 gives embedding results for the pi calculus with respect to reaction.

2 The pi-F calculus

We now present the pi-F calculus. We choose to define it using the 'commitment' style of Milner [21]. The intention is that the two parts of a communication – a commitment to communicate, followed by the exchange of names – are represented by separate constructs in the language. This choice leads to a simpler labelled transition system.

Milner uses output $\overline{x}$ and input x for the commitment, and introduces new types of process to describe the exchange: a 'concretion process' ready to send its names, an 'abstraction process' ready to receive the names, and a derived 'application operator' @ which consummates the commitment with a substitution of names. In the pi-F calculus we have chosen instead to augment the language of processes by adding *datums* $\langle x \rangle$, which are the names ready to be communicated; the commitment is consummated with an explicit fusion of these names. The following table illustrates how concretions and abstractions are represented with datums.

$\overline{u}.\langle x \rangle P$	*pi process*	$\overline{u}.(\langle x \rangle \vert P)$	*pi-F process*
$\langle x \rangle P$	*pi concretion*	$\langle x \rangle \vert P$	*pi-F process*
$(x)P$	*pi abstraction*	$(\nu x)(\langle x \rangle \vert P)$	*pi-F process*

$$\langle y \rangle P \mathbin{@} (x)Q \;=\; P \mid Q\{y/x\} \qquad (\langle y \rangle | P) \mathbin{@} (\nu x)(\langle x \rangle | Q) \;=\; (\nu x)(y{=}x \mid P \mid Q)$$

Note that, just as the names in a concretion or abstraction cannot be re-ordered, neither can datums: $\langle x \rangle | \langle y \rangle | P$ is not equivalent to $\langle y \rangle | \langle x \rangle | P$. The use of datums was first introduced in Milner's action calculus framework [20]. The results in this paper do not depend on them.

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

Definition 1 (Syntax) *The set $\mathcal{P}_F$ of* processes *of the pi-F calculus is*

$$
\begin{array}{rcll}
P & ::= & \mathbf{0} & \textit{Null process} \\
& \mid & P \mid P & \textit{Parallel composition} \\
& \mid & !P & \textit{Replication} \\
& \mid & (\nu x)P & \textit{Scope restriction} \\
& \mid & \overline{x}.P & \textit{Output action} \\
& \mid & x.P & \textit{Input action} \\
& \mid & \langle x \rangle & \textit{Datum} \\
& \mid & x{=}y & \textit{Fusion}
\end{array}
$$

Contexts are given by $E \ ::= \ \_ \mid P|E \mid E|P \mid !E \mid (\nu x)E \mid \overline{x}.E \mid x.E$. (The E stands for 'environment'; we avoid the letter C which is used for concretions).

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 would also be possible, similar perhaps to the sorting discipline for the pi calculus [21]. Replication denotes an unbounded number of copies of a processes. It is only defined on process of arity zero. This is because non-zero arity processes have datums (or 'wiring'), and it does not make sense to have unbounded wiring in a term.

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

Definition 2 *The* structural congruence *between processes, written $\equiv$, is the smallest congruence satisfying the axioms given in Figure 2, and closed with respect to contexts (ie. if $P \equiv Q$ then $E[P] \equiv E[Q]$ for all E).*

We now comment on some of the axioms for structural congruence. Our intuition is that explicit fusions give rise to an equivalence relation on names (Definition 4). This is the origin of the three axioms for the reflexivity, symmetry and transitivity of fusions; the subtraction axiom allows names to be removed from the equivalence relation via restriction. We sometimes write an explicit fusion ϕ instead of $\widetilde{x}{=}\widetilde{y}$ when it is not important which particular names are fused.

In the introduction we defined an explicit fusion as something that exists concurrently, and that interchanges names. The final axioms perform this interchange in small steps. Because the fusion is not consumed by substitution, Standard axioms for $|$ and $\mathbf{0}$ and $!$:

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

Standard scope axioms:

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

$(\nu x)(P|Q) \equiv (\nu x)P \mid Q$ if $x \notin \mathrm{fn}(Q)$

Fusion axioms:		Small-step substitution:
$x{=}x \equiv \mathbf{0}$	*Reflexivity*	$x{=}y \mid x.P \equiv x{=}y \mid y.P$
$x{=}y \equiv y{=}x$	*Symmetry*	$x{=}y \mid \overline{x}.P \equiv x{=}y \mid \overline{y}.P$
$x{=}y \mid y{=}z \equiv x{=}z \mid y{=}z$	*Transitivity*	$x{=}y \mid z.P \equiv x{=}y \mid z.(x{=}y\|P)$
$(\nu x)(x{=}y) \equiv \mathbf{0}$	*Subtraction*	$x{=}y \mid \overline{z}.P \equiv x{=}y \mid \overline{z}.(x{=}y\|P)$
		$x{=}y \mid \langle x \rangle \equiv x{=}y \mid \langle y \rangle$

Fig. 2: The structural congruence between pi-F process, written $\equiv$, is the smallest equivalence relation satisfying these axioms and closed with respect to contexts.

its continued existence is ensured. We can use the small-step interchange to deduce a large-step capture-avoiding substitution $x{=}y|P \equiv x{=}y|P\{y/x\}$ along with α-conversion. For example,

$$\begin{aligned} & (\nu x)(\overline{x}.\mathbf{0}) \\ \equiv\ & (\nu x)(\nu y)\big(\, x{=}y \mid \overline{x}.\mathbf{0}\,\big) && \textit{create fresh bound name } y \textit{ as an alias for } x \\ \equiv\ & (\nu x)(\nu y)\big(\, x{=}y \mid \overline{y}.\mathbf{0}\,\big) && \textit{substitute } y \textit{ for } x \\ \equiv\ & (\nu y)(\overline{y}.\mathbf{0}) && \textit{remove the now-unused bound name } x \end{aligned}$$

When an output and an input react together, the result is a fusion of their datums. We define reaction (Definition 3) in terms of a 'connection' operator @ between processes; this is a symmetric generalisation of Milner's application operator. The definition of the connection operator first requires that all the datums be factored out into what we call the *interface* of a process (Definition 6). And the definition of the interface first requires a definition of the equivalence relation $E(P)$ generated by a process (Definition 4). So as not to lose sight of the end goal, we state it first.

Definition 3 *The reaction relation $\searrow$ between processes is the smallest relation closed with respect to $\_ \mid \_$, $(\nu x)\_$ and $\_ \equiv \_$, which satisfies*

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

for P and Q of the same arity.

The rest of this section leads to a formal definition of the @ operator.

Definition 4 (Equivalence relation) *The equivalence relation $E(P)$ generated by the pi-F process P is as follows:*

$$E(P|Q) \quad = \quad E(P) \uplus E(Q) \qquad \textit{equivalence-closed union}$$

$$
\begin{array}{rcll}
E(\,(\nu x)P\,) & = & E(P)\backslash x & \textit{removing name from equivalence class} \\
E(x{=}y) & = & \boldsymbol{I} \uplus (x,y) & \textit{smallest equivalence containing } x{=}y \\
E(!P) & = & E(P) & \textit{replication doesn't affect fusion} \\
E(\_) & = & \boldsymbol{I} & \textit{otherwise, the identity relation}
\end{array}
$$

We write $P \vdash x{=}y$ *if* $(x,y) \in E(P)$.

The equivalence relation $E(P)$ fully characterises the explicit fusions in structural congruence:

Lemma 5 $x{=}y|P \equiv P$ *if and only if* $(x,y) \in E(P)$.

Proof. In the forward direction, first prove that $P \equiv Q$ implies $E(P) = E(Q)$ by induction on the derivation of $P \equiv Q$ (Figure 2). Hence if $x{=}y|P \equiv P$ then $E(P) \uplus (x,y) = E(P)$, and hence $(x,y) \in E(P)$.

The reverse direction is by induction on the structure of P. We give the two interesting cases. For replication, we assume $(x,y) \in E(!P)$. This must have been deduced from $(x,y) \in E(P)$. Using the induction hypothesis, $P \equiv x{=}y|P$. Now $!P \equiv P|!P \equiv x{=}y|P|!P \equiv x{=}y|!P$. For the parallel case, assume $(x,y) \in E(P|Q)$. This must have been deduced from a finite chain $(x,z_1) \in E(P)$, $(z_1,z_2) \in E(Q)$, $\ldots(z_n,y) \in E(P)$. Apply the induction hypothesis to each element in the chain to get $P|Q \equiv x{=}z_1|\ldots|z_n{=}y \mid P|Q$. The result follows directly. □

Definition 6 (Interface) *It is possible to factor out the datums from a process. In particular, every pi-F process is structurally congruent to one in the* standard form

$$(\nu\widetilde{x})(\,\langle\widetilde{y}\rangle \mid P\,)$$

where the $\widetilde{x}$ *are distinct and contained in the* $\widetilde{y}$*, and* P*'s top level contains no further datums nor any fusions involving any* $x \in \widetilde{x}$*. We write a standard form* $(\nu\widetilde{x})(\langle\widetilde{y}\rangle|P)$ *as* $I{\cdot}P$ *where* $I = (\nu\widetilde{x})(\langle\widetilde{y}\rangle|\_)$*. We call context* I *the* interface *and process* P *the* contents.

The interface effectively factors out the 'concretion' part of a process. Interfaces are unique up to α-conversion and $E(P)$. For example, consider the term

$$(\nu xyz)(\langle xyu\rangle \mid y{=}v \mid u{=}u' \mid P).$$

Standard form requires that the outermost restricted names be contained in the datums; hence z must be pushed inside using structural congruence. It also requires that they not be fused; hence y must be pushed inside. This means that the standard form has minimal outermost restrictions, leading to the uniqueness of its interface (up to alpha-renaming and free fusions):

$$\equiv \quad (\nu x)(\langle xvu\rangle \mid u{=}u' \mid (\nu z)P\,) \quad \equiv \quad (\nu x')(\langle x'vu'\rangle \mid u{=}u' \mid (\nu z)P\,).$$

Moreover, because of the stipulation that $E(P)$ fuses no bound names, we ensure that the content P of the $I{\cdot}P$ is also unique. Specifically, given two congruent standard forms

$$(\nu\widetilde{x}_1)(\langle\widetilde{y}_1\rangle \mid P_1) \quad\equiv\quad (\nu\widetilde{x}_2)(\langle\widetilde{y}_2\rangle \mid P_2)$$

then there exist names $\widetilde{x}$ and substitutions $\sigma_1 : \widetilde{x}_1{\to}\widetilde{x}$ and $\sigma_2 : \widetilde{x}_2{\to}\widetilde{x}$ such that $\sigma_1\widetilde{x}_1 = \widetilde{x}$ and $\sigma_2\widetilde{x}_2 = \widetilde{x}$ and $P_1 \vdash \sigma_1\widetilde{y}_1 = \sigma_2\widetilde{y}_2$ and $\sigma_1 P_1 \equiv \sigma_2 P_2$. Note that $E(P_1) = E(P_1\sigma)$, from the assumption that P_1 fuses no names in $\widetilde{x}_1$.

We use standard forms to define the connection operator @ between processes of the same arity. Assume two processes P_1 and P_2 with standard forms $(\nu\widetilde{x}_1)(\langle\widetilde{y}_1\rangle|Q_1)$ and $(\nu\widetilde{x}_2)(\langle\widetilde{y}_2\rangle|Q_2)$ respectively. Then P_1 @ P_2 is defined (only up to structural congruence) by

$$P_1 \mathbin{@} P_2 \quad=\quad (\nu\widetilde{x}_1\widetilde{x}_2)(\widetilde{y}_1{=}\widetilde{y}_2 \mid Q_1 \mid Q_2)$$

Observe that $P_1@P_2 \equiv P_2@P_1$.

We have described above how datums may be factored out of a process. Should one wish also to factor out the fusions from a process, some additional axioms are required. These were first introduced by Engelfriet [4] for the pi calculus, to prove decidability of the structural congruence.

$$!(P|Q) \equiv\, !P \mid !Q \qquad !!P \equiv\, !P \qquad !x{=}y \equiv x{=}y$$

These axioms all express the intuition that $!P$ represents an unlimited number of copies of P. The final axiom is actually a generalisation of the more usual $!\mathbf{0} \equiv \mathbf{0}$. With these additions, every process P is structurally congruent to another process $\phi \mid P_1$ with $E(\phi) = E(P)$ and $E(P_1) = \mathbf{I}$.

3 Bisimulation for the pi-F calculus

We now define a strong *bisimulation congruence* for the pi-F calculus. This is a standard technique for judging whether two processes have the same behaviour in all contexts, based on whether they make the same observable (labelled) transitions at each stage of their reduction. We choose to use CCS-style labels $\overline{x}$ and x (for a commitment to send or receive), and τ (for a commitment to a particular internal action). We also choose to give an *open* style of bisimulation [25]. 'Open' traditionally means that the bisimulation relation is closed with respect to substitutions; but in our setting it is more natural to close with respect to explicit fusions. Given P and Q with zero arity,

$$P\ \mathcal{S}\ Q \text{ implies for all } x, y, \text{ if } x{=}y|P \xrightarrow{\alpha} P' \text{ then } x{=}y|Q \xrightarrow{\alpha} Q' \text{ and } P'\ \mathcal{S}\ Q'. \quad (1)$$

It is standard from the pi calculus that open bisimulation generates a congruence; we prove the same result for the pi-F calculus (Theorem 13). Surprisingly, and in contrast to the pi calculus, this congruence for the pi-F calculus is the largest that is contained in bisimulation [29]. This means that closing with respect to explicit fusions is in fact equivalent to closing with respect to arbitrary contexts. We return to this point in the conclusions.

Equation 1 above has an infinite quantification over fusion contexts. In fact, we do not need to consider all such contexts. Instead we introduce *fusion transitions*, generated by the axiom

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

The label $?x{=}y$ declares that the process can react in the presence of the specific explicit fusion $x{=}y$. Fusion transitions allow us to define bisimulation without having to quantify over fusion contexts. However, if we were to require that a fusion transition $P \xrightarrow{?x=y} P'$ implies $Q \xrightarrow{?x=y} Q'$, the resulting bisimulation would be stronger than Equation 1. That is because $Q \xrightarrow{?x=y} Q'$ not only declares that Q can react in a context $x{=}y$ as required, but also implies that Q contains input and output processes on free channel names x and y. We therefore remove the implication:

$$P \mathrel{\mathcal{S}} Q \text{ and } P \xrightarrow{?x=y} P' \text{ implies } x{=}y|Q \xrightarrow{\tau} Q' \text{ and } x{=}y|P' \mathrel{\mathcal{S}} Q'.$$

This equation is now equivalent to Equation 1. The above technique of avoiding quantification over substitutions is known as 'symbolic' bisimulation [13].

In fact, the strong bisimulation that does not remove the implication (ie. that requires the fusion transitions to match exactly) is also interesting. It 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 pi calculus without replication or summation, of whether strong bisimulation is closed with respect to substitution.

The remainder of this section is devoted to proving that bisimulation is a congruence. We give two labelled transition systems for the pi-F calculus: a *quotiented LTS* in which we explicitly close the labelled transitions with respect to the structural congruence, and a *structural 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 since a term's transitions $P \xrightarrow{\alpha} P'$ can be deduced simply by induction on the structure of P. The fusion transitions are necessary for this structured LTS. We use the structured LTS to prove that bisimulation is a congruence.

The Quotiented LTS

The quotiented LTS is given in Figure 3. The technique of quotienting by structural congruence was first used for the pi calculus in [19], inspired by the Chemical Abstract Machine of Berry and Boudol [2]. Notice that the structural congruence rule allows fusions to affect the labels on transitions: for example, the process $x{=}y \mid \overline{x}.P$ can undergo the transition $\xrightarrow{\overline{y}}$ as well as $\xrightarrow{\overline{x}}$, because it is structurally congruent to $x{=}y \mid \overline{y}.P$.

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

$$\frac{P \xrightarrow{\alpha} P' \quad Q:0}{P \mid Q \xrightarrow{\alpha} P' \mid Q} \qquad \frac{P \xrightarrow{\alpha} P' \quad x \notin \alpha}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'} \qquad \frac{Q \equiv P \xrightarrow{\alpha} P' \equiv Q'}{Q \xrightarrow{\alpha} Q'}$$

Fig. 3: *Quotiented* labelled transition system for processes of arity 0. The labels $?x{=}y$ and $?y{=}x$ are equivalent.

Proposition 7 $P \xrightarrow{\tau} Q$ *if and only if* $P \searrow Q$ *and* $P : 0$.

Proof. The rules for deducing $\searrow$ are analogous to those for deducing $\xrightarrow{\tau}$. $\square$

We now define bisimulation. Our intuition is that two processes should be considered bisimilar if and only if they have the same interface and if one process can do a labelled transition then so can the other.

Definition 8 (Fusion bisimulation) *A symmetric relation* $\mathcal{S}$ *is a* fusion bisimulation *if and only if whenever* $P\ \mathcal{S}\ Q$ *then either* $P, Q : 0$ *and*

1. $P \xrightarrow{\alpha} P'$ *implies* $Q \xrightarrow{\alpha} Q'$ *and* $P'\ \mathcal{S}\ Q'$ *for labels* $\alpha \in \{\overline{x}, x, \tau\}$
2. $P \xrightarrow{?x{=}y} P'$ *implies* $x{=}y \mid Q \xrightarrow{\tau} Q'$ *and* $x{=}y \mid P'\ \mathcal{S}\ Q'$
3. $E(P) = E(Q)$

or $P, Q : m > 0$ *and there exists a common interface* I *such that* P *and* Q *have standard forms* $I{\cdot}P_1$ *and* $I{\cdot}Q_1$ *respectively, and* $P_1\ \mathcal{S}\ Q_1$.

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

Fusion bisimulation is defined on processes of arbitrary arities, just as Milner defined bisimulation on abstractions and concretions as well as processes [21]. For the abstraction $(x)P$ he additionally considers all $P\{y/x\}$. We do not have to, since the job of quantifying over all fusions is already performed by Part 2 of the definition. Part 2 expresses clearly our intentions for the $\xrightarrow{?x{=}y}$ label: 'If P can react in the presence of a fusion $x{=}y$, then so can Q.'

We remark that in general a term P has infinitely many reactions $P \xrightarrow{\alpha} P'$. For instance, $!\overline{u} \xrightarrow{\overline{u}} !\overline{u}$ but also $\xrightarrow{\overline{u}} \mathbf{0}|!\overline{u}$ and $\xrightarrow{\overline{u}} \overline{u}|\overline{u}|!\overline{u}$. However, the image P under transitions $\xrightarrow{\alpha}$ is always finite up to structural congruence. Labelled transitions and fusion bisimilarity are only defined up to structural congruence.

The Structured LTS

Our goal is to show that the fusion bisimulation in Definition 8 is a congruence. However, although the quotiented LTS of Figure 3 is simple to define thanks to the presence of the structural congruence rule, the same rule is awkward for proofs. In particular, a bisimulation proof normally assumes some particular $P \xrightarrow{\alpha} P'$ and deduces a corresponding $Q \xrightarrow{\alpha} Q'$. Enumerating all possible

$$\overline{x}.P \xrightarrow{\overline{x}} P \qquad x.P \xrightarrow{x} P \qquad \frac{P \xrightarrow{\alpha} P' \quad x \notin \alpha}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'}$$

$$\frac{P \xrightarrow{\overline{x}} P' \quad Q \xrightarrow{y} Q'}{P|Q \xrightarrow{?x{=}y} P'@Q'} \qquad \frac{P \xrightarrow{\alpha} P' \quad Q}{P|Q \xrightarrow{\alpha} P'|Q} \qquad \frac{P \mid !P \xrightarrow{\alpha} P' \quad P}{!P \xrightarrow{\alpha} P'}$$

$$\frac{P \xrightarrow{?x{=}x} Q}{P \xrightarrow{\tau} Q} \qquad \frac{P \xrightarrow{\alpha} Q \quad P \vdash \alpha = \beta}{P \xrightarrow{\beta} Q} \qquad \frac{P \xrightarrow{\alpha} P' \equiv P_1'}{P \xrightarrow{\alpha} P_1'}$$

Fig. 4: *Structured* labelled transition system. The left hand sides are all assumed to have arity zero. The parallel rules have mirror cases, which we have omitted. Recall from Definition 9 that $P \vdash \alpha = \beta$ means that P contains sufficient explicit fusions to interchange α and β. Note that no rule has a structural congruence on its left hand side.

transitions of P is awkward in the quotiented LTS because P is quotiented by structural congruence – if $P \equiv P_1 \xrightarrow{\alpha} P'$ then $P \xrightarrow{\alpha} P'$ – and so this requires an additional induction on the derivation of $P \equiv P_1$.

We therefore introduce a *structured* LTS. This describes exactly the same labelled transition system (Corollary 12), and so generates the same bisimulation relation $\sim$ as the quotiented LTS. However, the structured LTS has no structural congruence on its left hand side, thereby allowing us to analyse each labelled transition according only to the label and the structure of the process (Lemma 10). This analysis is used in Theorem 13 to prove that bisimulation is a congruence. The structured LTS is also defined only on processes of arity 0.

The structured LTS is given in Figure 4. The non-standard rules are those for fusions. Two rules are needed to introduce the fusion label $\xrightarrow{?x{=}y}$, according to whether the output came from the left or the right side of the parallel composition. These are analogous to the τ-rules for the pi calculus. And given an identity fusion transition $P \xrightarrow{?x{=}x} Q$ we can then deduce the τ-transition $P \xrightarrow{\tau} Q$. We also use the notation $P \vdash \alpha = \beta$ to indicate that P contains sufficient explicit fusions to interchange the labels α and β. This generalises the notation $P \vdash x{=}y$ given in Definition 4.

Definition 9 (Label equality) *$P \vdash \tau = \tau$ and $P \vdash ?x{=}y = ?y{=}x$ and*

$$\begin{array}{ll} P \vdash x = y & \text{if } (x,y) \in E(P) \\ P \vdash \overline{x} = \overline{y} & \text{if } (x,y) \in E(P) \\ P \vdash ?x{=}y = ?u{=}v & \text{if } (x,u) \in E(P) \text{ and } (y,v) \in E(P) \end{array}$$

If a process undergoes a particular labelled transition, then it can undergo any other *equal* labelled transition. For example the process $x{=}y \mid x.P$ can undergo the transition $\xrightarrow{y}$ as well as $\xrightarrow{x}$.

For the following results we distinguish the structured LTS from the quotiented LTS by writing $\longrightarrow_s$ for its transitions. But since the two LTSs turn out to be exactly the same (Corollary 12) the distinction is not generally required.

Lemma 10 *If we have a transition $P \xrightarrow{\alpha}_s P'$, then the transition is in fact one of the following:*

$$\begin{array}{ll}
\overline{x}.Q \xrightarrow{\overline{x}}_s \equiv Q & \\
x.Q \xrightarrow{x}_s \equiv Q & \\
(\nu z)Q \xrightarrow{\alpha}_s \equiv (\nu z)Q' & \textit{with } Q \xrightarrow{\beta}_s Q',\ Q \vdash \alpha{=}\beta,\ z \notin \alpha \\
!Q \xrightarrow{\alpha}_s \equiv Q' \mid !Q & \textit{with } Q \xrightarrow{\beta}_s Q',\ Q \vdash \alpha{=}\beta \\
!Q \xrightarrow{\tau}_s \equiv Q'@Q'' \mid !Q & \textit{with } Q \xrightarrow{\overline{u}}_s Q',\ Q \xrightarrow{u} Q'' \\
!Q \xrightarrow{?x{=}y}_s \equiv Q'@Q'' \mid !Q & \textit{with } Q \xrightarrow{\overline{u}}_s Q',\ Q \xrightarrow{v} Q'',\ Q \vdash ?x{=}y = ?u{=}v \\
Q_1 \mid Q_2 \xrightarrow{\alpha}_s \equiv Q_1' \mid Q_2 & \textit{with } Q_1 \xrightarrow{\beta}_s Q_1',\ Q_1|Q_2 \vdash \alpha{=}\beta,\ \textit{or vice versa} \\
Q_1 \mid Q_2 \xrightarrow{\tau}_s \equiv Q_1'@Q_2' & \textit{with } Q_1 \xrightarrow{\overline{u}}_s Q_1',\ Q_2 \xrightarrow{u} Q_2',\ \textit{or vice versa} \\
Q_1 \mid Q_2 \xrightarrow{?x{=}y}_s \equiv Q_1'@Q_2' & \textit{with } Q_1 \xrightarrow{\overline{u}}_s Q_1',\ Q_2 \xrightarrow{v} Q_2',\ \textit{or vice versa} \\
& \qquad Q_1|Q_2 \vdash ?x{=}y = ?u{=}v
\end{array}$$

Proof. For most processes and transitions, the proof involves a simple case analysis. For replication, the proof is by induction on the derivation of the transition.□

Lemma 11 $P \equiv P_1 \xrightarrow{\alpha}_s P'$ *implies* $P \xrightarrow{\alpha}_s P'$.

Proof. By a lengthy induction on the derivation of structural congruence. For every rule in the structural congruence, we use Lemma 10 to analyse every possible transition taken by each side of the rule. □

Proposition 12 $P \xrightarrow{\alpha} P'$ *if and only if* $P \xrightarrow{\alpha}_s P'$

Proof. In the forward direction, by induction on the derivation of $P \xrightarrow{\alpha} P'$. Most of the rules of the quotiented LTS (Figure 3) are present in the structured LTS (Figure 4). The only one not present is structural congruence, when $P \xrightarrow{\alpha} P'$ is deduced from $P \equiv Q \xrightarrow{\alpha} Q' \equiv P'$. By the induction hypothesis, $Q \xrightarrow{\alpha}_s Q'$. The structured LTS already provides for $\equiv$ on its right hand side, and Lemma 11 provides for the left side, yielding $P \xrightarrow{\alpha}_s P'$ as required.

The reverse direction first needs the straightforward lemma (for the quotiented LTS) that if $P \xrightarrow{\overline{x}} P'$ then P contains a free $\overline{x}.Q$, if $P \xrightarrow{x} P'$ then a free $x.Q$, and if $P \xrightarrow{?x{=}y} P'$ then a free $\overline{x}.Q_1$ and $y.Q_2$ or vice versa. The rest of the proof is by induction on the derivation of $P \xrightarrow{\alpha}_s P'$. □

Theorem 13 (Congruence) $P \sim Q$ *implies* $E[P] \sim E[Q]$.

Proof. We construct the smallest relation $\mathcal{S}$ which contains $\sim$, which is closed with respect to structural congruence, and which satisfies

1. if $P \; \mathcal{S} \; Q$ then $(\nu x)P \;\; \mathcal{S} \;\; (\nu x)Q$, and $\alpha.P \;\; \mathcal{S} \;\; \alpha.Q$, and $!P \;\; \mathcal{S} \;\; !Q$
2. if $P_1 \; \mathcal{S} \; Q_1$ and $P_2 \; \mathcal{S} \; Q_2$ then $P_1|P_2 \; \mathcal{S} \; Q_1|Q_2$.

Clearly if $P \sim Q$ then $E[P] \; \mathcal{S} \; E[Q]$ for all contexts. It remains to prove that $\mathcal{S}$ is a fusion bisimulation, which we do by induction on the construction of $\mathcal{S}$. Take any $P_0 \; \mathcal{S} \; Q_0$, which must have been deduced from one of the closure properties of $\mathcal{S}$. An interesting case is $P_0 = \, !P$ and $Q_0 = \, !Q$ and $P \;\mathcal{S}\; Q$, which in fact generalises the parallel case. Again, following Corollary 12, we use Lemma 10 to analyse the possible transitions undergone by P_0. There are four parts of the bisimulation definition 8 to satisfy.

1. Assume $P_0 \xrightarrow{\overline{u}} P_0'$. (The input case follows similarly). From Lemma 10 this transition is actually

   $$!P \xrightarrow{\overline{u}}\equiv P_1' \mid !P \quad \text{with} \quad P \xrightarrow{\overline{v}} P_1', \; P \vdash u{=}v.$$

   Using Definition 6 we can rewrite $P_1' \mid !P$ as $I{\cdot}(P'|!P)$ since $!P$ has arity 0 and without loss of generality we may assume that the bound names of I are not the same as any free names in P or Q. Thus we obtain

   $$!P \xrightarrow{\overline{u}}\equiv I{\cdot}(P'|!P) \quad \text{with} \quad P \xrightarrow{\overline{v}} I{\cdot}P', \; P \vdash u{=}v.$$

   By the induction hypothesis we get $Q \xrightarrow{\overline{v}} I{\cdot}Q'$ with $P' \;\mathcal{S}\; Q'$. We also get $E(P) = E(Q)$, allowing us to convert the $\xrightarrow{\overline{v}}$ into a $\xrightarrow{\overline{u}}$. And since $\mathcal{S}$ is structurally closed, we deduce $I{\cdot}(P'|!P) \; \mathcal{S} \; I{\cdot}(Q'|!Q)$.
2. Assume $P_0 \xrightarrow{?x{=}y} P_0'$. From Lemma 10 there are two possibilities for what this transition actually is. The first is

   $$!P \xrightarrow{?x{=}y}\equiv P'@P'' \mid !P \quad \text{with} \quad P \xrightarrow{\overline{u}} P', \; P \xrightarrow{v} P'', \; P \vdash ?u{=}v = ?x{=}y.$$

   From the induction hypothesis we deduce that Q can also undergo these transitions, giving $!Q \xrightarrow{?u{=}v} Q'@Q'' \mid !Q$ with $P' \;\mathcal{S}\; Q'$ and $P'' \;\mathcal{S}\; Q''$. Therefore, since $Q \vdash \; ?u{=}v = \; ?x{=}y$ just as P did, the process $x{=}y \mid !Q$ can make the appropriate τ transition and the case is finished. The other possibility is that the transition $!P \xrightarrow{?x{=}y} P_0'$ comes solely from $P \xrightarrow{?u{=}v} P'$, but this case is substantially the same.
3. Assume $P_0 \xrightarrow{\tau} P_0'$. This is substantially the same as the previous case.
4. Trivially, $E(!P) = E(!Q)$, since $E(\_)$ is preserved by replication.

Proofs for the other closure properties of $\mathcal{S}$ follow the same lines. □

4 Embedding the fusion calculus

In this section we consider the embedding of the fusion calculus into the pi-F calculus. Apart from its lack of explicit fusions, the fusion calculus has basically the same syntax as the pi-F calculus. (Although this fact is a little obscured by our stylistic choice to use datums for the pi-F calculus).

The fusion and pi-F calculi have different reaction relations. In particular, the pi-F calculus always allows a reaction between any input and output, while the fusion calculus only allows it if there are enough enclosing restrictions to remove the resulting explicit fusions. But despite this difference, the two calculi share the same equivalence relation (are fully abstract): two processes are judged equivalent in the fusion calculus (up to 'hyper-equivalence') if and only if they are judged equivalent in the pi-F calculus (up to fusion bisimulation).

We first recall the fusion calculus from [23]. Then we prove the full abstraction result.

Definition 14 (Fusion calculus) *The set of fusion processes $\mathcal{P}_{fu}$ is*

$$P \ ::= \mathbf{0} \ \big| \ P|P \ \big| \ !P \ \big| \ (\nu x)P \ \big| \ \overline{u}\widetilde{x}.P \ \big| \ u\widetilde{x}.P.$$

Its structural congruence $\equiv_{fu}$ is as in Figure 2 without the fusion rules. Its reaction relation satisfies the following rule and is closed with respect to $\equiv_{fu}$ and contexts $\_|\_$ and $(\nu x)P$:

$$(\nu\widetilde{u})(\overline{u}\widetilde{y}.P \mid u\widetilde{x}.Q \mid R) \quad \searrow_{fu} \quad P\sigma \mid Q\sigma \mid R\sigma,$$

where $ran(\sigma), dom(\sigma) \subseteq \{\widetilde{x}, \widetilde{y}\}$ and $\widetilde{u} = dom(\sigma)\backslash ran(\sigma)$ and $\sigma(v) = \sigma(w)$ if and only if $(v, w) \in E(\widetilde{x}{=}\widetilde{y})$.

The side-conditions on the reaction relation describe a natural concept. Consider the equivalence relation generated from the equalities $\widetilde{x}{=}\widetilde{y}$. The side-conditions ensure that, for each equivalence class, every element is mapped by σ to a single free witness.

The labelled transition system for the fusion calculus is given in Figure 5. We explain two unconventional aspects of the LTS.

(1) The fusion calculus uses a 'tell' transition $\xrightarrow{x=y}$ which indicates that an internal reaction has caused a fusion: for example, $\overline{u}x.P \mid uy.Q \xrightarrow{x=y} P \mid Q$. This fusion has its effect during the transition. It has potentially global effect, up to some delimiting restriction, and so the transition can only be discharged in the presence of that restriction:

$$\frac{\overline{u}x.P \mid uy.Q \mid R \xrightarrow{x=y} P \mid Q \mid R}{(\nu x)(\overline{u}x.P \mid uy.Q \mid R) \xrightarrow{\mathbf{I}} (P|Q|R)\{y/x\}}$$

Here the *identity* fusion transition $P \xrightarrow{\mathbf{I}} P'$ has no fusing effect, and is equivalent to the conventional τ transition $P \xrightarrow{\tau} P'$. As usual, it is a lemma that $P \searrow_{\mathrm{fu}} P'$ if and only if $P \xrightarrow{\mathbf{I}} P'$. (We sometimes write $P \xrightarrow{\phi} P'$ to indicate that the transition causes a fusion, but without specifying which names are fused.)

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

$$\frac{P \xrightarrow{\overline{u}\widetilde{x}} P' \quad Q \xrightarrow{u\widetilde{y}} Q'}{P \mid Q \xrightarrow{\widetilde{x}=\widetilde{y}} P' \mid Q'} \qquad \frac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$$

$$\frac{P \xrightarrow{\alpha} P' \quad x \notin \mathrm{n}(\alpha)}{(\nu x)P \xrightarrow{\alpha} (\nu x)P'} \qquad \frac{P \xrightarrow{\widetilde{x}=\widetilde{y}} P' \quad (\widetilde{x}=\widetilde{y}) \vdash u{=}v \quad u \neq v}{(\nu u)P \xrightarrow{\widetilde{x}=\widetilde{y}\setminus u} P'\{v/u\}}$$

$$\frac{P \equiv Q \xrightarrow{\alpha} Q' \equiv P'}{P \xrightarrow{\alpha} P'} \qquad \frac{P \xrightarrow{(\widetilde{y})a\widetilde{z}} P' \quad x \in \widetilde{z} - \widetilde{y} \quad a \notin \{x, \overline{x}\}}{(\nu x)P \xrightarrow{(x\widetilde{y})a\widetilde{z}} P'}$$

$$\alpha ::= \overline{u}\widetilde{x} \mid u\widetilde{x} \mid \widetilde{x}=\widetilde{y} \qquad \gamma ::= \alpha \mid (\nu\widetilde{y})\overline{u}\widetilde{x} \mid (\nu\widetilde{y})u\widetilde{x}$$

Fig. 5: Labelled transition system for the fusion calculus. For the first restriction rule, the free names $\mathrm{n}(\alpha)$ of a fusion label are those names related to different names. For the second, u and v are related by the fusion but are not identical.

(2) The fusion calculus distinguishes between binding labels γ and non-binding labels α. The rules for communication, parallel composition and structural congruence only apply to non-binding labels. To deduce a transition $P \mid (\nu x)\overline{u}x.Q \xrightarrow{(x)\overline{u}x} P \mid Q$ it is necessary first to push the restriction to the outside with structural congruence, then deduce a transition from the contents $P \mid \overline{u}x.Q$, and finally re-apply the restriction. This procedure is used in Lemma 17 to deduce some derived transition rules that are closer in spirit to those of the pi-F calculus.

Hyper-equivalence is the standard bisimulation congruence for the fusion calculus. Its definition makes use of *substitutive effects*, which we recall here. We also introduce *partial* substitutive effects which will be used in Lemma 17.

Definition 15 (Substitutive effect) *The substititive effect of a fusion $\widetilde{x}=\widetilde{z}$ is a substitution which sends all members of each equivalence classes to one representative of that class.*

A partial *substitutive effect of a fusion $\widetilde{x}=\widetilde{z}$ with respect to a set of names $\widetilde{y}$ is a substitution which again involves a representative from each class, and satisfies:*

1. *the names outside $\widetilde{y}$ are not substituted;*
2. *the names inside $\widetilde{y}$ are substituted by their respective representatives;*
3. *the representatives are chosen from outside $\widetilde{y}$ when possible.*

For a partial substitutive effect σ, define $dom(\sigma) = \{x : x\sigma \neq x\}$.

Note that substitutive effects yield the side condition on fusion reaction (Definition 14). They are generalised by partial substitutive effects: a substitutive effect of $\phi = \widetilde{x} = \widetilde{z}$ is a partial substitutive effect of ϕ with respect to $\{\widetilde{x}, \widetilde{z}\}$. In essence, partial substitutive effects allow for fewer than necessary restrictions as compared to the fusion reaction relation. This makes them useful in relation to the fusion LTS.

Definition 16 (Hyper-equivalence) *A symmetric relation $\mathcal{S}$ is a* hyper-bisimulation *for the fusion calculus if whenever $P \mathrel{\mathcal{S}} Q$ then for all substitutions σ,*

- *if $P\sigma \xrightarrow{\gamma} P'$ with $\mathrm{bn}(\gamma) \cap \mathrm{fn}(Q\sigma) = \emptyset$ then $Q\sigma \xrightarrow{\gamma} Q'$ and $P'\rho \mathrel{\mathcal{S}} Q'\rho$ for some substitutive effect ρ of γ.*

Hyper-equivalence $\sim_{fu}$ *is the largest hyper-bisimulation.*

Note that the original fusion calculus paper defines bisimulation and hyper-equivalence separately, while we have combined the definitions for convenience.

Lemma 17 *The following transitions can be derived from the fusion LTS (Figure 5). We list only the send transitions; receive transitions are the same.*

1. *$P \xrightarrow{(\widetilde{y})\overline{u}\widetilde{x}}_{fu} P'$ implies $(\nu z)P \xrightarrow{(\widetilde{y})\overline{u}\widetilde{x}}_{fu} (\nu z)P'$ if $z \notin \{\widetilde{x}, \widetilde{y}, u\}$.*
2. *$P \xrightarrow{(\widetilde{y})\overline{u}\widetilde{x}}_{fu} P'$ implies $P|Q \xrightarrow{(\widetilde{y})\overline{u}\widetilde{x}}_{fu} P'|Q$ assuming $\{\widetilde{y}\} \cap \mathrm{fn}(Q) = \emptyset$.*
3. *$P \xrightarrow{(\widetilde{y})\overline{u}\widetilde{x}}_{fu} P'$ implies $!P \xrightarrow{(\widetilde{y})\overline{u}\widetilde{x}}_{fu} P' \mid !P$ assuming $\{\widetilde{y}\} \cap \mathrm{fn}(!P) = \emptyset$.*
4. *$P \xrightarrow{(\widetilde{y}_1)\overline{u}\widetilde{x}_1}_{fu} P'$ and $Q \xrightarrow{(\widetilde{y}_2)u\widetilde{x}_2}_{fu} Q'$ imply $P|Q \xrightarrow{(\widetilde{x}_1 = \widetilde{x}_2)\backslash dom(\sigma)}_{fu} (\nu\widetilde{y}_1\widetilde{y}_2)(P'\sigma|Q'\sigma)$ for any partial substitutive effect σ of $\widetilde{x}_1 = \widetilde{x}_2$ with respect to $\{\widetilde{y}_1, \widetilde{y}_2\}$.*

The final derived rule is subtle. Its consequent may be restated as $P|Q \xrightarrow{(\widetilde{x}_1 = \widetilde{x}_2)\backslash\widetilde{y}_a}_{\mathrm{fu}} \equiv (\nu\widetilde{y}_b)(P'\sigma|Q'\sigma)$ where as many of the binders $\widetilde{y}_1, \widetilde{y}_2$ as possible are removed by interchanging them with other names from the fusion $\widetilde{x}_1 = \widetilde{x}_2$. These form the set $\widetilde{y}_a$. The set $\widetilde{y}_b$ contains those names that cannot be removed. More formally, the names $\widetilde{y}_a$ and $\widetilde{y}_b$ are distinct and partition $\widetilde{y}_1\widetilde{y}_2$. The fusion $\widetilde{x}_1 = \widetilde{x}_2$ entails that each name in $\widetilde{y}_a$ is fused with an element not in $\widetilde{y}_a$, and σ substitutes each element in $\widetilde{y}_a$ accordingly. The names $\widetilde{y}_b$ are not affected by the fusion $(\widetilde{x}_1 = \widetilde{x}_2)\backslash\widetilde{y}_a$. We assume no clashes: $\widetilde{y}_1$ and $\widetilde{y}_2$ are distinct, and $\{\widetilde{y}_1\} \cap \mathrm{fn}(Q) = \{\widetilde{y}_2\} \cap \mathrm{fn}(P) = \emptyset$.

Full abstraction for fusion calculus

The translation $(\cdot)^*$ of processes from fusion calculus to pi-F is trivial:

$$\begin{aligned}
(\mathbf{0})^* &= \mathbf{0} \\
(P|Q)^* &= P^* \mid Q^* \\
(!P)^* &= !(P^*) \\
\big((\nu x)P\big)^* &= (\nu x)P^* \\
(\overline{u}\widetilde{x}.P)^* &= \overline{u}.(\langle\widetilde{x}\rangle|P^*) \\
(u\widetilde{x}.P)^* &= u.(\langle\widetilde{x}\rangle|P^*)
\end{aligned}$$

We state now the connection between reaction relations in the two calculi. For a fusion process P,

$$P \searrow_{\mathrm{fu}} P' \text{ implies } P^* \searrow_F P'^*$$
$$P^* \searrow_F P'_1 \text{ implies } \exists \widetilde{u} : (\nu\widetilde{u})P \searrow_{\mathrm{fu}} P' \text{ and } P'^* \equiv_F (\nu\widetilde{u})P'_1.$$

The striking feature is that reaction of a process in the image of $(\cdot)^*$ does not necessarily result in a process also in the image. For example,

$$\overline{u}.(\langle y\rangle|P^*) \mid u.(\langle x\rangle|Q^*) \searrow_F x{=}y \mid P^* \mid Q^*.$$

The process on the left is in the image of the fusion calculus under $(\_)^*$, but the one on the right has an free explicit fusion and so is not. Essentially, because the fusion calculus has unbound output and input processes and yet lacks explicit fusions, it can only allow those reactions that have enough extra restrictions to discharge all the resulting explicit fusions.

The rest of this section is devoted to the connection between fusion and pi-F transitions, and then between fusion and pi-F bisimulation. For the labels, there are two issues.

1. The fusion calculus has a fusion transition $P \xrightarrow{x=y}_{\mathrm{fu}} P'$ that *tells* the environment that a fusion has occurred; the pi-F calculus has a different fusion transition $Q \xrightarrow{?x=y}_F Q'$ that *asks* for an explicit fusion to be present in order to allow reaction. But in fact, the 'ask' fusion label of the pi-F calculus is not actually needed in the quotiented LTS as discussed in Section 3 – it merely serves to avoid quantifying over all contexts. It can instead be deduced from the labels $\overline{x}$, y and τ. And as for the 'tell' label of the fusion calculus, it amounts to a τ transition in pi-F with some explicit fusions in the resulting process.
2. The fusion calculus has transitions $P \xrightarrow{(\widetilde{y})\overline{u}\widetilde{x}}_{\mathrm{fu}} P'$ in which the label carries the names to be communicated; the pi-F calculus uses CCS-style labels $Q \xrightarrow{\overline{u}}_F Q'$. In fact, apart from the channel name itself, all the other information conveyed in a fusion label is conveyed in the pi-F calculus by the interface of the resulting process. (This difference is merely due to our presentational style; it is not a fundamental difference between the two calculi.)

The following transitions illustrate the connection between fusion and pi-F transitions. The connection is stated formally in Lemma 18.

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

Lemma 18 *Given a pi process P then P^* has arity 0 and, for every subterm Q of P^*, $E(Q) = \boldsymbol{I}$. Furthermore,*

1. *if P undergoes a transition in the fusion calculus, the transition is one of*
   (a) $P \xrightarrow{(\widetilde{y})u\widetilde{x}}_{fu} P'$ *with* $P^* \xrightarrow{u}_F (\nu\widetilde{y})(\langle\widetilde{x}\rangle|P'^*)$*, or likewise for* $\xrightarrow{(\widetilde{y})\overline{u}\widetilde{x}}_{fu}$
   (b) $P \xrightarrow{\phi}_{fu} P'$ *with* $P^* \xrightarrow{\tau}_F \phi \mid P'^*$*;*
2. *if P^* undergoes a transition in the pi-F calculus, it is one of*
   (a) $P^* \xrightarrow{u}_F \equiv (\nu\widetilde{y})(\langle\widetilde{x}\rangle|P_1'^*)$ *with* $P \xrightarrow{(\widetilde{y})u\widetilde{x}}_{fu} P_1'$*, or likewise for* $\xrightarrow{\overline{u}}_F$
   (b) $P^* \xrightarrow{?x{=}y}_F \equiv \phi|P_1'^*$ *with* $\exists P_2' : P\{y/x\} \xrightarrow{\phi}_{fu} P_2',\ x{=}y|\phi|P_1'^* \equiv x{=}y|\phi|P_2'^*$
   (c) $P^* \xrightarrow{\tau}_F \equiv \phi|P_1'^*$ *with* $\exists P_2' : P \xrightarrow{\phi}_{fu} P_2',\ \phi|P_1'^* \equiv \phi|P_2'^*$.

Proof. The first part is proved by induction on the derivation of the transition $\longrightarrow_{\mathrm{fu}}$. The second part is proved by induction on the structure of P (which is also the structure of P^*) using Lemma 17. Part 2b looks complicated because the explicit fusion ϕ in $P \xrightarrow{?x{=}y}_F \phi|P_1'^*$ can have immediate effect on $P_1'^*$, but the fusion label ϕ in the transition $P \xrightarrow{\phi}_{\mathrm{fu}} P_2'$ only has effect after the process has been enclosed by a restriction. □

We now proceed to the main full abstraction result. In essence, the hyper-equivalence relation in the fusion calculus is pi-F bisimulation relation but with the interfaces and explicit fusions stripped away. There are two interesting parts to the proof. (1) Reconstructing an 'ask' fusion transition in the pi-F calculus from a τ transition in the fusion calculus; this uses the fact that ask transitions are not essential. (2) Reconstructing a 'tell' fusion transition in the fusion calculus from a τ transition in the pi-F calculus, via Lemma 18.2c above.

Theorem 19 *$P \sim_{fu} Q$ if and only if $P^* \sim_F Q^*$.*

Proof. In the forwards direction, we construct a relation $\mathcal{S}$ on pi-F processes such that $P\ \mathcal{S}\ Q$ if and only if P and Q have standard forms $I{\cdot}(\phi|P_1^*)$ and $I{\cdot}(\phi|Q_1^*)$ respectively, and $P_1 \sim_{\mathrm{fu}} Q_1$. We prove that $\mathcal{S}$ is a fusion bisimulation. Clearly the interfaces match; it is the contents that are more difficult. Consider $P\ \mathcal{S}\ Q$ with $P, Q : 0$ such that $P \equiv \phi|P_1^*$, $Q \equiv \phi|Q_1^*$ and $P_1 \sim_{\mathrm{fu}} Q_1$. There are four parts of fusion bisimulation (Definition 8) to satisfy. We use Lemmas 18 and 10 to analyse the possible transitions.

1. First consider the transition
$$\phi|P_1^* \equiv P \xrightarrow{\overline{u}}_F \equiv I{\cdot}P' \quad \text{with} \quad P_1^* \xrightarrow{\overline{v}}_F I{\cdot}P_1'^*,\ P' \equiv \phi|P_1'^*,\ \phi \vdash u{=}v,$$
   where $I = (\nu\widetilde{x})(\langle\widetilde{y}\rangle|\_)$ and $\widetilde{x}$ does not bind ϕ. Then $P_1 \xrightarrow{(\widetilde{x})\overline{v}\widetilde{y}}_{\mathrm{fu}} P_1'$. Since $P_1 \sim_{\mathrm{fu}} Q_1$, we obtain $Q_1 \xrightarrow{(\widetilde{x})\overline{v}\widetilde{y}}_{\mathrm{fu}} Q_1'$ with $P_1' \sim_{\mathrm{fu}} Q_1'$. By Lemma 18, $Q_1^* \xrightarrow{\overline{v}}_F I{\cdot}Q_1'^*$ and hence $\phi|Q_1^* \xrightarrow{\overline{u}}_F I{\cdot}(\phi|Q_1'^*)$. Finally, $I{\cdot}(\phi|P_1'^*)\ \mathcal{S}\ I{\cdot}(\phi|Q_1'^*)$ by construction of $\mathcal{S}$. An analogous result holds for the input case.
2. Now consider the transition
$$\phi|P_1^* \equiv P \xrightarrow{?x{=}y}_F P' \quad \text{with} \quad P_1^* \xrightarrow{?u{=}v}_F \psi|P_1'^*,\ P' \equiv \phi|\psi|P_1'^*,\ \phi \vdash ?u{=}v = ?x{=}y.$$

Given the transition $P_1^* \xrightarrow{?u=v}_F \psi|P_1'^*$ we need to reconstruct the fact that Q_1^* can undergo a τ transition: writing ρ for a substitutive effect of ψ,

$$
\begin{aligned}
&P_1^* \xrightarrow{?u=v}_F \psi|P_1'^* \\
&\qquad \Rightarrow\ \exists P_2' : P_1\{u/v\} \xrightarrow{\psi}_{\text{fu}} P_2' \text{ with } u{=}v|\psi|P_1'^* \equiv u{=}v|\psi|P_2'^* \\
&\qquad \Rightarrow\ Q_1\{u/v\} \xrightarrow{\psi}_{\text{fu}} Q_2' \text{ with } P_2'\rho \sim_{\text{fu}} Q_2'\rho \\
&\qquad \Rightarrow\ Q_1^*\{u/v\} \xrightarrow{\tau}_F \psi|Q_2'^* \text{ with } (P_2'\rho)^* \ \mathcal{S}\ (Q_2'\rho)^* \\
&\qquad \Rightarrow\ u{=}v|\phi|Q_1^* \xrightarrow{\tau}_F u{=}v|\phi \mid \psi \mid Q_2'^*
\end{aligned}
$$

Finally, $P_2'\rho^*\ \mathcal{S}\ Q_2'\rho^*$ implies $\psi|P_2'^*\ \mathcal{S}\ \psi|Q_2'^*$. From the closure properties of $\mathcal{S}$, and since $u{=}v|P' \equiv u{=}v|\phi|\psi|P_2'^*$, we fulfill the requirement that $u{=}v|P'\ \mathcal{S}\ u{=}v|\phi|\psi|Q_2'^*$. An analogous result holds for the τ transition.

In the reverse direction, we construct a relation $\mathcal{S}$ on fusion processes such that $P\ \mathcal{S}\ Q$ if and only if $P^* \sim_F Q^*$. It remains to prove that the relation $\mathcal{S}$ is a hyper-equivalence. Note that $\mathcal{S}$ is closed with respect to substitution, since the substitution $\{y/x\}$ can be expressed as the context $(\nu x)(x{=}y|\_)$, and $\sim_F$ is closed with respect to all contexts (Theorem 13), and $(\_)^*$ preserves substitution. It is therefore enough to analyse a label $P \xrightarrow{\gamma}_{\text{fu}} P'$ without substitution, since all substitutions $P\sigma \xrightarrow{\gamma}_{\text{fu}} P''$ automatically follow. Lemma 18 accounts for output and input labels. For a 'tell' fusion label, suppose that $P \xrightarrow{\phi}_{\text{fu}} P'$. From Lemmas 18 and 10,

$$
P \xrightarrow{\phi}_{\text{fu}} P' \ \Rightarrow\ P^* \xrightarrow{\tau}_F \phi|P'^* \ \Rightarrow\ Q^* \xrightarrow{\tau}_F \phi|Q'^* \text{ with } \phi|P'^* \sim_F \phi|Q'^*.
$$

From Lemma 18 we see that Q has a corresponding transition $Q \xrightarrow{\phi}_{\text{fu}} Q_1'$ with $\phi|Q_1'^* \equiv \phi|Q'^*$. Applying an appropriate restriction context to $\phi|P'^* \sim_F \phi|Q'^*$, we get the desired substitutive effect ρ of ϕ: that is, $P'^*\rho \sim_F Q'^*\rho$, and hence $P'\rho\ \mathcal{S}\ Q_1'\rho$. $\square$

5 Embedding the pi calculus

In this section we give an embedding of the pi calculus into the pi-F calculus, and show that it is fully abstract with resect to reaction. It is not fully abstract with respect to bisimulation, because the pi calculus allows terms such as $\nu xy.(\overline{u}xy \mid P)$ in which no context can make x and y equal in P. However, the pi-F context $uzz \mid \_$ can. This makes pi-F contexts more discriminating than pi contexts. We return to this difference in the conclusions.

We first recall the pi calculus from [21]. Then we prove the results.

Definition 20 (Pi calculus) *The pi calculus is*

$$
\begin{aligned}
P &::= \mathbf{0} \mid P|P \mid !P \mid (\nu x)P \mid \overline{u}\widetilde{x}.P \mid u(\widetilde{x}).P && \textit{Processes} \\
E &::= \_ \mid P|E \mid E|P \mid !E \mid (\nu x)E \mid \overline{u}\widetilde{x}.E \mid u(\widetilde{x}).E && \textit{Environments}
\end{aligned}
$$

$$u(\widetilde{x}).P \xrightarrow{u(\widetilde{x})} P \qquad\qquad \overline{u}\,\widetilde{x} \xrightarrow{\overline{u}\widetilde{x}} \mathbf{0}$$

$$\frac{P \xrightarrow{\mu} P' \quad y \notin \mu}{(\nu y)P \xrightarrow{\mu} (\nu y)P'} \qquad\qquad \frac{P \xrightarrow{(\widetilde{z})\overline{u}\widetilde{x}} P' \quad y \neq u,\ y \in \widetilde{x}\backslash\widetilde{z}}{(\nu y)P \xrightarrow{(y\widetilde{z})\overline{u}\widetilde{x}} P'}$$

$$\frac{P|!P \xrightarrow{\mu} P'}{!P \xrightarrow{\mu} P'} \qquad\qquad \frac{P \xrightarrow{\mu} P' \quad \mathrm{bn}(\mu) \cap \mathrm{fn}(Q) = \emptyset}{P|Q \xrightarrow{\mu} P'|Q}$$

$$\frac{P \equiv_\alpha Q \xrightarrow{\mu} Q' \equiv_\pi P'}{P \xrightarrow{\mu} P'} \qquad\qquad \frac{P \xrightarrow{(\widetilde{z})\overline{u}\widetilde{y}} P' \quad Q \xrightarrow{u(\widetilde{x})} Q' \quad \widetilde{z} \cap \mathrm{fn}(Q) = \emptyset}{P|Q \xrightarrow{\tau} (\nu\widetilde{z})(P'|Q'\{\widetilde{y}/\widetilde{x}\})}$$

Fig. 6: Structured labelled transition system for the pi calculus. The transitions of $P|Q$ have mirror cases, which we have omitted. The notation $\equiv_\alpha$ is alpha-renaming. Labels μ range over τ, input $u(\widetilde{x})$ and possibly-binding output $(\widetilde{z})\overline{u}\widetilde{x}$.

Its structural congruence $\equiv_\pi$ is as in Figure 2 minus the fusion rules. Its reaction relation is given by the following axiom, and closed with respect to $\equiv_\pi$ and contexts $\_|\_$ and $(\nu x)\_$:

$$\overline{u}\widetilde{y} \mid u(\widetilde{x}).P \ \searrow_\pi \ P\{\widetilde{y}/\widetilde{x}\}.$$

The labelled transitions for the pi calculus are given in Figure 6. It is a standard result that $P \searrow_\pi P'$ if and only if $P \xrightarrow{\tau}_\pi P'$.

Embedding the pi calculus

The translation $(\cdot)^*$ of processes from asynchronous pi calculus to asynchronous pi-F is trivial:

$$\begin{aligned}
(\mathbf{0})^* &= \mathbf{0} \\
(P|Q)^* &= P^* \mid Q^* \\
(!P)^* &= !(P^*) \\
\big((\nu x)P\big)^* &= (\nu x)P^* \\
(\overline{u}\,\widetilde{x})^* &= \overline{u}.\langle\widetilde{x}\rangle \\
(u(\widetilde{x}).P)^* &= u.(\nu\widetilde{x})(\langle\widetilde{x}\rangle|P^*)
\end{aligned}$$

Proposition 21 below states that reaction is preserved between pi and pi-F calculi: for pi processes P,

$$P \searrow_\pi P' \text{ implies } P^* \searrow_F P'^*$$
$$P^* \searrow_F P_1' \text{ implies } P \searrow_\pi P' \text{ and } P'^* \equiv_F P_1'.$$

For example, the pi reaction $\overline{u}yP \mid u(x)Q \searrow_\pi P \mid Q\{y/x\}$ corresponds to the pi-F reaction

$$
\begin{array}{lll}
\multicolumn{3}{l}{\overline{u}.(\langle y\rangle|P^*) \mid u.(\nu x)(\langle x\rangle|Q^*)} \\
\searrow_F & (\langle y\rangle|P^*) \mathbin{@} (\nu x)(\langle x\rangle|Q^*) & \\
\equiv_F & (\nu x)(x{=}y \mid P^* \mid Q^*) & \textit{renaming if necessary} \\
\equiv_F & (\nu x)(x{=}y \mid P^* \mid Q^*\{y/x\}) & \textit{substituting } y \textit{ for } x \\
\equiv_F & P^* \mid Q^*\{y/x\} & \textit{remove unused bound } x.
\end{array}
$$

We remark that reaction of a process in the image of $(\cdot)^*$ always results in another process in the image of $(\cdot)^*$. Even though the reaction temporarily results in a fusion $x{=}y$, one of those fused names must necessarily have arisen from a pi abstraction $(x)Q$ and so the fusion can be factored away. This contrasts with the fusion calculus, where reaction led outside the image of $(\cdot)^*$.

Proposition 21

1. $P \xrightarrow{\tau}_\pi P'$ *implies* $P^* \xrightarrow{\tau}_F P'^*$
2. $P^* \xrightarrow{\tau}_F P'_1$ *implies* $P \xrightarrow{\tau}_\pi P'$ *and* $P'^* \equiv_F P'_1$

Proof. We use a similar technique to that for the fusion calculus (Lemma 18). P^* has arity 0, and for every subterm Q of P^* then $E(Q) = \mathbf{I}$. Moreover,

1. if P makes a transition in the pi calculus, it is one of
   (a) $P \xrightarrow{(\widetilde{z})\overline{u}\widetilde{x})}_\pi\equiv_\pi P'$ with $P^* \xrightarrow{\overline{u}}_F (\nu\widetilde{z})(\langle\widetilde{x}\rangle|P'^*)$
   (b) $P \xrightarrow{u(\widetilde{x})}_\pi\equiv_\pi P'$ with $P^* \xrightarrow{u}_F (\nu\widetilde{x})(\langle\widetilde{x}\rangle|P'^*)$
   (c) $P \xrightarrow{\tau}_\pi\equiv_\pi P'$ with $P^* \xrightarrow{\tau}_F P'^*$;
2. if P^* makes a transition in the pi-F calculus, it is one of
   (a) $P^* \xrightarrow{\overline{u}}_F\equiv_\pi (\nu\widetilde{z})(\langle\widetilde{x}\rangle|P'_1)$ with $P \xrightarrow{(\widetilde{z})\overline{u}\widetilde{x}}_\pi P'$ and $P'_1 \equiv_F P'^*$
   (b) $P^* \xrightarrow{u}_F\equiv_\pi (\nu\widetilde{x})(\langle\widetilde{x}\rangle|P'_1)$ with $P \xrightarrow{u(\widetilde{x})}_\pi P'$ and $P'_1 \equiv_F P'^*$
   (c) $P^* \xrightarrow{\tau}_F\equiv_\pi P'_1$ with $P \xrightarrow{\tau}_\pi P'$ and $P'_1 \equiv_F P'^*$
   (d) $P^* \xrightarrow{?x{=}x}_F\equiv_\pi P'_1$ with $P \xrightarrow{\tau}_\pi P'$ and $P'_1 \equiv_F P'^*$.

The proof is routine: an induction on the derivation of $P \xrightarrow{\mu}_\pi P'$ for Part 1, and an induction on the structure of P for Part 2 (using Lemma 10 to deduce possible transitions).

We just remark on the equivalence rule in the pi calculus,

$$\frac{P \equiv_\alpha Q \xrightarrow{\mu}_\pi Q' \equiv_\pi P'}{P \xrightarrow{\mu}_\pi P'}$$

which has alpha-renaming on the left. This allows us to rename any any bound labels μ in $Q \xrightarrow{\mu} Q'$ to avoid clashes, so allowing $Q|Q_1 \xrightarrow{\mu} Q'|Q_1$ for any Q_1. But in the pi-F calculus, alpha-renaming of labels is instead achieved through structural congruence on the right hand side. For instance, in

$$Q^* \xrightarrow{\overline{u}}_F (\nu\widetilde{z})(\langle\widetilde{x}\rangle \mid Q'^*)$$

the right hand side is structurally congruent to $(\nu\widetilde{z}')(\langle\widetilde{x}\rangle\sigma \mid Q'^{*}\sigma)$ where $\sigma = \{\widetilde{z}'/\widetilde{z}\}$. □

6 Conclusions

We have introduced *explicit fusions* of names, presented formally in a process calculus called pi-F. It comes from the same tradition as the fusion calculus, the solos calculus and the chi calculus: like these, it has non-binding input and the ability to fuse names. What distinguishes the pi-F calculus is that it uses explicit fusions to give a small-step account of reaction. The fact that pi-F writes fusions explicitly as part of its term algebra has significance for implementation and bisimulation, as we discuss below.

The connection between the pi-F, fusion and pi calculi is itself interesting. As mentioned, pi-F uses explicit fusions to provide a small-step account both of substitution in the pi calculus and of fusions in the fusion calculus. We have proved this by giving embeddings of both calculi into the pi-F calculus. The pi embedding is fully abstract with respect to reaction: a reaction made by a pi term corresponds to one made by its embedding, and vice versa. The fusion embedding does not share this property. This is to be expected. The pi-F reaction is a local reaction between input and output processes, whose result contains explicit fusions; in contrast, reaction in the fusion calculus requires the presence of an enclosing restriction, which then removes all explicit fusions immediately. Despite this difference, the embedding of the fusion calculus is fully abstract with respect to bisimulation. In fact, the pi-F calculus can be regarded as a simpler reformulation of the fusion calculus. This conclusion was unexpected. We did not create the pi-F calculus with such a simplification in mind. Instead we created it as an attempt to write – in process-calculus syntax – a symmetric variant of Milner's action calculus framework [10]. (A variant of this idea has more recently been explored by Schweimer [26]).

Theory

As discussed, pi-F calculus bisimulation coincides with fusion calculus hyper-equivalence. However, hyper-equivalence is non-standard. In contrast, we have been able to use standard bisimulations for the pi-F calculus [29], directly copying the usual definitions from pi. It is possible to use these pi definitions because the pi-F calculus has local reactions like the pi calculus. The result is an elegant theory of fusion bisimulation.

The story starts with Sangiorgi's *open bisimulation* for the pi calculus [25]. The defining feature of open bisimulation is that it is closed with respect to substitution: if P and Q are open bisimilar, then so are $P\sigma$ and $Q\sigma$ for all substitutions σ. Consider the terms $\nu xy.(\overline{u}xy \mid P)$ and $\nu xy.(\overline{u}xy \mid Q)$. In the pi calculus the restricted names x and y always remain distinct in P and Q, in the sense that reaction between $\overline{x} \mid y()$ remains impossible. It would therefore be too strong to require here that $P\{y/x\}$ be bisimilar to $Q\{y/x\}$. So, Sangiorgi only uses substitutions up to the *distinction* $x \neq y$. But in the fusion and pi-F calculi, the names x and y in the terms can be fused, for instance by a context $uzz \mid \_$. Hence distinctions are not required. Indeed, Parrow and Victor informed us (private communication) that one of their original motivations for the fusion calculus was to simplify distinctions.

In [29] we study several standard bisimulation definitions for the pi-F calculus: ground and barbed congruences, reduction-closed and not. In pi-F (and unlike pi) these definitions all yield the same relation, precisely because distinctions are not required. This reassures us that the bisimulation studied here is *the* right one for fusion-based calculi.

In recent work, Boreale and Montanari have presented the 'd-fusion' calculus, which combines fusions with distinctions [3]. Their idea is to recover in a fusion setting the expressivity of the pi calculus – that is, the ability to generate unfusable names. The resulting calculus is surprisingly expressive. For instance, it has a fully abstract encoding of distributed mixed choice.

We have not studied weak bisimulation, but we remark that it has some interesting properties. Fu has done much work [7] on weak bisimulation relations, an their corresponding axiomatisations for a fusion calculus without replication. With replication Merro [18] has shown that a pi process, called an *equator* [14], can encode a fusion. Equators are defined as $\mathcal{E}(x,y) \stackrel{\text{def}}{=} !x(\widetilde{u}).\overline{y}\widetilde{u} \mid !y(\widetilde{u}).\overline{x}\widetilde{u}$. In the pi calculus, we have $P\{y/x\} \approx \nu x.(P \mid \mathcal{E}(x,y))$ where $\approx$ denotes weak barbed asynchronous congruence. We can rewrite this result in the pi-F calculus to emphasise the link with explicit fusions:

$$\mathcal{E}(x,y) \approx x{=}y.$$

This means that the embedding of the pi calculus in the pi-F calculus is fully abstract with respect to weak asynchronous barbed congruence. (Hence, up to this congruence, the pi calculus looses the ability to generate 'unfusable' names.)

Implementation

Interaction in the fusion and chi calculi require that, of the names that would be fused, no more than one fused name is unrestricted. This should be seen as a global constraint: it means that a potential reaction between input and output must first involve a global search for sufficient name restrictions. However, when a process calculus is used as a programming language, an implementation normally executes restrictions by turning them into globally fresh names [32]. It seems difficult to reconcile this with the search for name restrictions.

Victor, Parrow and Laneve have subsequently considered how to implement their fusion calculus in a graph-rewriting model, with their *solos diagrams* [16]. They considered a relaxed version of the fusion calculus in which reaction is allowed even if insufficient restrictions are present, as in $P|\overline{u}x|uy \xrightarrow{x=y} P$. In implementation terms, the $x{=}y$ is left as a persistent edge between x and y. This is clearly an explicit fusion. Victor *et al* do not mention the connection, and do not themselves develop the theory of their persistent edges; however the connection has been explored more recently by Heindel [12].

The current authors and Laneve have been pursuing a different implementation strategy [8]. We treat each channel(-name) as a distributed location. Each input or output atom is deployed to its correct location: so $u(x).P$ is placed at location u, and $\overline{v}z.Q$ is placed at location v. Then we factor out the explicit fusions from a term and store them as a network of forwarders: thus, an explicit fusion $u{=}v$ is stored as a forwarder from u to v. The small step substitution axioms of the pi-F calculus correspond to operations in the implementation: for instance, $u{=}v \mid u(x).P \equiv u{=}v \mid v(x).P$ shows an atom being forwarded from u to v; and $x{=}y|x{=}z \equiv x{=}y|y{=}z$ shows an incremental update to the network of forwarders. Separately, in a technique inspired by the solos calculus [17], we encode away continuations. This is so that, when $u(x).P$ is forwarded over the network to v, the continuation P will not be bulky – instead it will be just a collection of explicit fusions.

One might wonder why we chose to retain fusions in our calculus, even though our implementation used forwarders. The reason was that some forwarder details seemed too implementation-specific. Notably, two 'conflicting' forwarders $u \to v$ and $u \to v'$ might result in an atom being mistakenly forwarded to v' when it should have gone to v. Our implementation resolved this by ensuring (lazily) that it always produced a confluent tree of forwarders. It seemed more elegant for the calculus to abstract away from these details.

Actually we went on to abandon the fusion implementation strategy, because the trees of forwarders seemed inherently fragile in the presence of failure. For instance, if $x{=}y{=}z$ is represented as one forwarder $x \to y$ and another $y \to z$, then a failure of y can indirectly break the $x{=}z$ connection. In effect, the forwarder implementation of fusions is too centralised. We now use *linear* forwarders, where each forwarder is used no more than once and is guaranteed never to conflict [9]. This has proved more reliable, since a failure in the implementation corresponds directly to a failure in the forwarder calculus.

The Highwire group at Microsoft has also been exploring the use of explicit fusions (private communication) for program-design reasons. In existing programming, when two remote machines open a TCP link between themselves, then each machine has its own 'socket name'; the fact that the two sockets are bound together is part of the TCP infrastructure. Explicit fusions seem appropriate for modelling this binding of names. Similarly, it seems useful to write stand-alone components which expose 'ports', and then write a separate configuration file which binds components together at their ports. Explicit fusions seem appropriate for this configuration file. The Highwire group used a distributed protocol for rendezvous within a set of fused names, rather than the forwarders described above.

It should be noted that distributed systems in general *need* some sort of protocol for distributed rendezvous, and that the same protocol can be used as a distributed implementation of fusions. A practical example is that of buying a plane ticket either with BA or Alitalia, where BA and Alitalia run on separate web servers. In process calculus terms, this corresponds to the distributed choice $\overline{ba}.P_1 \mid \overline{al}.P_2 \mid ba().Q + al().Q$. A protocol that implements this choice can also implement the fusion $\overline{ba}.P_1 \mid \overline{al}.P_2 \mid (x{=}ba{=}al \mid x().Q)$. One such protocol is presented in [30].

Acknowledgements. We thank Peter Sewell, Robin Milner and anonymous referees for their helpful comments. Wischik thanks EPSRC for financial support during his Ph.D.

References

1. M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Lévy. Explicit substitutions. *Journal of Functional Programming*, 1(4):375–416, 1991. http://research.microsoft.com/Users/luca/Papers/ExplicitSub.pdf
2. G. Berry and G. Boudol. The chemical abstract machine. *Theoretical Computer Science*, 96(1):217–248, 1992. http://www-sop.inria.fr/meije/personnel/Gerard.Berry/cham.ps
3. M. Boreale, M. Buscemi, and U. Montanari. D-fusion: a distinctive fusion calculus. Submitted for publication, 2004.
4. J. Engelfriet and T. Gelsema. Multisets and structural congruence of the pi-calculus with replication. *Theoretical Computer Science*, 211(1–2):311–337, 1999. http://www.elsevier.com/cas/tree/store/tcs/sub/1999/211/1-2/2715.pdf
5. C. Fournet, Lévy, and A. Schmitt. An asynchronous, distributed implementation of mobile ambients. In *Proceedings of IFIP TCS 2000*, LNCS 1872:348–364. http://research.microsoft.com/~fournet/papers/implementation-of-ambients-tcs.pdf
6. Y. Fu. The chi-calculus. In *ICAPDC 1997*, pages 74–81. IEEE, Computer Society Press.
7. Y. Fu. Bisimulation lattice of chi processes. In *ASIAN 1998*, LNCS 1538:245–262. http://link.springer-ny.com/link/service/series/0558/bibs/1538/15380245.htm
8. P. Gardner, C. Laneve, and L. Wischik. The fusion machine. In *CONCUR 2002*, LNCS 2421:418–433. http://www.wischik.com/lu/research/fm.html
9. P. Gardner, C. Laneve, and L. Wischik. Linear forwarders. In *CONCUR 2003*, LNCS 2761:415–430. http://www.wischik.com/lu/research/linfwd.html
10. P. Gardner and L. Wischik. Symmetric action calculi (abstract). Unpublished manuscript, 1999. http://www.wischik.com/lu/research/sac.html
11. P. Gardner and L. Wischik. Explicit fusions. In *MFCS 2000*, LNCS 1893:373–382. http://www.wischik.com/lu/research/explicit-fusions.html
12. T. Heindel. Cyclic lambda-graph reduction (call-by-need) and their process semantics. Master's thesis, University of Tuebingen, 2003.
13. M. Hennessy and H. Lin. Symbolic bisimulations. *Theoretical Computer Science*, 138(2):353–389, 1995.
14. K. Honda and N. Yoshida. On reduction-based process semantics. *Theoretical Computer Science*, 152(2):437–486, 1995. ftp://ftp.dcs.qmw.ac.uk/lfp/kohei/red-sem.ps.gz
15. K. Honda. Elementary structures in process theory (1): Sets with renaming. *Mathematical Structures in Computer Science*, 2001. ftp://ftp.dcs.qmw.ac.uk/lfp/kohei/ps.ps.gz
16. C. Laneve, J. Parrow, and B. Victor. Solo diagrams. In *TACS 2001*, LNCS 2215:127–144. http://www.docs.uu.se/~victor/tr/solodiagrams.shtml

17. C. Laneve and B. Victor. Solos in concert. In *ICALP'99*, LNCS 1644:513–523. http://www.docs.uu.se/~victor/tr/solos.shtml
18. M. Merro. On the expressiveness of chi, update, and fusion calculi. In *EXPRESS 1998*, volume 16.2 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers. http://www.elsevier.nl/locate/entcs/volume16.2.html
19. R. Milner. Functions as processes. *Mathematical Structures in Computer Science*, 2(2):119–141, 1992.
20. R. Milner. Calculi for interaction. *Acta Informatica*, 33(8):707–737, 1996. ftp://ftp.cl.cam.ac.uk/users/rm135/ac9.ps
21. R. Milner. *Communicating and Mobile Systems: The Pi-Calculus*. Cambridge University Press, 1999.
22. J. Niehren and M. Mueller. Constraints for free in concurrent computation. In *Asian Computing Science Conference*, LNCS 1023:171–186, 1995. ftp://ps-ftp.dfki.uni-sb.de/pub/papers/ProgrammingSysLab/ConstraintsFree.ps.Z
23. J. Parrow and B. Victor. The fusion calculus: Expressiveness and symmetry in mobile processes. In *LICS '98*, pages 176–185. IEEE, Computer Society Press. http://www.docs.uu.se/~victor/tr/fusion.shtml
24. D. Sangiorgi. Pi-calculus, internal mobility and agent-passing calculi. *Theoretical Computer Science*, 167(1–2):235–275, 1996. http://www.inria.fr/RRRT/RR-2539.html
25. D. Sangiorgi. A theory of bisimulation for the pi-calculus. *Acta Informatica*, 33(1):69–97, 1996. ftp://ftp-sop.inria.fr/meije/theorie-par/davides/sub.ps.gz
26. R. Schweimeier. *Categorical and Graphical Models of Programming Languages*. PhD thesis, University of Sussex, 2001.
27. A. Unyapoth and P. Sewell. Nomadic pict: Correct communication infrastructure for mobile computation. *ACM SIGPLAN Notices*, 36(3):116–127, 2001. http://www.acm.org/pubs/citations/proceedings/plan/360204/p116-unyapoth
28. B. Victor and J. Parrow. Concurrent constraints in the fusion calculus. In *ICALP '98*, LNCS 1443:455–469. http://www.docs.uu.se/~victor/tr/ccfc.shtml
29. L. Wischik and P. Gardner. Strong bisimulation for the explicit fusion calculus. In *FOSSACS 2004*. To appear. http://www.wischik.com/lu/research/eft.html
30. L. Wischik and D. Wischik. A reliable protocol for synchronous rendezvous (note). Technical Report 2004-1, University of Bologna, 2004. http://www.wischik.com/lu/research/verona.html
31. L. Wischik. *Explicit Fusions: Theory and Implementation*. PhD thesis, University of Cambridge, 2001. http://www.wischik.com/lu/research/efti.html
32. L. Wischik. Old names for nu. Submitted for publication., 2004. http://www.wischik.com/lu/research/oldnames.html