The Fusion Calculus: Expressiveness and Symmetry in Mobile Processes

Joachim Parrow *   Björn Victor †

DRAFT of December 5, 1997, 16:35

Abstract

We present the *fusion calculus* as a significant step towards a canonical calculus of concurrency. It simplifies and extends the π-calculus.

The fusion calculus contains the polyadic π-calculus as a proper subcalculus and thus inherits all its expressive power. The gain is that fusion contains actions akin to updating a shared state, and a scoping construct for bounding their effects. Therefore it is easier to represent computational models such as imperative and concurrent constraints formalisms. It is also easy to represent the so called strong reduction strategies in the λ-calculus, involving reduction under abstraction. In the π-calculus these tasks require elaborate encodings.

The dramatic main point of this paper is that we achieve these improvements by *simplifying* the π-calculus rather than adding features to it. The fusion calculus has only one binding operator where the π-calculus has two (input and restriction). It has a complete symmetry between input and output actions where the π-calculus has not. There is only one sensible variety of bisimulation congruence where the π-calculus has at least three (early, late and open). Proofs about the fusion calculus, for example in complete axiomatizations and full abstraction, therefore are shorter and clearer.

Our results on the fusion calculus in this paper are the following. We give a structured operational semantics in the traditional style. The novelty lies in a new kind of action, *fusion* actions for emulating updates of a shared state. We prove that the calculus contains the π-calculus as a subcalculus. We define and motivate the bisimulation equivalence and prove a simple characterization of its induced congruence, which is given two versions of a complete axiomatization for finite terms. The expressive power of the calculus is demonstrated by giving a straight-forward encoding of the strong lazy λ-calculus, which admits reduction under λ abstraction.

*Dept. of Teleinformatics, Royal Institute of Technology, Electrum 204, S-164 40 Kista, Sweden. Email: `joachim@it.kth.se`.

†Dept. of Computer Systems, Uppsala University, Box 325, S-751 05 Uppsala, Sweden. Email: `Bjorn.Victor@DoCS.UU.SE`.

1 Introduction

We present the *fusion calculus*, which we claim is a significant step towards a canonical calculus of concurrency. It simplifies and extends the π-calculus. We mention three indications of its greater expressiveness:

1. Computational models involving a shared state, such as imperative and concurrent constraints formalisms, can only be represented indirectly in the π-calculus by introducing extra processes for the state. In the fusion calculus the basic mechanisms for updating a shared state are already present and the models have simpler representations.

2. Strong reduction strategies in the λ-calculus, involving reduction under abstraction, are awkward to encode in the π-calculus. These are important e.g. for optimization techniques such as partial evaluation. In the fusion calculus reduction under abstraction can be represented at no extra cost.

3. The polyadic π-calculus is a subcalculus of the fusion calculus. Therefore everything that can be done in the polyadic π-calculus can be done in the fusion calculus without added complications.

The dramatic and unexpected result is that we achieve this improvement by *simplifying* the π-calculus rather than adding features to it. Proofs about the calculus, for example in complete axiomatizations and full abstraction, actually get shorter and clearer. We mention three reasons why the fusion calculus is simpler:

1. It has only one binding operator. All other process calculi for value-passing including the π-calculus has at least two: binding of input variables and restriction of local ports.

2. It has a complete symmetry between input and output actions; they are dual versions of a single kind of communication action. In the π-calculus they are not duals because input entails a binding where output does not.

3. It has only one sensible bisimulation equivalence, where the other value-passing calculi have a proliferation into "early", "late" and "open" varieties due to different strategies to instantiate variables.

The price to be paid is the introduction of a new kind of actions, "fusion" actions. These actions are easily and cleanly incorporated into the formal framework. In the following we shall elaborate on these points, comparing with the polyadic π-calculus.

Background on the π-calculus: The π-calculus postulates a set of *names* $u, v, \ldots, z$ used to represent communication ports, values, variables etc. (more refined versions of π have type systems where these can be distinguished but that is not necessary for the development in this paper). We use the *polyadic* calculus where an output or input along a port may carry more than one object. An interaction occurs between two processes, a sender performing an output and a receiver performing an input. The effect of the interaction is local to the receiver: its input variables become instantiated with the objects emitted by the sender. An example is:

$$\overline{u}\,vw\,.\,P \mid u(xy)\,.\,Q \;\xrightarrow{\tau}\; P \mid Q\{v/x, w/y\}$$

The output prefix $\overline{u}\,vw$ sends two objects v and w along u, and the input prefix $u(xy)$, which binds x and y, receives two objects. Their interaction gives rise to a substitution which is local to Q. Additionally these prefixes imply temporal precedence: P and Q may not perform any transitions until the prefixes have been performed.

Another kind of binding occurs in the *restriction* operator: in $(\boldsymbol{\nu}x)P$ the name x in P is local to P. This means that it is distinct from any name occurring outside P, and moreover that as P and the environment evolves during execution it will remain distinct.

Other kinds of bindings have been suggested though not developed, notably a *delayed input* construct such as $u(x){:}\,P$ which is input without the temporal precedence. Here P may execute and can also at any time receive something for x along u. The binding of x extends into P which is the reason this is different from $u(x)\,.\,\mathbf{0} \mid P$.

Fusions: In the fusion calculus the input primitive does *not* bind names. The effect of an interaction is that the output objects and input objects become identified in what we call a *fusion*. There are two differences between fusions and π-calculus interactions: the effect of a fusion is not necessarily local but regulated by the use of a *scope* operator, and fusions are symmetric with respect to input-output polarities. We now explain these points.

Formally a fusion is just an equivalence relation on names signifying that names related by it should be considered equal. This effect is applicable not only to the receiver but also to the sender and to any process running in parallel with them. Therefore it can be thought of as an update of a (not explicitly represented) shared state. For example, an interaction between $\overline{u}\,vw\,.\,P$ and $u\,xy\,.\,Q$ will result in a fusion of v and x, and of w and y. We write this equivalence as $\{v = x, w = y\}$. An interaction is written

$$\overline{u}\,vw\,.\,P \mid u\,xy\,.\,Q \;\xrightarrow{\{v=x,w=y\}}\; P \mid Q$$

The fusion is placed above the arrow to signify its unbounded scope. When further transitions from the derivative $P \mid Q$ are considered, they will be treated in an environment which identifies v with x and w with y. Our definition of equivalence between processes will ensure this effect. The fact that the fusion is "global" can be seen by including further parallel processes, these will also be similarly affected by the environment, for example in

$$\overline{u}\,vw\,.\,P \mid u\,xy\,.\,Q \mid R \mid S \;\xrightarrow{\{v=x,w=y\}}\; P \mid Q \mid R \mid S$$

This ability to express global effects is the reason why the fusion calculus gains in expressiveness over the π-calculus.

The fact that the interaction is symmetric can be seen by inverting the input-output polarities of the actions:

$$u\,vw\,.\,P \mid \overline{u}\,xy\,.\,Q \;\xrightarrow{\{v=x,w=y\}}\; P \mid Q$$

The equivalence relation of the fusion here is the same as before. Therefore the designations "input" and "output" are arbitrary and a more appropriate terminology might be "action" and "co-action", though in this paper we shall retain the more familiar terms.

Scopes: The single binding operator is called *scope* and is written $(x)P$, meaning that the x in P is local. In a sense it is the common denominator of input binding and restriction in the π-calculus: an instantiation of the bound name is neither mandatory (like in input) nor impossible (like in restriction). The main idea is that scopes can be used to delimit the extent of fusions. As a first example we show how the π-calculus interaction can be emulated by scoping the input names:

$$\overline{u}\, vw \,.\, P \mid \Big((x)(y)u\, xy \,.\, Q\Big) \mid R \mid S \quad \xrightarrow{\mathbf{1}} \quad P \mid \Big(Q\{v/x, w/y\}\Big) \mid R \mid S$$

Here $\mathbf{1}$ is an inert action (formally it is just the identity fusion) and the effect of the interaction does not extend beyond the scopes. Since x and y are bound they are substituted by some non-bound names fused to them.

The increased expressiveness over the π-calculus is that scopes may be placed in arbitrary places and not only directly over the input actions. For example

$$(x)\Big(\overline{u}\, vw \,.\, P \mid (y)\Big(u\, xy \,.\, Q \mid R\Big) \mid S\Big) \quad \xrightarrow{\mathbf{1}} \quad \Big(P \mid \Big(Q \mid R\Big)\{w/y\} \mid S\Big)\{v/x\}$$

where the scopes determine the extent and direction of the resulting substitutions.

As an example of the expressiveness the delayed input is now easy to define: in $(x)(ux \mid P)$ it is possible for P to execute before ux interacts with an action of kind $\overline{u}v$, fusing x in P with v. Another example is the polyadic input of the same name twice: $(z)u\, zz \mid P$. Here any two names emitted by P along u will be fused, even if they are bound in P, for example

$$(z)u\, zz \mid \overline{u}\, vw \,.\, P \xrightarrow{\{v=w\}} P$$

A similar effect is obtained by two monadic inputs in $(z)uz \,.\, uz \mid P$

Related work: Our earlier work [PV97] attacks some of these problems in a monadic calculus where only one object is transmitted in each interaction. The main result there, the *update calculus*, unfortunately turned out not to generalize to polyadic interactions. The reason is that updates (like substitutions) are directed, and it is unclear what a polyadic update like $[a/x, b/x]$ should mean. In contrast, the polyadic fusion $\{a = x, b = x\}$ is unproblematic: it is just the equivalence equating all of a, b and x. The fusion calculus has a higher degree of symmetry and this has profound implications for our technical proofs.

Another way to obtain an input-output symmetry is in the πI-calculus developed by Sangiorgi [San96a]. πI is a subcalculus of π: output objects must be bound by restriction, so both input and output bind their objects. Through clever encodings it has been established that πI exhibits a considerable expressive power despite this limitation. In contrast, the fusion calculus achieves the same symmetry with an increase in expressiveness, by allowing both input and output to carry non-bound objects.

The problem of modelling strong lazy reductions in the λ-calculus has been considered by Fu [Fu97] and by Sangiorgi [San96a]. Both have postulated solutions though not published any analysis. Fu extends the π-calculus to something more complicated, where we instead simplify it. Sangiorgi's solution is a comparatively complicated encoding in the πI-calculus, where we instead can formulate a straightforward modification of Milner's original encoding and proofs for the (non-strong) lazy λ-calculus.

There is a host of work on the π-calculus following the original presentation [MPW92]. We here only mention that inspiration for the present paper comes from Milner's encoding of the λ-calculus [Mil92], the axiomatizations by Parrow and Sangiorgi [PS95], and Sangiorgi's "open" bisimulation equivalence [San96b]. Milner's action calculi [Mil96] also aim to find a conceptually more fundamental formalism, though in a different direction.

Future work: Using the fusion calculus to encode the γ-calculus [Smo94, VP96] we should get a simple model which can be used to relate mobile processes to concurrent constraints. The possibility of a higher-order fusion calculus should be explored; intuitively, process passing breaks the symmetry of communication in the first-order fusion calculus. To model full λ-calculus reduction, we will look at extensions to multiway communication. Such extensions are probably easier in the fusion calculus than in the π-calculus where the distinction between input and output is a complication. It should be interesting to look at the relation between fusion calculus and the πI-calculus. Finally, analysis algorithms and tools such as the MWB [VM94] should be adapted and extended to handle the fusion calculus.

Main results and overview of the rest of the paper: In Section 2 we define the syntax and semantics of the fusion calculus in a way that has now become standard: first we give a structural congruence saying which agents are declared to have the same transitions, and then a labelled transition system where the labels are either communication actions or fusions. We also formally compare the fusion calculus with the π-calculus. In Section 3 we define bisimulation equivalence as the largest sensible (strong) bisimulation equivalence, and show that the largest congruence contained in it has a pleasant characterization through a kind of substitution-closed bisimulations. The resulting congruence, called hyperequivalence, is stronger than any of the equivalences which have been suggested for the π-calculus. In Section 4 we proceed to give complete axiomatizations in two versions, with and without the mismatch operator. Finally in Section 5 we demonstrate how the strong lazy reduction strategy in the λ-calculus has a straightforward encoding.

2 Syntax and Semantics

We assume an infinite set $\mathcal{N}$ of *names* ranged over by $u, v, \ldots, z$. Like in the π-calculus, names represent communication ports, which are also the values transmitted. We write $\tilde{x}$

for a (possibly empty) finite sequence $x_1 \cdots x_n$ of names. φ ranges over total equivalence relations over $\mathcal{N}$ (i.e. equivalence relations with $\mathrm{dom}(\varphi) = \mathcal{N}$) with only finitely many non-singular equivalence classes. We write $\{\tilde{x} = \tilde{y}\}$ to mean the smallest such equivalence relation relating each x_i with y_i, and write $\mathbf{1}$ for the identity relation.

Definition 1 *The* free actions, *ranged over by α are defined by*

$$\begin{array}{rlll} \alpha ::= & u\tilde{x} & & \text{(Input)} \\ & \overline{u}\tilde{x} & & \text{(Output)} \\ & \varphi & & \text{(Fusion)} \end{array}$$

Input and output actions are collectively called *communication* actions. In these $\tilde{x}$ are the *objects* of the action, and the name u is the *subject*. We write a to stand for either u or $\overline{u}$, thus $a\tilde{x}$ is the general form of a communication action. Fusion actions have neither subject nor objects.

Definition 2 *The* agents, *ranged over by $P, Q, \ldots$, are defined by*

$$\begin{array}{rlll} P ::= & \mathbf{0} & & \text{(Inaction)} \\ & \alpha \,.\, Q & & \text{(Prefix)} \\ & Q + R & & \text{(Summation)} \\ & Q \mid R & & \text{(Composition)} \\ & (x)Q & & \text{(Scope)} \end{array}$$

Prefixing an agent Q means that the agent must perform the action α before acting like Q. We often omit a trailing $\mathbf{0}$ and write α for $\alpha \,.\, \mathbf{0}$ if no confusion can arise. Like in the π-calculus, Summation is alternative choice and Composition lets agents act in parallel.

The Scope $(x)Q$ limits the scope of x to Q; no visible communication action of $(x)Q$ can have x as its subject, and fusion effects with respect to x are limited to Q. Restriction and input binding of π-calculus can be recovered as special cases of Scope.

The name x is said to be *bound* in $(x)P$. We write $(\tilde{x})P$ for $(x_1) \cdots (x_n)P$. The *free names* in P, denoted $\mathrm{fn}(P)$, are the names in P with a non-bound occurrence, here the names occurring in the fusion φ is defined to be the names in the non-singular equivalence classes, i.e. in the relation $\varphi - \mathbf{1}$. As usual we will not distinguish between alpha-variants of agents, i.e., agents differing only in the choice of bound names.

The action of a transition may be free or bound:

Definition 3 *A* bound *action is of the form $(\tilde{z})a\tilde{x}$ where $|\tilde{z}| > 0$ and all elements in $\tilde{z}$ are also in $\tilde{x}$. The* actions, *ranged over by γ, consist of the free actions and the bound actions.*

Note that there are no bound fusion actions. In the bound actions above, $\tilde{z}$ are the *bound objects* and the elements in $\tilde{x}$ that are not in $\tilde{z}$ are the *free objects*. Free actions have no bound objects. We further write $\mathrm{n}(\gamma)$ to mean all names occurring in γ (i.e., also including the subject of communication actions and the names in non-singular equivalence classes in fusion actions).

PREF $\dfrac{-}{\alpha\,.\,P \xrightarrow{\alpha} P}$	SUM $\dfrac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$
PAR $\dfrac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$	COM $\dfrac{P \xrightarrow{u\tilde{x}} P',\ \ Q \xrightarrow{\overline{u}\tilde{y}} Q',\ \ \lvert\tilde{x}\rvert = \lvert\tilde{y}\rvert}{P \mid Q \xrightarrow{\{\tilde{x}=\tilde{y}\}} P' \mid Q'}$
PASS $\dfrac{P \xrightarrow{\alpha} P',\ z \notin \mathrm{n}(\alpha)}{(z)P \xrightarrow{\alpha} (z)P'}$	SCOPE $\dfrac{P \xrightarrow{\varphi} P',\ z\,\varphi\,x, x \neq z}{(z)P \xrightarrow{\varphi\restriction z} P'\{x/z\}}$
OPEN $\dfrac{P \xrightarrow{(\tilde{y})a\,\tilde{x}} P', z \in \tilde{x} - \tilde{y}, a \notin \{z, \overline{z}\}}{(z)P \xrightarrow{(z\tilde{y})a\,\tilde{x}} P'}$	

Table 1: The Fusion Calculus: Laws of action.

For convenience we define $\varphi \restriction z$ to mean $\varphi \cap (\mathcal{N} - \{z\})^2 \cup \{(z,z)\}$, i.e., the equivalence relation φ with all references to z removed (except for the identity). For example, $\{x = z, z = y\} \restriction z = \{x = y\}$, and $\{x = y\} \restriction y = \mathbf{1}$.

We now define a structural congruence which equates all agents we will never want to distinguish for any semantic reason, and then use this when giving the transitional semantics.

Definition 4 *The* structural congruence, $\equiv$, *between agents is the least congruence satisfying the abelian monoid laws for Summation and Composition (associativity, commutativity and* $\mathbf{0}$ *as identity), and the scoping laws*

$$(x)\mathbf{0} \equiv \mathbf{0}, \quad (x)(y)P \equiv (y)(x)P, \quad (x)(P+Q) \equiv (x)P + (x)Q$$

and also the scope extension *law* $P \mid (z)Q \equiv (z)(P \mid Q)$ *where* $z \notin \mathrm{fn}(P)$.

Definition 5 *The family of* transitions $P \xrightarrow{\gamma} Q$ *is the least family satisfying the laws in Table 1. In this definition structurally equivalent agents are considered the same, i.e., if* $P \equiv P'$ *and* $Q \equiv Q'$ *and* $P \xrightarrow{\gamma} Q$ *then also* $P' \xrightarrow{\gamma} Q'$.

Use of the SCOPE rule entails a substitution of the scoped name z for a nondeterministically chosen name x related to it by φ. For the purpose of the equivalence defined below in Section 3 it will not matter which such x replaces z. The only rule dealing with bound actions is OPEN. Using structural congruence, pulling the relevant scope to top level, we can still infer e.g. $P \mid (x)ayx\,.\,Q \stackrel{(x)ayx}{\longrightarrow} P \mid Q$ using PREF and OPEN (provided $x \notin \mathrm{fn}(P)$, otherwise an alpha-conversion is necessary). It is clear that for the purpose of the semantics, fusion prefixes can be regarded as derived forms since $\{\tilde{y} = \tilde{z}\}\,.\,P$ has exactly the same transitions as $(u)(\overline{u}\tilde{y}\,.\,\mathbf{0} \mid u\tilde{z}\,.\,P)$ when $u \notin \mathrm{fn}(P)$. (In the same way the τ prefix can be regarded as derived in CCS and in the π-calculus.)

The following examples should be noted on how scoping delimits the effects of fusion:

$$\begin{array}{rcl}
P \mid \overline{u}y\,.\,Q \mid (z)uz\,.\,R & \stackrel{\mathbf{1}}{\longrightarrow} & P \mid Q \mid R\{y/z\} \\
P \mid (z)(\overline{u}y\,.\,Q \mid uz\,.\,R) & \stackrel{\mathbf{1}}{\longrightarrow} & P \mid (Q \mid R)\{y/z\} \\
(z)(P \mid \overline{u}y\,.\,Q \mid uz\,.\,R) & \stackrel{\mathbf{1}}{\longrightarrow} & (P \mid Q \mid R)\{y/z\} \\
(z)(P \mid \overline{u}xy\,.\,Q \mid uzw\,.\,R) & \stackrel{\{y=w\}}{\longrightarrow} & (P \mid Q \mid R)\{x/z\}
\end{array}$$

The gain in expressiveness over the π-calculus lies in this possibility to decouple the input action from the binding of the input object. As an example we can define an agent $(\tilde{z})(u\tilde{z}\,.\,\mathbf{0} \mid P)$ where the input is *delayed*, in the sense that it may happen at any time:

$$\begin{array}{llll}
(\tilde{z})(u\tilde{z}\,.\,\mathbf{0} \mid P) & \stackrel{\gamma}{\longrightarrow} & (\tilde{z})(u\tilde{z}\,.\,\mathbf{0} \mid P') & \text{if } P \stackrel{\gamma}{\longrightarrow} P' \text{ and } \tilde{z} \cap \mathrm{n}(\gamma) = \emptyset \\
(\tilde{z})(u\tilde{z}\,.\,\mathbf{0} \mid P) & \stackrel{(\tilde{z})u\tilde{z}}{\longrightarrow} & P & \text{for any } P
\end{array}$$

This agent has all transitions from P not involving names in $\tilde{z}$, and in any derivative so obtained it has the bound input transition $(\tilde{z})u\tilde{z}$.

As a final example assume $z \notin \mathrm{fn}(P)$ and consider the agent

$$(z)uz\,.\,vz\,.\,P$$

There are *two* inputs with object z under the same scope. The agent can interact with an environment that emits something along u,

$$(z)uz\,.\,vz\,.\,P \mid \overline{u}x\,.\,Q \quad \stackrel{\mathbf{1}}{\longrightarrow} \quad vx\,.\,P \mid Q$$

and will proceed input something for x along v. In other words, the object to be *replaced* in an interaction along v is first transmitted along u. This has no immediate counterpart in the π-calculus.

The polyadic π-calculus [Mil91] is isomorphic to a subcalculus of the fusion calculus, formed by omitting fusion prefixes and by requiring all input prefixes $u\tilde{x}\,.\,P$ to occur immediately under a scope $(\tilde{x})$ of the objects in the input prefix; the objects must be pairwise distinct. Call this subcalculus f_π. The isomorphism $\leftrightarrow$ between f_π and π is simply the homomorphic extension of

$$\begin{array}{lll}
(\tilde{x})u\tilde{x}\,.\,P & \leftrightarrow & u(\tilde{x})\,.\,P \\
(z)P & \leftrightarrow & (\boldsymbol{\nu} z)P \quad \text{if } P \text{ is not a free input prefix}
\end{array}$$

The rationale behind this is as follows. Consider a scope $(x)Q$ in f_π. If it is immediately over an input prefix, $Q = ux\,.\,P$, then before anything happens in P something must be received along u to replace x in P, precisely as the effect of a π-calculus input prefix $u(x) \,.\, P$. On the other hand, if it is not immediately over an input prefix, $Q \neq ux \,.\, P$, then there cannot be any input prefix inside Q for that particular x. Because of the syntactic restrictions in f_π, a free input ux not at the top level of Q must be enclosed by another scope (x), making the object different from the x in $(x)Q$. So this x cannot occur in an input and there are no fusion prefixes, therefore x will remain distinct from other names throughout execution of the agent. This is precisely the effect of $(\boldsymbol{\nu} x)Q$.

We identify actions of the π-calculus and of f_π as follows: $u(\tilde{x}) = (\tilde{x})u\tilde{x}$, $(\boldsymbol{\nu}\tilde{z})\overline{u}\,\tilde{y} = (\tilde{z})\overline{u}\,\tilde{y}$ and $\tau = \mathbf{1}$. We can then formalize the result as

Theorem 1 *For P a π-calculus agent and Q an f_π-calculus agent such that $P \xrightarrow{\gamma} P'$,*

1. *If $P \leftrightarrow Q$ then $Q \xrightarrow{\gamma} Q'$ and $P' \leftrightarrow Q'$.*
2. *If $P \leftrightarrow Q$ then $P \xrightarrow{\gamma} P'$ and $P' \leftrightarrow Q'$.*

PROOF: By induction on the depth of inference of the transition of P and Q, respectively.

We show one case, where $P \equiv P_1 \mid P_2$, $P_1 \xrightarrow{(\tilde{x})u} P_1'$, $P_2 \xrightarrow{(\tilde{z})\overline{u}\tilde{y}} P_2'$, and $P \xrightarrow{\mathbf{1}} (\tilde{z})(P_1'\{\tilde{y}/\tilde{x}\} \mid P_2')$.

Then $Q \equiv Q_1 \mid Q_2$ and by induction $Q_1 \xrightarrow{(\tilde{x})u\tilde{x}} Q_1'$, $Q_2 \xrightarrow{(\tilde{z})\overline{u}\tilde{y}} Q_2'$ and we want to show that $Q \xrightarrow{\mathbf{1}} (\tilde{z})(Q_1'\{\tilde{y}/\tilde{x}\} \mid Q_2')$. We observe that if a fusion agent $R \xrightarrow{(\tilde{x})a\tilde{y}} R'$, then by scope extrusion $R \equiv (\tilde{x})R_0$ and $R_0 \xrightarrow{a\tilde{y}} R'$, and we use this to write Q as $(\tilde{z})(\tilde{x})(Q_1'' \mid Q_2'')$, where $Q_1'' \xrightarrow{u\tilde{x}} Q_1'$ and $Q_2'' \xrightarrow{\overline{u}\tilde{y}} Q_2'$. Then $Q'' \equiv Q_1'' \mid Q_2'' \xrightarrow{\{\tilde{x}=\tilde{y}\}} Q_1' \mid Q_2'$ by COM, and $(\tilde{x})Q'' \xrightarrow{\mathbf{1}} (Q_1' \mid Q_2')\{\tilde{y}/\tilde{x}\}$ by SCOPE, but by alpha conversion $\tilde{x} \cap \mathrm{fn}(Q_2) = \emptyset$, so the substitution has no effect on Q_2'. Finally $(\tilde{z})(\tilde{x})Q'' \equiv Q \xrightarrow{\mathbf{1}} (Q_1'\{\tilde{y}/\tilde{x}\} \mid Q_2')$ as we wanted. □

The πI-calculus by Sangiorgi [San96a] is a subset of the π-calculus formed by requiring all output prefixes $\overline{u}\tilde{x}$ to occur immediately below a restriction $(\boldsymbol{\nu}\tilde{x})$ of the objects. Therefore that calculus is isomorphic to the subcalculus of f_π requiring that *all* prefixes — not merely input prefixes — occur under a scope of the object. As Sangiorgi has remarked the πI-calculus has a pleasing symmetry between input and output actions. The fusion calculus has the same kind of symmetry, achieved by extending rather than restricting the π-calculus, as implied by the following theorem where $\overline{P}$ $(\overline{\gamma})$ stands for P (γ) with all input/output polarities inverted:

Theorem 2 $P \xrightarrow{\gamma} Q$ *iff* $\overline{P} \xrightarrow{\overline{\gamma}} \overline{Q}$.

PROOF: By induction on the depth of deduction, using the fact that no rule refers to the polarities except the COM rule, which is symmetric. □

3 Equivalence and Congruence

Our notion of equivalence uses the well known idea of bisimulation. In essence, for two agents to be bisimilar, each transition from one agent must be mimicked from the other, reaching agents which are again bisimilar. In calculi where transitions can contain placeholders (such as variables) for values, care has been taken to properly define what "again bisimilar" means. For example, in the π-calculus there are varieties such as *early*, *late* [MPW92], and *open* [San96b] bisimulation. These differ in the use of universal quantification over the names, and each leads to a different congruence.

In the fusion calculus this diversity is mercifully absent; there appears to be only one sensible bisimulation congruence. The reason is that the fusion actions allow us to formulate more powerful contexts than what is possible in the π-calculus. Fusion actions can enforce a global substitution of *any* name, whereas in the π-calculus no action can enforce a substitution on a name bound by a restriction. Therefore a congruence for the fusion calculus must be closed under arbitrary substitutions in a way which will be made precise below.

Let us now consider the proper notion of bisimulation, after defining how substitutions and fusions should agree. We use σ to range over substitutions of names.

Definition 6 *A substitution σ* agrees with *the fusion φ if $\forall x, y : x\,\varphi\, y \Leftrightarrow \sigma(x) = \sigma(y)$.*

Definition 7 *A* simulation *is a binary relation $\mathcal{S}$ between agents such that $P\ \mathcal{S}\ Q$ implies:*

If $P \xrightarrow{\gamma} P'$ with the bound objects of γ not free in Q, then

$$Q \xrightarrow{\gamma} Q' \textit{ and } \begin{cases} P'\ \mathcal{S}\ Q' & \textit{if } \gamma \textit{ is a communication action} \\ P'\sigma\ \mathcal{S}\ Q'\sigma & \textit{if } \gamma \textit{ is a fusion action, for some } \sigma \textit{ which agrees with } \gamma \end{cases}$$

A bisimulation *is a relation $\mathcal{S}$ such that both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are simulations. P is bisimilar to Q, written $P \dot{\sim} Q$, if $P\ \mathcal{S}\ Q$ for some bisimulation $\mathcal{S}$.*

The interesting point in this definition is the treatment of fusion actions. A fusion $\{x = y\}$ represents an obligation to make x and y equal everywhere. Therefore, if γ above is such a fusion, it only makes sense to relate P' and Q' when a substitution $\{y/x\}$ or $\{x/y\}$ has been performed. Note that it does not matter which substitution we choose, since $P\{x/y\} \dot{\sim} Q\{x/y\}$ implies $P\{y/x\} \dot{\sim} Q\{y/x\}$, by the simple fact that $P\{x/y\}\{y/x\} \equiv P\{y/x\}$ and that bisimulation is closed under injective substitutions.

Consider the following examples:

$$\begin{array}{rcl} y \mid \overline{z} & \dot{\sim} & y\,.\,\overline{z} + \overline{z}\,.\,y \\ y \mid \overline{y} & \dot{\not\sim} & y\,.\,\overline{y} + \overline{y}\,.\,y \\ y \mid \overline{y} & \dot{\sim} & y\,.\,\overline{y} + \overline{y}\,.\,y + \mathbf{1} \\ \{y = z\}\,.\,(y \mid \overline{z}) & \dot{\sim} & \{y = z\}\,.\,(y\,.\,\overline{z} + \overline{z}\,.\,y + \mathbf{1}) \end{array}$$

The third example implies the fourth even though $y \mid \overline{z}$ is not bisimilar to $y\,.\,\overline{z} + \overline{z}\,.\,y + \mathbf{1}$. In fact, we also have

$$\{y = z\}\,.\,y\,.\,\mathbf{0} \dot{\sim} \{y = z\}\,.\,z\,.\,\mathbf{0}$$

Thus we argue that any sensible bisimulation-like equivalence must contain bisimilarity. But bisimilarity is not a congruence. An easy example is that $x \mid \overline{y}$ is bisimilar to $x\,.\,\overline{y} + \overline{y}\,.\,x$, but prefixing both with $\{x = y\}$ will result in non bisimilar agents. We therefore look for the largest congruence included in bisimilarity. This is achieved by closing the definition of bisimulation under arbitrary substitutions.

Definition 8 *A* hyperbisimulation *is a substitution closed bisimulation, i.e., a bisimulation $\mathcal{S}$ with the property that $P \mathcal{S} Q$ implies $P\sigma \mathcal{S} Q\sigma$ for any substitution σ. Two agents P and Q are* hyperequivalent*, written $P \sim Q$, if they are related by a hyperbisimulation.*

Theorem 3 *Hyperequivalence is a congruence.*

Proof: Through a case analysis over the operators in the calculus. All cases are simple. □

Note that the special case for fusion actions in the definition of bisimulation is significant even though hyperbisimulation closes under all substitutions! The point is that $P' \mathcal{S} Q'$ would be a too *strong* requirement if $\gamma = \{y = z\}$, since it would require bisimulation also under the substitutions which do not equate z and y.

It is interesting to relate hyperequivalence to open equivalence, which is a congruence in the π-calculus and hence also in f_π. For a definition of open we refer to [San96b], but even without the details of the definition a reader might appreciate the following arguments. Broadly speaking, for the purpose of hyperbisimulation, *all* names are considered universally quantified *after every transition*. That is, if we for a moment disregard the fusion actions (which are not part of the π-calculus), hyperbisimulation requires, if $P \mathcal{S} Q$ and $P \xrightarrow{\gamma} P'$, that $Q \xrightarrow{\gamma} Q'$ and $P'\sigma \mathcal{S} Q'\sigma$ *for all substitutions σ*. For late equivalence the corresponding requirement would be *for all substitutions σ involving a bound input name in γ*. For open equivalence the requirement would be *for all substitutions σ except those involving a bound output name in γ*.

In the π-calculus without the restriction operator the definitions of open and hyper would coincide. But with it hyper is strictly stronger, as evidenced by the f_π-calculus agents

$$(x\,y)\overline{u}\,xy\,.\,(x \mid \overline{y}) \qquad \text{and} \qquad (x\,y)\overline{u}\,xy\,.\,(x\,.\,\overline{y} + \overline{y}\,.\,x)$$

These are open equivalent but not hyperequivalent. To see this, after two bound output transitions the agents become

$$x \mid \overline{y} \qquad \text{and} \qquad x\,.\,\overline{y} + \overline{y}\,.\,x$$

and these are not bisimilar under the substitution $\{x/y\}$. This substitution, which affects names previously emitted as bound objects, is not relevant for open equivalence.

Now put the two agents in parallel with $(z)u\,zz$ (note that this agent is not in f_π because the two input objects are not distinct). There is then one more **1** transition from

$$(z)u\,zz \mid (x\,y)\overline{u}\,xy\,.\,(x \mid \overline{y})$$

than from

$$(z)u\,zz \mid (x\,y)\overline{u}\,xy\,.\,(x\,.\,\overline{y} + \overline{y}\,.\,x)$$

In essence, the parallel context $(z)u\, zz$ will enforce a fusion of any two names emitted along u. This demonstrates that open equivalence is not a congruence in the fusion calculus, and that this context cannot be encoded compositionally in the π-calculus. A similar argument holds for the monadic $(z)uz\,.\,uz$.

Even though open is not a congruence there might conceivably exist some other congruences larger than hyper, but our next theorem demonstrates that this is impossible. This result confirms that hyperequivalence is the only "interesting" congruence.

Theorem 4 *Hyperequivalence is the largest congruence in bisimilarity.*

The rest of this section is devoted to the proof.

We construct a context which given a set of names N can perform any substitutions of names in N, and capture all bound names emitted on a name in N.

Let N be any finite set of names, and $\mathcal{F}_N^a = \{s, r_x, e, d, \tilde{u} : x \in N, |\tilde{u}| \leq a\}$ consist of fresh names. We write $N + w$ for $N \cup w$ and $N - w$ for $N \setminus w$.

Define

$$\begin{array}{rcl} S_N^0(a) & = & \mathbf{0} \\ S_N^k(a) & = & \displaystyle\sum_{x,y \in N} s\,.\,\{x = y\}\,.\,d\,.\,S_N^{k-1}(a) \\ & + & \displaystyle\sum_{x \in N} r_x\,.\sum_{|\tilde{u}|<a, \tilde{u}\cap N=\emptyset} (\tilde{u})x\tilde{u}\,.\,\overline{e}\,\tilde{u}\,.\,S_{N+\tilde{u}}^{k-1}(a) \\ & + & \displaystyle\sum_{x \in N} \overline{r_x}\,.\sum_{|\tilde{u}|<a, \tilde{u}\cap N=\emptyset} (\tilde{u})\overline{x}\,\tilde{u}\,.\,e\tilde{u}\,.\,S_{N+\tilde{u}}^{k-1}(a) \end{array}$$

The first lemma states that bisimulation in an S_N^k context is closed under substitution.

Lemma 5 *For a fixed N and any k, and a the maximal arity of communication actions in P and Q, assume $S_N^k(a) \mid P \mathrel{\dot\sim} S_N^k(a) \mid Q$. Then $(S_N^{k-1}(a) \mid P)\{x/y\} \mathrel{\dot\sim} (S_N^{k-1}(a) \mid Q)\{x/y\}$ for all $x, y \in N$.*

PROOF: Reducing the first term of $S_N^k(a)$,

$$S_N^k(a) \mid P \xrightarrow{s} \xrightarrow{\{x=y\}} \xrightarrow{d} S_N^{k-1}(a) \mid P$$

which can only be imitated by $S_N^k(a) \mid Q$ by reducing $S_N^k(a)$ the same way. So after the first action, $\{x = y\}\,.\,d\,.\,S_N^{k-1}(a) \mid P \mathrel{\dot\sim} \{x = y\}\,.\,d\,.\,S_N^{k-1}(a) \mid Q$, so $(d\,.\,S_N^{k-1}(a) \mid P)\{x/y\} \mathrel{\dot\sim} (d\,.\,S_N^{k-1}(a) \mid Q)\{x/y\}$ and finally $(S_N^{k-1}(a) \mid P)\{x/y\} \mathrel{\dot\sim} (S_N^{k-1}(a) \mid Q)\{x/y\}$. □

The next lemmas state that an S_N^k context can capture any bound names in actions, while not affecting free actions:

Lemma 6 *For any N and any k, and a the maximal arity of communication actions in P, and α free,* subj$(\alpha) \notin \mathcal{F}_N^a$*,*

1. *if $P \xrightarrow{\alpha} P'$ then $S_N^k(a) \mid P \xrightarrow{\alpha} S_N^k(a) \mid P'$*
2. *if $S_N^k(a) \mid P \xrightarrow{\alpha} R$ then $P \xrightarrow{\alpha} P'$ and $R \equiv S_N^k(a) \mid P'$*

PROOF: 1. Trivial; 2. All initial actions of $S^k_N(a)$ are in $\mathcal{F}^a_N$. $\square$

Lemma 7 *For any N and any k, a the maximal arity of communication actions in P, $x \in N$ and $\tilde{w} \subseteq \tilde{y}$, $S^k_N(a) \mid P \xrightarrow{r_x}\xrightarrow{\mathbf{1}}\xrightarrow{(\tilde{w})\overline{e}\,\tilde{y}} R$ iff $R \equiv S^{k-1}_{N+\tilde{y}}(a) \mid P'$ and $P \xrightarrow{(\tilde{w})\overline{x}\,\tilde{y}} P'$*

PROOF: There is only one way for $S^k_N(a) \mid P$ to perform an r_x action;

$$S^k_N(a) \mid P \xrightarrow{r_x} \sum_{|\tilde{u}|<a, \tilde{u}\cap N=\emptyset} (\tilde{u})x\,\tilde{u}\,.\,\overline{e}\,\tilde{u}\,.\,S^{k-1}_{N+\tilde{u}}(a) \mid P \equiv R_0$$

and the only way for R_0 to complete the action sequence is if $P \xrightarrow{(\tilde{w})\overline{x}\,\tilde{y}} P'$; then

$$R_0 \xrightarrow{\mathbf{1}} (\tilde{w})(\overline{e}\,\tilde{y}\,.\,S^{k-1}_{N+\tilde{y}}(a) \mid P') \xrightarrow{(\tilde{w})\overline{e}\,\tilde{y}} S^{k-1}_{N+\tilde{y}}(a) \mid P'$$

as required. $\square$

Lemma 8 *For any N and any k, a the maximal arity of communication actions in P, $x \in N$ and $\tilde{w} \subseteq \tilde{y}$, $S^k_N(a) \mid P \xrightarrow{\overline{r_x}}\xrightarrow{\mathbf{1}}\xrightarrow{(\tilde{w})e\,\tilde{y}} R$ iff $R \equiv S^{k-1}_{N+\tilde{y}}(a) \mid P'$ and $P \xrightarrow{(\tilde{w})x\,\tilde{y}} P'$*

PROOF: Directly by Lemma 7 and Theorem 2. $\square$

In the Lemma below, we write $\mathrm{init}(P)$ for the set of initial actions of P, $\{\gamma : P \xrightarrow{\gamma}\}$.

Lemma 9 *If $P \mid (R+R') \dot{\sim} Q \mid (R+R')$ then $P \mid R \dot{\sim} Q \mid R$, where $\mathrm{init}(R)\cap\mathrm{init}(R') = \emptyset$, and $\mathrm{n}(\mathrm{init}(R,R')) \cap \mathrm{fn}(P,Q) = \emptyset$.*

PROOF: By showing that $\mathcal{S} = \{(P \mid R, Q \mid R) : P \mid (R+R') \dot{\sim} Q \mid (R+R'), \mathrm{init}(R) \cap \mathrm{init}(R') = \emptyset, \mathrm{n}(\mathrm{init}(R,R')) \cap \mathrm{fn}(P,Q) = \emptyset\}$ is a bisimulation. $\square$

Lemma 10 *$(S^k_N(a) \mid P)\{x/y\} \dot{\sim} (S^k_N(a) \mid Q)\{x/y\}$ implies $S^k_{N-y} \mid P\{x/y\} \dot{\sim} S^k_{N-y} \mid Q\{x/y\}$, if $\{x,y\} \subseteq N$.*

PROOF: It is easy to see that if $y \notin N$ and $x \in N$ and $k > 0$,

$$\begin{aligned} S^k_{N+y}(a) \quad \equiv \quad & S^k_N(a) + \sum_{z\in N} s\,.\,\{z=y\}\,.\,d\,.\,S^{k-1}_{N+y}(a) \\ + \quad & r_y\,.\sum_{|\tilde{u}|<a,\tilde{u}\cap(N+y)=\emptyset} (\tilde{u})y\tilde{u}\,.\,\overline{e}\,\tilde{u}\,.\,S^{k-1}_{N+y+\tilde{u}}(a) \\ + \quad & \overline{r_y}\,.\sum_{|\tilde{u}|<a,\tilde{u}\cap(N+y)=\emptyset} (\tilde{u})\overline{y}\tilde{u}\,.\,e\,\tilde{u}\,.\,S^{k-1}_{N+y+\tilde{u}}(a) \end{aligned}$$

(this can be shown by induction on k). Applying the substitution $\{x/y\}$ on this equation, we get

$$\begin{aligned} S^k_{N+y}(a)\{x/y\} \quad \equiv \quad & S^k_N(a) + \sum_{z\in N} s\,.\,\{z=x\}\,.\,d\,.\,S^{k-1}_{N+y}(a)\{x/y\} \\ + \quad & r_y\,.\sum_{|\tilde{u}|<a,\tilde{u}\cap(N+y)=\emptyset} (\tilde{u})x\tilde{u}\,.\,\overline{e}\,\tilde{u}\,.\,S^{k-1}_{N+y+\tilde{u}}(a)\{x/y\} \\ + \quad & \overline{r_y}\,.\sum_{|\tilde{u}|<a,\tilde{u}\cap(N+y)=\emptyset} (\tilde{u})\overline{x}\tilde{u}\,.\,e\,\tilde{u}\,.\,S^{k-1}_{N+y+\tilde{u}}(a)\{x/y\} \end{aligned}$$

where the summands prefixed by s are already in $S^k_N(a)$, and can be removed up to structural congruence. Since $\{r_y, \overline{r_y}\} \cap \mathrm{init}(S^k_N) = \emptyset$ and $r_y \notin \mathrm{fn}(P,Q)$, we can use Lemma 9 to prove that indeed the implication of this lemma holds. $\square$

The following Lemma states that if two agents are bisimilar in an S^k_N context, then they must also be hyperequivalent:

Lemma 11
$\mathcal{S} \equiv \{(P,Q) : a = \mathrm{am}(P,Q), \forall N : \mathrm{fn}(P,Q) \subseteq N, \forall k : S^k_N(a) \mid P \dot{\sim} S^k_N(a) \mid Q\}$ *is a hyperbisimulation, where* $\mathrm{am}(P,Q)$ *is the maximal arity of communication actions in* P *or* Q*.*
PROOF:

1. $\mathcal{S}$ is closed under substitution. If $(P,Q) \in \mathcal{S}$ then $(S^k_N(a) \mid P) \dot{\sim} (S^k_N(a) \mid Q)$, and by Lemma 5 we have $(S^k_N(a) \mid P)\{x/y\} \dot{\sim} (S^k_N(a) \mid Q)\{x/y\}$, and by Lemma 10 we have $S^k_{N-y}(a) \mid P\{x/y\} \dot{\sim} S^k_{N-y}(a) \mid Q\{x/y\}$, and since $y \notin \mathrm{fn}(P\{x/y\}, Q\{x/y\})$, $(P\{x/y\}, Q\{x/y\}) \in \mathcal{S}$.

2. $\mathcal{S}$ is a bisimulation. If $(P,Q) \in \mathcal{S}$, then $S^k_N \mid P \dot{\sim} S^k_N \mid Q$ for any k and N s.t. $\mathrm{fn}(P,Q) \subseteq N$. If $k = 0$ then trivially $P \dot{\sim} Q$. If $k > 0$, then

   (a) if γ is free, then by Lemma 6 $S^k_N \mid P \xrightarrow{\gamma} S^k_N \mid P'$, and then $S^k_N \mid Q \xrightarrow{\gamma} Q''$ and by Lemma 6 $Q'' \equiv S^k_N \mid Q'$ and $Q \xrightarrow{\gamma} Q'$, and $S^k_N \mid P' \dot{\sim} S^k_N \mid Q'$, so $(P', Q') \in \mathcal{S}$.

   (b) if $\gamma = (\tilde{w})x\tilde{u}$ where all names in $\tilde{w}$ are in $\tilde{u}$, then by Lemma 7, $S^k_N \mid P \xrightarrow{\overline{r}x}\xrightarrow{\mathbf{1}}\xrightarrow{(\tilde{w})\overline{e}\tilde{u}} S^{k-1}_{N+\tilde{u}} \mid P'$ and then $S^k_N \mid Q \xrightarrow{\overline{r}x}\xrightarrow{\mathbf{1}}\xrightarrow{(\tilde{w})\overline{e}\tilde{u}} Q''$ and by Lemma 7 $Q'' \equiv S^{k-1}_{N+\tilde{u}} \mid Q'$ and $Q \xrightarrow{\gamma} Q'$, and since $\mathrm{fn}(P',Q') \subseteq N + \tilde{u}$, $(P',Q') \in \mathcal{S}$.

   (c) if $\gamma = (\tilde{w})x\tilde{u}$, then directly from Theorem 2 and the previous case, $(P',Q') \in \mathcal{S}$.

□

Now Theorem 4 is a consequence of Lemma 11, using S^k_N contexts: If $\mathcal{R}$ is a bisimulation congruence, $(P,Q) \in \mathcal{R}$ implies that for all contexts $\mathcal{C}$, $(\mathcal{C}[P], \mathcal{C}[Q]) \in \mathcal{R}$, so consider the family of contexts $\mathcal{C}[\bullet] = S^k_N(a) \mid \bullet$, for all k, $N = \mathrm{fn}(P,Q)$, and a the maximal arity of communication actions in P, Q. Then by Lemma 11, $\mathcal{R}$ is a hyperbisimulation. Thus there can not exist a larger (or "weaker") bisimulation congruence than hyperequivalence for the fusion calculus.

To relate the fusion calculus to its monadic predecessor, the *update calculus* [PV97], we regard both a monadic fusion prefix $\{x = y\} . P$ and an update prefix $[x/y] . P$ as defined by $(u)(\overline{u}\, x \mid uy . P)$ (where u is fresh).

Theorem 12 *For* P *and* Q *monadic fusion calculus processes, it holds that if* $P \sim Q$ *in the update calculus, then* $P \sim Q$ *in the fusion calculus.*
PROOF: We show that $\mathcal{S} = \{(P,Q) : P \sim Q$ in the update calculus$\}$ is a hyperequivalence in the fusion calculus. using the facts that $P \xrightarrow{\{x=y\}} P'$ iff $P \xrightarrow{[x/y]} P'$ or $P \xrightarrow{[y/x]} P'$, and if $P \xrightarrow{\mathbf{1}} P'$ in the fusion calculus, then either $P \xrightarrow{\mathbf{1}} P'$ or $P \xrightarrow{(y)[y/x]} P''$ where $P' \equiv P''\{x/y\}$ in the update calculus.

Assume $(P,Q) \in \mathcal{S}$ and $P \xrightarrow{\gamma} P'$. Then:

1. if γ is a communication action, $P \xrightarrow{\gamma} P'$ in the update calculus and $Q \xrightarrow{\gamma} Q'$ in the update calculus s.t. $P' \sim Q'$, so $Q \xrightarrow{\gamma} Q'$ in the fusion calculus and $(P', Q') \in \mathcal{S}$.
2. if $\gamma = \{x = y\}$, then either $P \xrightarrow{[x/y]} P'$ and $Q \xrightarrow{[x/y]} Q'$ s.t. $P' \sim Q'$, so $Q \xrightarrow{\{x=y\}} Q'$ and $(P', Q') \in \mathcal{S}$; or $P \xrightarrow{[y/x]} P'$ etc.
3. if $\gamma = \mathbf{1}$, then either
   (a) $P \xrightarrow{\mathbf{1}} P'$ which is a simple case handled like the communication action above, or
   (b) $P \xrightarrow{(y)[y/x]} P''$ where $P' \equiv P''\{x/y\}$, and $Q \xrightarrow{(y)[y/x]} Q''$ s.t. $P''\{y/x\} \sim Q''\{y/x\}$. Since $\sim$ is closed under substitution also in the update calculus, this means that $P''\{x/y\} \sim Q''\{x/y\}$. $Q \xrightarrow{(y)[y/x]} Q''$ implies that $Q \xrightarrow{\mathbf{1}} Q'$ in the fusion calculus and $Q' \equiv Q''\{x/y\}$. So $(Q', P') \in \mathcal{S}$.

$\square$

The reverse does not hold, since $[x/y]P \sim [y/x]P$ in the fusion calculus but not in the update calculus; we disregard the direction of the substitution when moving from update to fusion actions.

4 Axiomatization

We now introduce two additional operators in the fusion calculus: if P is an agent then so are $[x = y]P$, a *match*, and $[x \neq y]P$, a *mismatch*, with the laws of action

$$\text{MATCH} \ \frac{P \xrightarrow{\alpha} P'}{[x = x]P \xrightarrow{\alpha} P'} \qquad \text{MISM} \ \frac{P \xrightarrow{\alpha} P',\, x \neq y}{[x \neq y]P \xrightarrow{\alpha} P'}$$

These operators are found in earlier work on the π-calculus; match has been used for a sound axiomatization of late congruence [MPW92] as well as a complete axiomatization of open equivalence [San96b], and mismatch has been used for complete axiomatizations of the late and early congruence [PS95]. Below we give complete axiomatizations of hyperequivalence, both with and without mismatch.

Here and in the following we use M, N to stand for a match or a mismatch operator, and define $\mathrm{n}([x = y]) = \mathrm{n}([x \neq y]) = \{x, y\}$. We add the scope law $(x)MP \equiv M(x)P$, *if* $x \notin \mathrm{n}(M)$ to the structural congruence. We write "match sequence" for a sequence of match *and* mismatch operators, ranged over by $\tilde{M}$, $\tilde{N}$, and we say that $\tilde{M}$ *implies* $\tilde{N}$, written $\tilde{M} \Rightarrow \tilde{N}$, if the conjunction of all matches and mismatches in $\tilde{M}$ logically implies all elements in $\tilde{N}$, and that $\tilde{M} \Leftrightarrow \tilde{N}$ if $\tilde{M}$ and $\tilde{N}$ imply each other. We write $\sum_{i \in I} P_i$ for finite general summation, $P_1 + \cdots + P_n$.

For the axiomatization of hyperequivalence we subsume the fact that the equivalence is a congruence. We also use some of the laws for structural congruence (see Table 2). The

Summation		
S1	$P + \mathbf{0} = P$	
S2	$P + Q = Q + P$	
S3	$P + (Q + R) = (P + Q) + R$	
Scope		
R0	$(x)\mathbf{0} = \mathbf{0}$	
R1	$(x)(y)P = (y)(x)P$	
R2	$(x)(P + Q) = (x)P + (x)Q$	
Match and Scope		
RM1	$(x)[y = z]P = [y = z](x)P$	if $x \neq y, x \neq z$

Table 2: Axioms from structural congruence.

axioms of the fusion calculus are given in Table 3, and in Table 4 we present some derived rules (whose names start with **D**). (The names of the axioms have historical reasons.)

Definition 9 *A substitution σ* agrees with *a match sequence $\tilde{M}$, and $\tilde{M}$ agrees with σ, if for all x, y which appear in $\tilde{M}$ it holds that $\sigma(x) = \sigma(y)$ iff $\tilde{M} \Rightarrow [x = y]$.*

Of the derived rules, **DM1** follows from $P \stackrel{\mathbf{M5}}{=} [x = x]P + [x \neq x]P \stackrel{\mathbf{M4}}{=} [x = x]P + \mathbf{0} \stackrel{\mathbf{S1}}{=} [x = x]P$; **DM2** follows from $[x = y]\alpha\,.\,P \stackrel{\mathbf{M2}}{=} [x = y](\alpha\,.\,P)\{x/y\} \stackrel{\mathbf{DM1}}{=} [x = y](\alpha\,.\,[x = y]P)\{x/y\} \stackrel{\mathbf{M2}}{=} [x = y]\alpha\,.\,[x = y]P$; **DM3** is derivable because we can write $\tilde{M}$ with the matches closest to P, and then apply **M2** repeatedly. The only worry is if the accumulated substitution becomes ambiguous, e.g. if $\{x/y\}\{z/y\}$ occurs. Since σ agrees with $\tilde{M}$, this means that $\tilde{M} \Rightarrow [x = y]$ and $\tilde{M} \Rightarrow [z = y]$, but then $\tilde{M} \Rightarrow [x = z]$ as well, and a corresponding substitution must occur in σ. **DM4** follows from $M\mathbf{0} \stackrel{\mathbf{M4}}{=} M[x \neq x]\mathbf{0} \stackrel{\mathbf{M1}}{=} [x \neq x]\mathbf{0} \stackrel{\mathbf{M4}}{=} \mathbf{0}$. **DM5** can be derived from $P \stackrel{\mathbf{M5}}{=} MP + \neg MP \stackrel{\mathbf{S4}}{=} (MP + MP) + \neg MP \stackrel{\mathbf{S3}}{=} MP + (MP + \neg MP) \stackrel{\mathbf{M5}}{=} MP + P$, where $\neg[x = y] \stackrel{def}{=} [x \neq y]$ and $\neg[x \neq y] \stackrel{def}{=} [x = y]$.

The derived rules **DRM** for match and scope correspond to the mismatch cases of the axioms **RM**, which apply to matches. **DRM1** is derived along the lines of [PS95], using **R2,M3,M4,M5,RM1** and the axioms for sum and congruence. **DRM2** follows from $(x)P \stackrel{\mathbf{M5,R1}}{=} (x)[x = y]P + (x)[x \neq y]P \stackrel{\mathbf{RM2,S1}}{=} (x)[x \neq y]P$. Finally, **DF1** follows from **F1** and **M2**, while **DF2** follows from **F2** and **DF1**.

It is interesting to note that all axioms except the two for fusion are well known from previous axiomatizations of the π-calculus. For example, all the axioms from Sangiorgi's axiomatization of open bisimulation equivalence for the π-calculus [San96b] hold for hyperequivalence in the fusion calculus, except the axiom treating restriction and the weaker form of congruence wrt input prefix. Please note also that our expansion law **E** is syntactically simpler than the corresponding laws for π-calculus equivalences.

Distinction relations between names have been used in axiomatizing π-calculus equivalences in [MPW92, San96b], but are not needed in our axiomatization of fusion calculus. In

Summation		
S4	$P + P = P$	
Match		
M1	$\tilde{M}P = \tilde{N}P$	if $\tilde{M} \Leftrightarrow \tilde{N}$
M2	$[x = y]P = [x = y](P\{x/y\})$	
M3	$MP + MQ = M(P + Q)$	
M4	$[x \neq x]P = \mathbf{0}$	
M5	$P = [x = y]P + [x \neq y]P$	
Scope		
R3	$(x)\alpha\,.\,P = \alpha\,.\,(x)P$	if $x \notin \mathrm{n}(\alpha)$
R4	$(x)\alpha\,.\,P = \mathbf{0}$	if x is the subject of α
Match and Scope		
RM2	$(x)[x = y]P = \mathbf{0}$	if $x \neq y$
Fusion		
F1	$\varphi\,.\,P = \varphi\,.\,[x = y]P$	if $x\,\varphi\,y$
F2	$(z)\varphi\,.\,P = \varphi \restriction z\,.\,P$	if $z \notin \mathrm{fn}(P)$
Expansion		
E	for $P \equiv \Sigma_i M_i(\tilde{x}_i)\alpha_i.P_i$, $Q \equiv \Sigma_j N_j(\tilde{y}_j)\beta_j.Q_j$, $$P \mid Q = \sum_i M_i(\tilde{x}_i)\alpha_i\,.\,(P_i \mid Q) + \sum_j N_j(\tilde{y}_j)\beta_j\,.\,(P \mid Q_j) + \sum_{\alpha_i \mathrm{opp} \beta_j} M_i N_j(\tilde{x}_i\,\tilde{y}_j)[u_i = v_j]\{\tilde{z}_i = \tilde{w}_j\}\,.\,(P_i \mid Q_j)$$ where $\alpha_i \mathrm{opp} \beta_j$ means $\alpha_i \equiv \overline{u_i}\tilde{z}_i$ and $\beta_j \equiv v_j\tilde{w}_j$.	

Table 3: Axioms.

Match		
DM1	$[x = x]P = P$	
DM2	$[x = y]\alpha\,.\,P = [x = y]\alpha\,.\,[x = y]P$	
DM3	$\tilde{M}P = \tilde{M}(P\sigma)$	for σ agreeing with $\tilde{M}$
DM4	$M\mathbf{0} = \mathbf{0}$	
DM5	$MP + P = P$	
Match and Scope		
DRM1	$(x)[y \neq z]P = [y \neq z](x)P$	if $x \neq y, x \neq z$
DRM2	$(x)[x \neq y]P = (x)P$	if $x \neq y$
Fusion		
DF1	$\varphi\,.\,P = \varphi\,.\,(P\sigma)$	where σ agrees with φ
DF2	$(z)\varphi\,.\,P = \varphi \restriction z\,.\,(P\{w/z\})$	if $z\varphi w$ and $z \neq w$

Table 4: Derived rules.

both calculi **RM2**, **DRM2** and **DM2** are valid, saying that a new name (by Scope or Restriction) is not equal to another name in the immediately following action (**RM2,DRM2**), and that if two names are equal they stay equal (**DM2**). However, in the π-calculus the law $[x \neq y]\alpha\,.\,P \;=\; [x \neq y]\alpha\,.\,[x \neq y]P$ holds for early and late congruence [PS95], while it is not valid for hyperequivalence.

In the rest of this section we first present the completeness proof for the axiom system of Tables 2 and 3, and then give an axiom system for fusion calculus without mismatch, along with a completeness proof for that system.

We begin with some basic definitions and one Lemma from [PS95].

Definition 10 *The* depth *of an agent P, $\mathrm{d}(P)$, is defined inductively as follows:*
$\mathrm{d}(\mathbf{0}) = 0$, $\mathrm{d}(\alpha\,.\,P) = 1 + \mathrm{d}(P)$, $\mathrm{d}((\tilde{x})P) = \mathrm{d}(MP) = \mathrm{d}(P)$, $\mathrm{d}(P \mid Q) = \mathrm{d}(P) + \mathrm{d}(Q)$, $\mathrm{d}(P + Q) = \max(\mathrm{d}(P), \mathrm{d}(Q))$.

Definition 11 *A match sequence $\tilde{M}$ is* complete on *a set of names V if for some equivalence relation $\mathcal{R}$ on V, called the equivalence relation* corresponding to *$\tilde{M}$, it holds that $\tilde{M} \Rightarrow [x = y]$ iff $x \,\mathcal{R}\, y$, and $\tilde{M} \Rightarrow [x \neq y]$ iff $\neg(x \,\mathcal{R}\, y)$*

A complete match sequence $\tilde{M}$ is maximally consistent *in the sense that adding something not already implied by $\tilde{M}$ makes $\tilde{M}$ inconsistent.*

Lemma 13 ([PS95]) *Let V be a set of names and let $\tilde{M}$ be complete on V.*

1. *If $\tilde{N}$ is another match sequence with names in V, then either $\tilde{M}\tilde{N}$ is unsatisfiable or $\tilde{M}\tilde{N} \Leftrightarrow \tilde{M}$.*

2. *If $\tilde{N}$ is another match sequence complete on V such that $\tilde{M}$ and $\tilde{N}$ both agree with the same substitution σ, then $\tilde{M} \Leftrightarrow \tilde{N}$.*

PROOF:

1. Follows from the fact that $\tilde{M}$ is maximally consistent.

2. By 1 above, both $\tilde{M}\tilde{N} \Leftrightarrow \tilde{M}$ and $\tilde{N}\tilde{M} \Leftrightarrow \tilde{N}$.

□

Lemma 14 $\sum_{i \in I} \tilde{M}_i P = P$, *if each $\tilde{M}_i$ complete on $\mathrm{fn}(P)$ and all equivalence classes of $\mathrm{fn}(P)$ are in $\{\tilde{M}_i : i \in I\}$.*
PROOF: Use **M1** to rearrange match sequences, **M5** to remove match/mismatch pairs and **S** to rearrange sums. □

The completeness of our axiom system says that if two agents are hyperequivalent, we can prove this using our axioms. We show that this is true in a standard way: first we define a head normal form for agents, and show that any agent can be written on this form. We can then find exactly which part of one agent that was used to simulate the transition of the other.

Definition 12 *An agent P is in* head normal form (HNF) on V *(a finite set of names) if P is on the form*

$$\sum_{i \in I} \tilde{M}_i(\tilde{x}_i)\alpha_i \,.\, P_i$$

where for all i, $\tilde{x}_i \cap V = \emptyset$, $\tilde{x}_i \subseteq \mathrm{obj}(\alpha_i)$ and $\tilde{M}_i$ is complete on V.

Lemma 15 *For all agents P and finite V such that $\mathrm{fn}(P) \subseteq V$, there is an agent H such that $\mathrm{d}(H) \leq \mathrm{d}(P)$, H is in HNF on V, and $\vdash P = H$ from the axioms of Tables 2 and 3.*
PROOF:

1. $P \equiv \mathbf{0}$ is in HNF.

2. $P \equiv Q + R$ just requires induction.

3. $P \equiv \alpha \,.\, P'$ (where α is free):

   For each $i \in I$, let $\tilde{M}_i$ be complete on V, together corresponding to all possible equivalence classes of V, so intuitively $\vee_{i \in I} \tilde{M}_i \Leftrightarrow True$. Then by Lemma 14

   $$P = \sum_{i \in I} \tilde{M}_i \alpha \,.\, P'$$

   which is in HNF, and the depth is equal to $\mathrm{d}(P)$.

4. $P \equiv MP'$

   By induction $P' = \sum_{j \in J} \tilde{M}_j(\tilde{y}_j)\alpha_j \,.\, P'_j$, with depth $\leq \mathrm{d}(P') = \mathrm{d}(P)$, so

   $$P = M(\sum_{j \in J} \tilde{M}_j(\tilde{y}_j)\alpha_j \,.\, P'_j) \overset{\mathbf{M3}}{=} \sum_{j \in J} M\tilde{M}_j(\tilde{y}_j)\alpha_j \,.\, P'_j = \sum_{i \in I} \tilde{M}_i(\tilde{y}_i)\alpha_i \,.\, P'_i$$

   for $I \subseteq J$, using Lemma 13(1), with depth $\leq \mathrm{d}(P)$. If J is empty, $P = \mathbf{0}$ by **DM4**.

5. $P \equiv (x)P'$

   Again, $P' = \sum_{j \in J} \tilde{M}_j(\tilde{y}_j)\alpha_j \,.\, P'_j$, with depth $\leq \mathrm{d}(P') = \mathrm{d}(P)$, and

   $$P = (x)\sum_{j \in J} \tilde{M}_j(\tilde{y}_j)\alpha_j \,.\, P'_j \overset{\mathbf{R2}}{=} \sum_{j \in J}(x)\tilde{M}_j(\tilde{y}_j)\alpha_j \,.\, P'_j$$

   If J is empty, $P = \mathbf{0}$ by **R0**. Otherwise, each summand $(x)\tilde{M}_j(\tilde{y}_j)\alpha_j \,.\, P'_j$, call it R, can be reduced and rewritten to HNF, without increasing the depth, by **RM** and **R** axioms.

   (a) $R \overset{\mathbf{RM2}}{=} \mathbf{0}$, if $[x = y]$ occurs in $\tilde{M}_j$ for any y.

(b) $R \overset{\mathbf{RM1,DRM1,DRM2}}{=} \tilde{M}_j'(x)(\tilde{y}_j)\alpha_j \,.\, P_j'$, where $\tilde{M}_j'$ is $\tilde{M}_j$ without any mismatches $[x \neq y]$ for any y. If $x \in \mathrm{n}(\alpha_j)$ but $x \neq \mathrm{subj}(\alpha_j)$, we have a HNF already. If $x \notin \mathrm{n}(\alpha_j)$, we use **R1** and **R3** to push the scoping over the prefix, arriving at $\tilde{M}_j'(x)(\tilde{y}_j)\alpha_j \,.\, (x)P_j'$, which is in HNF. Otherwise, if $x = \mathrm{subj}(\alpha_j)$, we have $\tilde{M}_j'(\tilde{y}_j)\mathbf{0}$ by **R4**, and using **DM4** and **R0** repeatedly, we get $R = \mathbf{0}$.

6. $P \equiv Q \mid R$. Apply **E**, which does not increase the depth, and induction.

□

Theorem 16 *$P \sim Q$ iff $\vdash P = Q$ from the axioms of Tables 2 and 3.*
PROOF: That $\vdash P = Q$ implies $P \sim Q$ is easily verified by exhibiting the appropriate bisimulations. The other implication is more interesting:

By Lemma 15 we can assume P and Q to be in HNF on $\mathrm{fn}(P, Q)$. The proof is by induction on the depth of $P + Q$, showing that each summand of P is provably equal to a summand of Q.

If the depth is 0, $P \equiv Q \equiv \mathbf{0}$ and we are done. Otherwise, let $\tilde{M}(\tilde{x})\alpha \,.\, P'$ be a summand of P, σ a substitution agreeing with $\tilde{M}$, and $\sigma(w) = w$ for all $w \notin \mathrm{fn}(P, Q)$. Then $P\sigma \overset{((\tilde{x})\alpha)\sigma}{\longrightarrow} P'\sigma$. By alpha-conversion $\tilde{x} \cap \mathrm{fn}(Q) = \emptyset$, and by $P \sim Q$, $Q\sigma \overset{((\tilde{x})\alpha)\sigma}{\longrightarrow} Q''$, and

$$
\begin{array}{rcll}
P'\sigma & = & Q'' & \text{by induction, if } \alpha \text{ a communication action} \quad (1) \\
P'\sigma\rho & = & Q''\rho & \text{by induction, if } \alpha \text{ a fusion action} \quad (2)
\end{array}
$$

where ρ agrees with $\alpha\sigma$.

By alpha-conversion, we can write the summand of Q used to simulate the action of $P\sigma$ as $\tilde{N}(\tilde{x})\beta \,.\, Q'$. Then $\tilde{N}$ agrees with σ and $(\tilde{N}(\tilde{x})\beta \,.\, Q')\sigma \overset{((\tilde{x})\beta)\sigma}{\longrightarrow} Q'\sigma$, so $Q'' \equiv Q'\sigma$ and $((\tilde{x})\alpha)\sigma = ((\tilde{x})\beta)\sigma$, which implies

$$\alpha\sigma = \beta\sigma \tag{3}$$

since σ is identity for names not free in P, Q.

$\tilde{M}$ and $\tilde{N}$ are complete, so by Lemma 13(2) we know $\tilde{M} \Leftrightarrow \tilde{N}$. Since σ agrees with $\tilde{M}$

$$\tilde{M}(\tilde{x})\alpha \,.\, P' \overset{\mathbf{DM3}}{=} \tilde{M}((\tilde{x})\alpha \,.\, P')\sigma$$

In the case that α is a communication action,

$$
\begin{array}{rcl}
\tilde{M}((\tilde{x})\alpha \,.\, P')\sigma & = & \tilde{M}(\tilde{x})(\alpha\sigma) \,.\, (P'\sigma) \\
& \overset{(3)}{=} & \tilde{M}(\tilde{x})(\beta\sigma) \,.\, (P'\sigma) \\
& \overset{(1)}{=} & \tilde{M}(\tilde{x})(\beta\sigma) \,.\, (Q'\sigma) \\
& \overset{\mathbf{M1}}{=} & \tilde{N}(\tilde{x})(\beta\sigma) \,.\, (Q'\sigma) \\
& = & \tilde{N}((\tilde{x})\beta \,.\, Q')\sigma \\
& \overset{\mathbf{DM3}}{=} & \tilde{N}(\tilde{x})\beta \,.\, Q'
\end{array}
$$

In the case that α is a fusion action,

$$\begin{array}{rcl}
\tilde{M}((\tilde{x})\alpha\,.\,P')\sigma & \stackrel{\mathbf{DF1}}{=} & \tilde{M}(\tilde{x})(\alpha\sigma)\,.\,(P'\sigma\rho) \\
& \stackrel{(2)}{=} & \tilde{M}(\tilde{x})(\alpha\sigma)\,.\,(Q'\sigma\rho) \\
& \stackrel{\mathbf{DF1}}{=} & \tilde{M}(\tilde{x})(\alpha\sigma)\,.\,(Q'\sigma) \\
& \stackrel{(3)}{=} & \tilde{M}(\tilde{x})(\beta\sigma)\,.\,(Q'\sigma) \\
& \stackrel{\mathbf{M1}}{=} & \tilde{N}(\tilde{x})(\beta\sigma)\,.\,(Q'\sigma) \\
& = & \tilde{N}((\tilde{x})\beta\,.\,Q')\sigma \\
& \stackrel{\mathbf{DM3}}{=} & \tilde{N}(\tilde{x})\beta\,.\,Q'
\end{array}$$

Thus we have proven that each summand of P is provably equal to a summand of Q, and then it suffices to use the **S** rules to infer that P and Q are provably equal. $\square$

By dropping axioms **M4** and **M5**, and promoting **DM5** to an axiom, we get an axiomatization of hyperequivalence for the fusion calculus without mismatch, which can be proved complete along the lines of [San96b].

Call the new set of axioms $\mathcal{E}$. We proceed to prove the completeness of this axiom system. Because we no longer have mismatch, we cannot use the HNF defined in Definition 12. Instead we use a "head-only" variant of Sangiorgi's *strong normal form.*

Definition 13 *An agent P is in* mismatch-free head normal form (mHNF) *if P is on the form $\sum_{i\in I}\tilde{M}_i(\tilde{x}_i)\alpha_i\,.\,P_i$, where*

1. *$\forall i : \tilde{x}_i \cap \mathrm{fn}(P) = \emptyset$, and $\tilde{x}_i \subseteq \mathrm{obj}(\alpha_i)$*
2. *if $i \neq j$ then $\tilde{M}_i(\tilde{x}_i)\alpha_i\,.\,P_i \not\succ \tilde{M}_i(\tilde{x}_i)\alpha_i\,.\,P_i + \tilde{M}_j(\tilde{x}_j)\alpha_j\,.\,P_j$*

In the following, we write $\sigma_{\tilde{M}}$ for a substitution which agrees with $\tilde{M}$ and which is identity for all names not in $\tilde{M}$.

We know that

$$(\tilde{N}(\tilde{x})\alpha\,.\,P)\sigma_{\tilde{M}} \stackrel{(\tilde{x})\alpha\sigma_{\tilde{M}}}{\longrightarrow} P\sigma_{\tilde{M}} \quad \text{iff } \tilde{M} \Rightarrow \tilde{N} \qquad (4)$$

since $\tilde{M} \Rightarrow \tilde{N}$ iff $\sigma_{\tilde{M}}$ agrees with $\tilde{N}$, and we bear in mind that $\tilde{M} \Rightarrow \tilde{N}$ iff $\tilde{M}\tilde{N} \Leftrightarrow \tilde{M}$.

Lemma 17 *For all agents P there is an agent H such that $\mathrm{d}(H) \leq \mathrm{d}(P)$, H is in mHNF, and $\vdash P = H$ from the axioms of $\mathcal{E}$.*

PROOF: By induction on the depth of P. The base case, $P \equiv \mathbf{0}$, is trivial.

For the induction case, we write P as $P' \equiv \sum_{i\in I}\tilde{M}_i(\tilde{x}_i)\alpha_i\,.\,P_i$ (using **E** in case P is a parallel composition, distributing match and scope, pushing scope inward). The first requirement on a mHNF is handled by alpha-conversion. We need to show that if Q and

R are two summands of P' such that $Q \sim Q + R$, then $\vdash Q = Q + R$, and thus we can remove R from P'.

Assume $Q \sim Q + R$. Let $R \equiv \tilde{M}(\tilde{x})\alpha \,.\, R_1$, and write Q as $\tilde{N}(\tilde{x})\beta \,.\, Q_1$. Since hyper-equivalence is closed under substitution, $Q\sigma_{\tilde{M}} \sim Q\sigma_{\tilde{M}} + R\sigma_{\tilde{M}}$, and then

$$
\begin{array}{rcll}
(Q+R)\sigma_{\tilde{M}} & \xrightarrow{(\tilde{x})\alpha\sigma_{\tilde{M}}} & R_1\sigma_{\tilde{M}} & \\
Q\sigma_{\tilde{M}} & \xrightarrow{(\tilde{x})\beta\sigma_{\tilde{M}}} & Q_1\sigma_{\tilde{M}} & \\
\alpha\sigma_{\tilde{M}} & = & \beta\sigma_{\tilde{M}} & \qquad (5)\\
R_1\sigma_{\tilde{M}} & \sim & Q_1\sigma_{\tilde{M}} \quad \text{if } \alpha \text{ a communication action} & \qquad (6)\\
R_1\sigma_{\tilde{M}}\rho & \sim & Q_1\sigma_{\tilde{M}}\rho \quad \text{if } \alpha \text{ is a fusion action} & \qquad (7)\\
\tilde{M}\tilde{N} & \overset{(4)}{\Leftrightarrow} & \tilde{M} & \qquad (8)
\end{array}
$$

where the substitution ρ agrees with $\alpha\sigma_{\tilde{M}}$, and since $R_1\sigma_{\tilde{M}}$ or $R_1\sigma_{\tilde{M}}\rho$ and $Q_1\sigma_{\tilde{M}}$ or $Q_1\sigma_{\tilde{M}}\rho$ can be written in mHNF by induction, we can use Theorem 18, in conjunction with (6) and (7) to get

$$
\begin{array}{rcll}
R_1\sigma_{\tilde{M}} & = & Q_1\sigma_{\tilde{M}} \quad \text{if } \alpha \text{ a communication action} & \qquad (9)\\
R_1\sigma_{\tilde{M}}\rho & = & Q_1\sigma_{\tilde{M}}\rho \quad \text{if } \alpha \text{ is a fusion action} & \qquad (10)
\end{array}
$$

Now, in the case that α is a communication action,

$$
\begin{array}{rcl}
R & \equiv & \tilde{M}(\tilde{x})\alpha \,.\, R_1 \overset{\mathbf{DM3}}{=} \tilde{M}(\tilde{x})\alpha\sigma_{\tilde{M}} \,.\, R_1\sigma_{\tilde{M}}\\
 & \overset{(9)}{=} & \tilde{M}(\tilde{x})\alpha\sigma_{\tilde{M}} \,.\, Q_1\sigma_{\tilde{M}} \overset{(5)}{=} \tilde{M}(\tilde{x})\beta\sigma_{\tilde{M}} \,.\, Q_1\sigma_{\tilde{M}}\\
 & \overset{\mathbf{DM3}}{=} & \tilde{M}(\tilde{x})\beta \,.\, Q_1 \overset{(8)}{=} \tilde{M}\tilde{N}(\tilde{x})\beta \,.\, Q_1\\
 & \equiv & \tilde{M}Q
\end{array}
$$

So, $Q + R = Q + \tilde{M}Q \overset{\mathbf{DM5}}{=} Q$, and the term R can be dropped from P'.
For the case that α is a fusion action,

$$
\begin{array}{rcl}
R & \equiv & \tilde{M}(\tilde{x})\alpha \,.\, R_1 \overset{\mathbf{DM3}}{=} \tilde{M}(\tilde{x})\alpha\sigma_{\tilde{M}} \,.\, R_1\sigma_{\tilde{M}}\\
 & \overset{\mathbf{DF1}}{=} & \tilde{M}(\tilde{x})\alpha\sigma_{\tilde{M}} \,.\, R_1\sigma_{\tilde{M}}\rho \overset{(10)}{=} \tilde{M}(\tilde{x})\alpha\sigma_{\tilde{M}} \,.\, Q_1\sigma_{\tilde{M}}\rho\\
 & \overset{\mathbf{DF1}}{=} & \tilde{M}(\tilde{x})\alpha\sigma_{\tilde{M}} \,.\, Q_1\sigma_{\tilde{M}} \overset{(5)}{=} \tilde{M}(\tilde{x})\beta\sigma_{\tilde{M}} \,.\, Q_1\sigma_{\tilde{M}}\\
 & \overset{\mathbf{DM3}}{=} & \tilde{M}(\tilde{x})\beta \,.\, Q_1 \overset{(8)}{=} \tilde{M}\tilde{N}(\tilde{x})\beta \,.\, Q_1\\
 & \equiv & \tilde{M}Q
\end{array}
$$

and also in this case $Q + R = Q + \tilde{M}Q = Q$. $\square$

Given this result, we can concentrate the proof of completeness on agents in mHNF.

Theorem 18 *If P and Q contain no mismatch operators, then $P \sim Q$ iff $\vdash P = Q$ from the axioms in $\mathcal{E}$.*
PROOF: That $\vdash P = Q$ implies $P \sim Q$ is easily verified by exhibiting the appropriate bisimulations. We study the other implication in more detail.

The proof goes by induction on $\mathrm{d}(P+Q)$. The base case is trivial.

In the induction case, we show that each summand of P (which can be written in mHNF) can be transformed into a summand of Q (also written in mHNF).

Let $P_1 \equiv \tilde{M}(\tilde{x})\alpha\,.\,P_1'$ be a summand of P such that $P\sigma_{\tilde{M}} \xrightarrow{(\tilde{x})\alpha\sigma_{\tilde{M}}} P_1'\sigma_{\tilde{M}}$. Let $Q_1 \equiv \tilde{N}(\tilde{x})\beta\,.\,Q_1'$ be the summand of Q simulating this transition. Then $\tilde{M} \Rightarrow \tilde{N}$, and $\sigma_{\tilde{M}}$ agrees with $\tilde{N}$, so

$$
\begin{array}{rcll}
\sigma_{\tilde{M}} & = & \sigma_{\tilde{N}}\rho & \text{for some } \rho \qquad (11)\\
Q\sigma_{\tilde{M}} & \xrightarrow{(\tilde{x})\beta\sigma_{\tilde{M}}} & Q_1'\sigma_{\tilde{M}} & \\
\alpha\sigma_{\tilde{M}} & = & \beta\sigma_{\tilde{M}} & \qquad (12)\\
P_1'\sigma_{\tilde{M}} & \sim & Q_1'\sigma_{\tilde{M}} & \text{if } \alpha \text{ a communication action}\\
P_1'\sigma_{\tilde{M}}\rho & \sim & Q_1'\sigma_{\tilde{M}}\rho & \text{if } \alpha \text{ is a fusion action}
\end{array}
$$

where the substitution ρ agrees with $\alpha\sigma_{\tilde{M}}$, and using Lemma 17, by induction

$$
\begin{array}{rcll}
P_1'\sigma_{\tilde{M}} & = & Q_1'\sigma_{\tilde{M}} & \text{if } \alpha \text{ a communication action}\\
P_1'\sigma_{\tilde{M}}\rho & = & Q_1'\sigma_{\tilde{M}}\rho & \text{if } \alpha \text{ is a fusion action} \qquad (13)
\end{array}
$$

If $\tilde{M} \Leftrightarrow \tilde{N}$, we can now proceed just like the end of the proof of Theorem 16. We already know that $\tilde{M} \Rightarrow \tilde{N}$, and show that $\neg(\tilde{N} \Rightarrow \tilde{M})$ leads to a contradiction.

We know that $P \sim Q$; consider the transition $Q\sigma_{\tilde{N}} \xrightarrow{(\tilde{x})\beta\sigma_{\tilde{N}}} Q_1'\sigma_{\tilde{N}}$. P cannot simulate this transition using the term P_1, since by assumption $\neg(\tilde{N} \Rightarrow \tilde{M})$. Instead, let $P_2 \equiv \tilde{L}(\tilde{x})\gamma\,.\,P_2'$ be the term used.

We present the case where β is a fusion action; the case where it is not is simpler. Then

$$
\begin{array}{rcll}
Q\sigma_{\tilde{N}} & \xrightarrow{(\tilde{x})\beta\sigma_{\tilde{N}}} & Q_1'\sigma_{\tilde{N}} & \\
P\sigma_{\tilde{N}} & \xrightarrow{(\tilde{x})\gamma\sigma_{\tilde{N}}} & P_2'\sigma_{\tilde{N}} & \\
\beta\sigma_{\tilde{N}} & = & \gamma\sigma_{\tilde{N}} & \qquad (14)\\
Q_1'\sigma_{\tilde{N}}\rho' & \sim & P_2'\sigma_{\tilde{N}}\rho' & \qquad (15)
\end{array}
$$

where the substitution ρ' agrees with $\beta\sigma_{\tilde{N}}$, and $\tilde{N} \Rightarrow \tilde{L}$.

We now show that in this case, $P_1 + P_2 \sim P_2$, which is in contradiction to the fact that P is in mHNF. That means that indeed $\tilde{N} \Rightarrow \tilde{M}$, which is what we need to finish the proof of the Theorem.

To prove $P_1 + P_2 \sim P_2$, we look at the case where P_2 makes a transition which is to be matched by P_1:

$$\begin{array}{rcl} (P_1 + P_2)\sigma_{\tilde{M}} & \stackrel{(\tilde{x})\alpha\sigma_{\tilde{M}}}{\longrightarrow} & P_1'\sigma_{\tilde{M}} \\ P_2\sigma_{\tilde{M}} & \stackrel{(\tilde{x})\gamma\sigma_{\tilde{M}}}{\longrightarrow} & P_2'\sigma_{\tilde{M}} \end{array}$$

The transition of P_2 is possible since $\tilde{M} \Rightarrow \tilde{N} \Rightarrow \tilde{L}$. From (12), (11) and (14), we see that

$$\alpha\sigma_{\tilde{M}} = \gamma\sigma_{\tilde{M}} \tag{16}$$

and we need to show that

$$P_1'\sigma_{\tilde{M}}\rho \sim P_2'\sigma_{\tilde{M}}\rho \tag{17}$$

By the soundness of the axiom system we can use rule **DF1** to rewrite P_1 as $\tilde{M}(\tilde{x})\alpha\,.\,(P_1'\varrho)$ where ϱ agrees with α, and similarly P_2 as $\tilde{L}(\tilde{x})\gamma\,.\,(P_2'\varrho')$ and Q_1 as $\tilde{N}(\tilde{x})\beta\,.\,(Q_1'\varrho'')$, where ϱ' agrees with γ and ϱ'' with β. Then, using (12), (14) and (16), the trailing substitutions in (13), (15) and (17) are ineffectuous, and we know

$$\begin{array}{rcll} P_2'\sigma^{\gamma}\sigma_{\tilde{N}} & \sim & Q_1'\sigma^{\beta}\sigma_{\tilde{N}} & \text{by (15)} \\ P_2'\sigma^{\gamma}\sigma_{\tilde{N}}\rho & \sim & Q_1'\sigma^{\beta}\sigma_{\tilde{N}}\rho & \text{by (18) and Definition 8} \\ & \equiv & Q_1'\sigma^{\beta}\sigma_{\tilde{M}} & \text{by (11)} \\ & \sim & P_1'\varrho\sigma_{\tilde{M}} & \text{by (13)} \end{array} \tag{18}$$

which suffices for (17) to be true.

We have now proved that $\tilde{M} \Leftrightarrow \tilde{N}$, and can proceed just like the end of the proof of Theorem 16, from where we found the same fact from Lemma 13(2).

This conclusion the proof of Theorem 18. □

5 Encoding the λ-Calculus

Most process calculi with the ability of treating processes as first class objects (directly or indirectly) have attempted to embed or encode the λ-calculus, e.g. [Bou89, Tho90, Mil92, San93, San94, San95, Bou97].

The lazy λ-calculus [Abr89] has been modelled and analysed in the π-calculus by Milner [Mil92] and Sangiorgi [San93, San95]. The *strong* lazy reduction strategy, where reduction is possible under abstraction, has not been encoded in the pure π-calculus. The problem is that input binding in π-calculus can only be done by temporal guarding, $u(x)\,.\,P$, while the λ-calculus reduction rule allows the body of an abstraction to reduce *before* the abstracted variable is instantiated. In the fusion calculus, by decoupling the input action from the binding of the object, we can define a *delayed* input $(x)(u\,x \mid P)$ with the desired property.

Using other calculi with variants of delayed input, Sangiorgi [San96a] and Fu [Fu97] have presented encodings (into πI and χ-calculus, respectively). However they have not published any formal analysis of these encodings. In this section we present an encoding into the fusion calculus, and prove its accuracy. Our encoding is simpler than Sangiorgi's, and our calculus is more symmetric than that of Fu.

The strong lazy reduction strategy has the following reduction rules, where M and N range over λ terms:

$$(\lambda x\,.\,M)N \rightarrow_\lambda M\{N/x\} \qquad \frac{M \rightarrow_\lambda M'}{MN \rightarrow_\lambda M'N} \qquad \frac{M \rightarrow_\lambda M'}{\lambda x\,.\,M \rightarrow_\lambda \lambda x\,.\,M'}$$

To model persistent terms, we now add the usual replication operator of the π-calculus (first defined in [Mil92]) to our calculus. Essentially a replication $!\,P$ can be seen as "any number of copies of P in parallel". It has one transition rule: $!\,P \xrightarrow{\gamma} P'$ if $P \mid !\,P \xrightarrow{\gamma} P'$.

We only need to change Milner's encoding of the (non-strong) lazy reduction strategy [Mil92] minutely in the case of an abstraction, replacing an input prefix by a delayed input:

$$\begin{array}{rcl} [\![x]\!]u & = & \overline{x}\,u \\ [\![\lambda x\,.\,M]\!]u & = & (x\,v)(u\,xv \mid [\![M]\!]v) \\ [\![MN]\!]u & = & (v)([\![M]\!]v \mid (x)\overline{v}\,xu\,.\,[\![x := N]\!]) \end{array}$$

where $[\![x := N]\!]$ stands for $!\,(w)xw\,.\,[\![N]\!]w$.

We now show that the encoding is accurate, in the sense that there is a strong operational correspondence between reduction in the λ-calculus term and its encoding into the fusion calculus, and that convergence and divergence properties are preserved by the encoding. Below, we write $\tilde{N}$ for $N_1, \ldots, N_k$ and $[\![\tilde{x} := \tilde{N}]\!]$ for $[\![x_1 := N_1]\!] \mid \cdots \mid [\![x_n := N_n]\!], n \geq 0$. Let $\mathcal{L}$ stand for the family of λ-calculus terms, and $\mathcal{F}$ for the family of fusion calculus agents.

We define a correspondence between a λ term and its encoding by handling substitutions explicitly: a substitution $\{N/x\}$ on a λ term is represented by a fusion agent $[\![x := N]\!]$, which acts as an environment binding x to N.

Definition 14 *The relation $\mathcal{S} \subseteq \mathcal{L} \times \mathcal{F}$ contains all pairs (L, P) such that for some $k \geq 0$, $M, N_1, \ldots, N_k \in \mathcal{L}$, the following holds: $\mathrm{fv}(N_i) \cap \{x_1, \ldots, x_i\} = \emptyset$ for $1 \leq i \leq k$, $L \equiv M\{\tilde{N}/\tilde{x}\}$, and $P \equiv (\tilde{x})([\![M]\!]u \mid [\![\tilde{x} := \tilde{N}]\!])$ for any fresh u.*

So especially, for $k = 0$, $(M, [\![M]\!]u) \in \mathcal{S}$.

Theorem 19 *For $(L, P) \in \mathcal{S}$,*

1. *if $L \rightarrow_\lambda L'$, then $P \xrightarrow{\mathbf{1}}{}^{+} P'$ such that $(L', P') \in \mathcal{S}$.*
2. *if $P \xrightarrow{\mathbf{1}} P'$, then either $(L, P') \in \mathcal{S}$ or $L \rightarrow_\lambda L'$ such that $(L', P') \in \mathcal{S}$.*

PROOF: The proof goes by induction on the depth of the deduction of the reduction of L and transition of P, respectively, and induction on $k-i$.

1. By induction on the depth of the deduction of the reduction of L.

   Base case: $M \equiv (\lambda x_0.M_0)M_2$. Then $L \equiv M\{\tilde{N}/\tilde{x}\} \rightarrow_\lambda M_0\{M_2/x_0\}\{\tilde{N}/\tilde{x}\} \equiv L'$, and $P \equiv (\tilde{x})((v)((x_0\, w)(v\, zw \mid [\![M_0]\!]w) \mid (z)\overline{v}\, zu\,.\,[\![z := M_2]\!]) \mid [\![\tilde{x} := \tilde{N}]\!]) \xrightarrow{\mathbf{1}} (x_0)(\tilde{x})([\![M_0]\!]u \mid [\![x_0 := M_2]\!] \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv P'$, and $(L', P') \in \mathcal{S}$.

   Induction case:

   (a) $M \equiv \lambda x_0.M_0$. Then $L \equiv \lambda x_0.M_0\{\tilde{N}/\tilde{x}\} \rightarrow_\lambda \lambda x_0.M_0'\{\tilde{N}/\tilde{x}\}$ because $L_0 \equiv M_0\{\tilde{N}/\tilde{x}\} \rightarrow_\lambda M_0'\{\tilde{N}/\tilde{x}\} \equiv L_0'$.
   $P \equiv (\tilde{x})((x_0\, v)(u\, x_0 v \mid [\![M_0]\!]v) \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv (x_0\, v)(u\, x_0 v \mid (\tilde{x})([\![M_0]\!]v \mid [\![\tilde{x} := \tilde{N}]\!]))$. Define $P_0 \equiv (\tilde{x})([\![M_0]\!]v \mid [\![\tilde{x} := \tilde{N}]\!])$. Since the pair (L_0, P_0) is in $\mathcal{S}$ and $L_0 \rightarrow_\lambda L_0'$, by induction $P_0 \xrightarrow{\mathbf{1}}{}^{+} P_0'$ s.t. $(L_0', P_0') \in \mathcal{S}$ and $P_0' \equiv (\tilde{x})([\![M_0']\!]v \mid [\![\tilde{x} := \tilde{N}]\!])$. $P_0 \xrightarrow{\mathbf{1}}{}^{+} P_0'$ implies that $P \xrightarrow{\mathbf{1}}{}^{+} (x_0\, v)(u\, x_0 v \mid (\tilde{x})([\![M_0']\!]v \mid [\![\tilde{x} := \tilde{N}]\!])) \equiv (\tilde{x})([\![\lambda x_0.M_0']\!]u \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv P'$, and $(L', P') \in \mathcal{S}$ as desired.

   (b) $M \equiv M_1 M_2$ and $M_1\{\tilde{N}/\tilde{x}\} \rightarrow_\lambda M_1'\{\tilde{N}/\tilde{x}\}$. Then $L \rightarrow_\lambda (M_1' M_2)\{\tilde{N}/\tilde{x}\} \equiv L'$.
   $P \equiv (\tilde{x})((v)([\![M_1]\!]v \mid (z)\overline{v}\, zu\,.\,[\![z := M_2]\!]) \mid [\![\tilde{x} := \tilde{N}]\!])$. Since $M_1\{\tilde{N}/\tilde{x}\} \rightarrow_\lambda M_1'\{\tilde{N}/\tilde{x}\}$, by induction we know that $(\tilde{x})([\![M_1]\!]v \mid [\![\tilde{x} := \tilde{N}]\!]) \xrightarrow{\mathbf{1}}{}^{+} (\tilde{x})([\![M_1']\!]v \mid [\![\tilde{x} := \tilde{N}]\!])$ which is related in $\mathcal{S}$ to $M_1'\{\tilde{N}/\tilde{x}\}$. By the PAR and PASS rules, this means that $P \xrightarrow{\mathbf{1}}{}^{+} (\tilde{x})((v)([\![M_1']\!]v \mid (z)\overline{v}\, zu\,.\,[\![z := M_2]\!]) \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv (\tilde{x})([\![M_1' M_2]\!] \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv P'$, and again $(L', P') \in \mathcal{S}$.

   (c) $M \equiv x_i$ where $x_i \in \tilde{x}$, and $L \equiv x_i\{\tilde{N}/\tilde{x}\} \equiv N_i\{\tilde{N}/\tilde{x}\} \rightarrow_\lambda N_i'\{\tilde{N}/\tilde{x}\} \equiv L'$.
   $P \equiv (\tilde{x})(\overline{x_i}\, u \mid [\![\tilde{x} := \tilde{N}]\!]) \xrightarrow{\mathbf{1}} (\tilde{x})([\![N_i]\!]u \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv P''$. Since $(L, P'') \in \mathcal{S}$ and $\mathrm{fv}(N_i) \cap \{x_1, \ldots, x_i\} = \emptyset$, then by induction on $k-i$ or by previous cases, the lemma holds.

2. By induction on the depth of the deduction of the $\mathbf{1}$-transition of $P \equiv (\tilde{x})([\![M]\!]u \mid [\![\tilde{x} := \tilde{N}]\!])$.

   Base case: $M \equiv x_i$ and $x_i \in \tilde{x}$. Then $P \xrightarrow{\mathbf{1}} (\tilde{x})([\![N_i]\!]u \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv P'$, and $L \equiv x_i\{\tilde{N}/\tilde{x}\} \equiv N_i\{\tilde{N}/\tilde{x}\}$, and thus $(L, P') \in \mathcal{S}$.

   Induction case:

   (a) $M \equiv \lambda x_0.M_0$. Then $P \equiv (\tilde{x})((x_0\, w)(u\, x_0 w \mid [\![M_0]\!]w) \mid [\![\tilde{x} := \tilde{N}]\!])$
   $\equiv (x_0\, w)(u\, x_0 w \mid (\tilde{x})([\![M_0]\!] \mid [\![\tilde{x} := \tilde{N}]\!]))$, so by induction
   $P \xrightarrow{\mathbf{1}} (x_0\, w)(u\, x_0 w \mid (\tilde{x})([\![M_0']\!]w \mid [\![\tilde{x} := \tilde{N}]\!])) \equiv P'$, s.t.
   $M_0\{\tilde{N}/\tilde{x}\} \rightarrow_\lambda M_0'\{\tilde{N}/\tilde{x}\}$ or $M_0'\{\tilde{N}/\tilde{x}\} \equiv M_0\{\tilde{N}/\tilde{x}\}$. Now,
   $P' \equiv (\tilde{x})([\![\lambda x_0.M_0']\!]u \mid [\![\tilde{x} := \tilde{N}]\!])$, and either $L \rightarrow_\lambda (\lambda x_0.M_0')\{\tilde{N}/\tilde{x}\} \equiv L'$ s.t. $(L', P') \in \mathcal{S}$, or $(L, P') \in \mathcal{S}$.

(b) $M \equiv M_1 M_2$. There are two cases; either M_1 is an abstraction $\lambda x_0.M_0$ and $P \equiv (\tilde{x})((v)((x_0\, w)(v\, x_0 w \mid [\![M_0]\!]w) \mid (z)\overline{v}\, zu\,.\,[\![z := M_2]\!]) \mid [\![\tilde{x} := \tilde{N}]\!])$ reduces by communication over v, or P reduces in some other way.

i. $P \xrightarrow{1} (x_0)(\tilde{x})([\![M_0]\!]u \mid [\![x_0 := M_2]\!] \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv P'$, and $L \to_\lambda M_0\{M_2/x_0\}\{\tilde{N}/\tilde{x}\} \equiv L'$, and $(L', P') \in \mathcal{S}$.

ii. $P \equiv (\tilde{x})((v)([\![M_1]\!]v \mid (z)\overline{v}\, zu\,.\,[\![z := M_2]\!]) \mid [\![\tilde{x} := \tilde{N}]\!]) \xrightarrow{1}$ $(\tilde{x})((v)([\![M_1']\!]v \mid (z)\overline{v}\, zu\,.\,[\![z := M_2]\!]) \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv (\tilde{x})([\![M_1' M_2]\!]u \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv P'$ like previously. $L \to_\lambda (M_1' M_2)\{\tilde{N}/\tilde{x}\} \equiv L'$ and again, $(L', P') \in \mathcal{S}$.

□

Lemma 20 *For $(L, P) \in \mathcal{S}$, if $L \not\to_\lambda$, then $P \xrightarrow{1}{}^{*} P'$ such that $P' \not\xrightarrow{1}$ and $(L, P') \in \mathcal{S}$.*
Proof: By structural induction on M, where $L \equiv M\{\tilde{N}/\tilde{x}\}$.

The base case where $M \equiv z$ and $z \notin \tilde{x}$ is trivial.

Induction:

1. $M \equiv \lambda x_0.M_0$ s.t. $M_0\{\tilde{N}/\tilde{x}\} \not\to_\lambda$. Then $P \equiv (z\, w)(u\, zw \mid (\tilde{x})([\![M_0]\!]w \mid [\![\tilde{x} := \tilde{N}]\!])$ and since $P_0 \equiv (\tilde{x})([\![M_0]\!]w \mid [\![\tilde{x} := \tilde{N}]\!])$ is related in $\mathcal{S}$ to $L_0 \equiv M_0\{\tilde{N}/\tilde{x}\}$, and by induction $P_0 \xrightarrow{1}{}^{*} P_0'$ s.t. $P_0' \not\xrightarrow{1}$ and $(L_0, P_0') \in \mathcal{S}$. This implies that $P \xrightarrow{1}{}^{+} P'$ s.t. $P' \not\xrightarrow{1}$ and $(L', P') \in \mathcal{S}$.
2. $M \equiv M_1 M_2$. Holds by similar reasoning.
3. $M \equiv x_i$ where $x_i \in \tilde{x}$. Then $P \xrightarrow{1} (\tilde{x})([\![N_i]\!]u \mid [\![\tilde{x} := \tilde{N}]\!]) \equiv P'$, and since $(L, P') \in \mathcal{S}$ and $\mathrm{fn}(N_i) \cap \{x_1, \ldots, x_i\} = \emptyset$, by induction on $k - i$ or previous cases the lemma holds.

□

The preservation of convergence (and thereby divergence) properties follows from the preceding Lemma and Theorem.

Theorem 21 *For $(L, P) \in \mathcal{S}$, if $L \downarrow L'$ then $P \downarrow P'$ with $(L', P') \in \mathcal{S}$, and if $P \downarrow P'$ then $L \downarrow L'$ with $(L', P') \in \mathcal{S}$.*
Proof: Follows by induction on the number of reductions/transitions before convergence, using of Theorem 19, and from Lemma 20. □

References

[Abr89] S. Abramsky. The lazy lambda calculus. In D. Turner, editor, *Research Topics in Functional Programming*. Addison-Wesley, 1989.

[Bou89] G. Boudol. Towards a lambda-calculus for concurrent and communicating systems. In J. Díaz and F. Orejas, editors, *Proceedings of TAPSOFT '89, Volume 1*, volume 351 of *LNCS*, pages 149–161. Springer, 1989.

[Bou97] G. Boudol. The π-calculus in direct style. In *Proceedings of POPL '97*, pages 228–241. ACM, Jan. 1997.

[Fu97] Y. Fu. A proof-theoretical approach to communication. In P. Degano, R. Gorrieri and A. Marchetti-Spaccamela, editors, *Proceedings of ICALP '97*, volume 1256 of *LNCS*, pages 325–335. Springer, 1997.

[Mil91] R. Milner. The polyadic π-calculus: a tutorial. Technical Report ECS-LFCS-91-180, LFCS, University of Edinburgh, Oct. 1991. Also in *Logic and Algebra of Specification*, ed. F. L. Bauer, W. Brauer and H. Schwichtenberg, Springer, 1993.

[Mil92] R. Milner. Functions as processes. *Journal of Mathematical Structures in Computer Science*, 2(2):119–141, 1992.

[Mil96] R. Milner. Calculi for interaction. *Acta Informatica*, 33(8):707–737, Nov. 1996.

[MPW92] R. Milner, J. Parrow and D. Walker. A calculus of mobile processes, Parts I and II. *Journal of Information and Computation*, 100:1–77, Sept. 1992.

[PS95] J. Parrow and D. Sangiorgi. Algebraic theories for name-passing calculi. *Journal of Information and Computation*, 120(2):174–197, 1995.

[PV97] J. Parrow and B. Victor. The update calculus. In *Proceedings of AMAST'97*, volume 1349 of *LNCS*. Springer, 1997. Full version available as Technical report DoCS 97/93, Uppsala University.

[San93] D. Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms*. PhD thesis, LFCS, University of Edinburgh, 1993.

[San94] D. Sangiorgi. The lazy lambda calculus in a concurrency scenario. *Journal of Information and Computation*, 111(1):120–131, 1994. An extended abstract appeared in Proc. of LICS'92.

[San95] D. Sangiorgi. Lazy functions and mobile processes. Rapport de Recherche RR-2515, INRIA Sophia-Antipolis, 1995.

[San96a] D. Sangiorgi. π-calculus, internal mobility and agent-passing calculi. *Theoretical Computer Science*, 167(1,2):235–274, 1996. Also as Rapport de Recherche RR-2539, INRIA Sophia-Antipolis, 1995. Extracts of parts of the material contained in this paper can be found in the Proceedings of TAPSOFT'95 and ICALP'95.

[San96b] D. Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33:69–97, 1996. Earlier version published as Report ECS-LFCS-93-270, University of Edinburgh. An extended abstract appeared in the *Proceedings of CONCUR '93*, LNCS 715.

[Smo94] G. Smolka. A foundation for higher-order concurrent constraint programming. In J.-P. Jouannaud, editor, *Proceedings of Constraints in Computational Logics*, volume 845 of *LNCS*, pages 50–72. Springer, Sept. 1994. Available as Research Report RR-94-16 from DFKI Kaiserslautern.

[Tho90] B. Thomsen. *Calculi for Higher Order Communicating Systems*. PhD thesis, Imperial College, University of London, Sept. 1990.

[VM94] B. Victor and F. Moller. The Mobility Workbench — a tool for the π-calculus. In D. Dill, editor, *Proceedings of CAV '94*, volume 818 of *LNCS*, pages 428–440. Springer, 1994.

[VP96] B. Victor and J. Parrow. Constraints as processes. In U. Montanari and V. Sassone, editors, *Proceedings of CONCUR '96*, volume 1119 of *LNCS*, pages 389–405. Springer, 1996.