Symmetric action calculi

Philippa Gardner and Lucian J. Wischik *

February 1999

Abstract

Many calculi for describing interactive behaviour involve *names*, *name-abstraction* and *name-restriction*. Milner's reflexive action calculi provide a framework for exploring such calculi. It is based on names and name-abstraction. We introduce an alternative framework, the *symmetric action calculi*, based on names, *co-names* and name-restriction (or *hiding*). Name-abstraction is intepreted as a derived operator. The symmetric framework conservatively extends the reflexive framework. It allows for a natural intepretation of a variety of calculi: we give interpretations for the π-calculus, the π_I-calculus and a variant of the fusion calculus. We then give a combinatory version of the symmetric framework, in which name-restriction also is expressed as a derived operator. This combinatory account provides an intermediate step between our non-standard use of names in graphs, and the more standard graphical structure arising from category theory. To conclude, we briefly illustrate the connection between our work and Yoshida's process graphs.

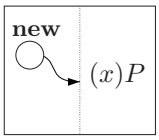
1 Introduction

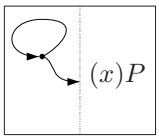
There are many calculi for describing interactive behaviour based on *names*, *name-abstraction* and *name-restriction*. Examples of such calculi include the π-calculus [MPW92], the fusion calculus [VP98] (which aims to be more foundational than the π-calculus and which does not use name-abstraction), calculi for describing features in distributed systems such as the ambient calculus [CG98] and the distributed π-calculus [Sew97], the spi-calculus for analysing security protocols [AG97] and the ν-calculus [PS93]. The naming constructions are also present in conventional imperative programming languages. Milner introduced *action calculi* [Mil96] as a graphical framework within which to explore a foundational understanding of such calculi. We introduce the *symmetric action calculi*, which conservatively extend the reflexive action calculi [Mil94] and which have different naming primitives.

The action calculus framework treats name $\langle x \rangle$ and *name-abstraction* $(x)P$ as primitive. Name-abstraction declares x as a local name which expects an input. The way this calculus expresses the restriction $(\nu x)P$ is by supplying

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

a dummy input. The following graphs illustrate the two techniques used to achieve this:

(a) 

(b) 

The first graph interprets the restriction $(\nu x)P$ by declaring a constant node **new**; it corresponds to the action term $\mathbf{new}\cdot(x)P'$. The second graph interprets restriction with a reflexive loop, and corresponds to the action term $\uparrow((x)(\langle x\rangle \otimes P'))$.

The symmetric action calculi in contrast are based on the notion of name x^+, *co-name* x^- and *name-hiding* $(\_)_x$. The name-hiding operator just declares that x is a local name; this fits in well with the intuitive notion of restriction. Graphically we represent $(t)_x$ by simply crossing out (or hiding) the name x. Name-abstraction is interpreted by $(x^- \otimes P)_x$, where the co-name x^- provides the input port previously incorporated within the abstraction operator.

We start in section 2 with a brief review of Milner's existing reflexive action calculi. Section 3 introduces the symmetric action calculi, and shows by way of example how the π- and π_I-calculi and a variant of the fusion calculus can be naturally expressed within the symmetric framework. The symmetric framework conservatively extends the reflexive framework. Our proof of this uses weak normal forms, known as *molecular forms*. Section 4 introduces symmetric molecular forms, briefly reviews reflexive molecular forms, and outlines of the proof of conservativity. Finally in Section 5 we give a combinatoric presentation of the symmetric action calculi. In this presentation name-hiding is a derived operator. (Analogously, λ-abstraction is a derived operator in combinatory logic.) Symmetric combinators provide an intermediate step between the non-standard use of names in symmetric action calculi graphs, and the more standard graphical structures arising from category theory. A full understanding of the categorical models, however, requires further investigation. The concluding section briefly explores the connection between symmetric action calculi and Yoshida's process graphs [Yos95] (which are a non-deterministic extension of Lafont's interaction nets [Laf97]).

2 Reflexive Action Calculi

In this section we summarise the reflexive action calculi and briefly explain the corresponding graphs. By way of example we give an action calculus corresponding to the synchronous π-calculus [MPW92]. A more detailed account of action calculi may be found in the introductory paper [Mil96], and of reflexive action calculi in [Mil94]. The categorical models and corresponding type-theory presentation is given in [GH97], and a tutorial paper [Gar99a] links the action calculi with other graphical grameworks studied in the literature.

The plan is as follows. We construct a set of terms (Definition 2.1) and quotient the terms by an equational theory (Definition 2.2). We call the equivalence classes so generated *actions*; the set of all actions is called the *action calculus*.

The intention is that an action correspond to a process, and that a *reaction relation* on actions correspond to reactions between processes.

An *action calculus* is specified by a *signature* $\mathbb{K} = (\mathsf{P}, \mathcal{K})$, which consists of a set P of basic types $p, q, \ldots$ called *primes*, and a set $\mathcal{K}$ of constants $K, L, \ldots$ called *controls*. Each control has an associated arity $((m_1, n_1), \ldots, (m_r, n_r)) \to (m, n)$. Here the ms and ns are finite sequences of primes. These sequences are called *tensor arities*; we write ε for the empty sequence and $\otimes$ for concatenation. We assume a fixed denumerable set X of *names* $x, y, \ldots$, each of which has a prime arity, and we write $x\!:\!p$ to indicate that x has the prime arity p.

Definition 2.1 [Terms] The set of *reflexive terms* over signature $\mathbb{K}$, denoted by $\mathsf{RT}(\mathbb{K})$, is the algebra freely generated by the basic operators: identity $\mathbf{id}_m$, composition $\cdot$, tensor $\otimes$, permutation $\mathbf{p}_{m,n}$, reflexion $\uparrow_k$, datum $\langle x \rangle$, abstraction $(x)\_$ and the controls K. A term t is assigned an arity $t : m \to n$, for input arity m and output arity n, using the following rules:

$$\mathbf{id}_m : m \to m \qquad \frac{s : k \to l \quad t : l \to m}{s \cdot t : k \to m} \qquad \frac{s : k \to m \quad t : l \to n}{s \otimes t : k \otimes l \to m \otimes n}$$

$$\mathbf{p}_{m,n} : m \otimes n \to n \otimes m \qquad \frac{t : k \otimes m \to k \otimes n}{\uparrow_k t : m \to n}$$

$$\langle x \rangle : \epsilon \to p, \quad x\!:\!p \qquad \frac{t : m \to n}{(x)t : p \otimes m \to n} \quad x\!:\!p$$

$$\frac{t_1 : m_1 \to n_1 \quad \ldots \quad t_r : m_r \to n_r}{K(t_1, \ldots, t_r) : k \to l} \quad K : ((m_1, n_1), \ldots, (m_r, n_r)) \to (k, l) \text{ in } \mathbb{K}$$

If a term contains no controls we call it a *wiring term*. The notions of *free* and *bound name* are standard: $(x)t$ binds x in t, and $\langle x \rangle$ represents a free occurrence of x. We write $t\{y/x\}$ to denote capture-avoiding substitution. (We assume the ability to generate fresh names, so that substitution does not involve α-conversion. α-conversion is a consequence of the equational theory in Definition 2.2.) The set of names free in $s, t, \ldots$ is denoted by $fn(s, t, \ldots)$. Given a possibly empty sequence of names $\vec{x} = x_1, \ldots, x_r$ with $x_1 : p_1, \ldots x_r : p_r$, we write $|\vec{x}|$ for $p_1 \otimes \ldots \otimes p_r$. Composition $\cdot$ binds more tightly than tensor $\otimes$. In this paper we implicitly assume that terms have the correct arities; all equations are between terms of the same arity.

It is standard category-theory practice to describe graphs using algebraic theories. Action-calculi graphs differ from standard graphs in that they incorporate names and name-abstraction within the graph. Figure 1 illustrates the sort of graphs described by terms in the reflexive action calculus; the equational theory on terms given in Definition 2.2 intuitively corresponds to graph equality. The graphs were introduced in [Jen99], and [Gar99a] provides an overview.

The equational theory upon reflexive action terms is generated by the axioms of a strict symmetric monoidal category, the axioms for reflexion and two additional axioms for naming.

Definition 2.2 [The theory RAC] The *equational theory* RAC is the set of equations upon reflexive action terms generated by the following axioms:

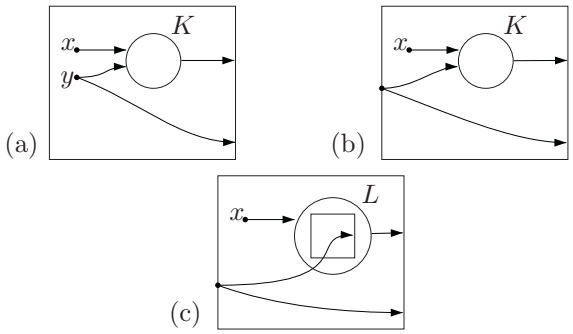


Figure 1: Graph (a) represents the term $(\langle x\rangle \otimes \langle y\rangle) \cdot K \otimes \langle y\rangle$ which has arity $0 \to 2$. Graph (b) represents $(y)((\langle x\rangle \otimes \langle y\rangle) \cdot K \otimes \langle y\rangle) : 1 \to 2$, in which the wire formerly marked y becomes part of the left interface. Graph (c) represents $(y)(\langle x\rangle \cdot L(\langle y\rangle) \otimes \langle y\rangle)$ and illustrates a wire crossing a control boundary through the use of the abstracted name y. In all examples $K : () \to (2,1)$ and $L : ((0,1)) \to (1,1)$.

1. the axioms of a strict symmetric monoidal category

$$
\begin{array}{ll}
A1. & s \cdot \mathbf{id}_n = s = \mathbf{id}_m \cdot s \\
A2. & s \otimes \mathbf{id}_\varepsilon = s = \mathbf{id}_\varepsilon \otimes s \\
A3. & \mathbf{id}_m \otimes \mathbf{id}_n = \mathbf{id}_{m \otimes n} \\
A4. & s \cdot (t \cdot u) = (s \cdot t) \cdot u \\
A5. & s \otimes (t \otimes u) = (s \otimes t) \otimes u \\
A6. & (s \cdot t) \otimes (u \cdot v) = (s \otimes u) \cdot (t \otimes v)
\end{array}
$$

$$
\begin{array}{lll}
P1. & \mathbf{p}_{k,m} \cdot (s \otimes t) = (t \otimes s) \cdot \mathbf{p}_{l,n}, & s : m \to n, \quad t : k \to l \\
P2. & \mathbf{p}_{m,n} \cdot \mathbf{p}_{n,m} = \mathbf{id}_{m \otimes n} & \\
P3. & \mathbf{p}_{m \otimes n,k} = (\mathbf{id}_m \otimes \mathbf{p}_{n,k}) \cdot (\mathbf{p}_{m,k} \otimes \mathbf{id}_n) &
\end{array}
$$

2. the axioms for reflexion

$$
\begin{array}{ll}
R1. & \mathbf{id}_\epsilon = \uparrow_k \mathbf{id}_k \\
R2. & \mathbf{id}_k = \uparrow_k \mathbf{p}_{k,k} \\
R3. & (\uparrow_k t) \otimes \mathbf{id}_m = \uparrow_k (t \otimes \mathbf{id}_m) \\
R4. & (\uparrow_k s) \cdot t = \uparrow_k (s \cdot (\mathbf{id}_k \otimes t)) \\
R5. & s \cdot (\uparrow_k t) = \uparrow_k ((\mathbf{id}_k \otimes s) \cdot t) \\
R6. & \uparrow_l (\uparrow_k t) = \uparrow_k \uparrow_l ((\mathbf{p}_{l,k} \otimes \mathbf{id}_m) \cdot t \cdot (\mathbf{p}_{k,l} \otimes \mathbf{id}_n))
\end{array}
$$

3. the naming axioms, illustrated in Figure 2

$$
\begin{array}{lll}
\sigma. & (\langle y\rangle \otimes \mathbf{id}_m) \cdot (x)t = t\{y/x\}, & x{:}p, \quad y : p \\
\delta. & (x)((\langle x\rangle \otimes \mathbf{id}) \cdot t) = t, & x \notin fn(t)
\end{array}
$$

Remark 2.3 Without $R1$, the reflexion axioms correspond to a category with a trace operator [JSV96]. In fact the axiom $R1$ does not hold for arbitrary traced monoidal categories. It was first suggested as an action calculi axiom by Jensen in [Mil94] and is included here because it is necessary for the conservativity

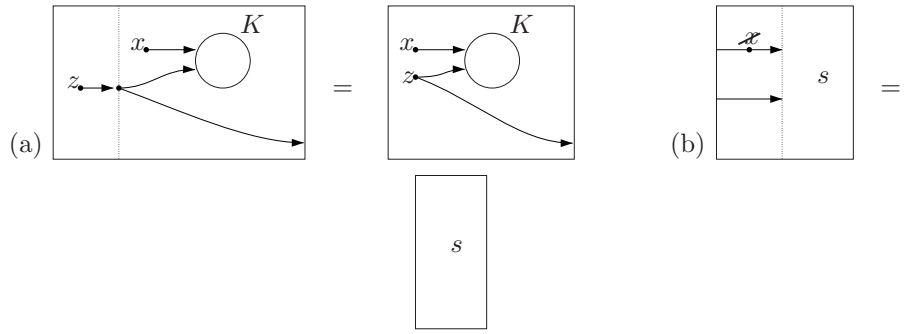


Figure 2: Graph (a) illustrates an instance of the σ-axiom. Graph (b) represents the δ-axiom. The equations for the two graphs are: (a) $(\langle z\rangle \otimes \mathbf{id}_\epsilon)\cdot(y)((\langle x\rangle \otimes \langle y\rangle)\cdot K \otimes \langle y\rangle) = (\langle x\rangle \otimes \langle z\rangle)\cdot K \otimes \langle z\rangle$ and (b) $(x)((\langle x\rangle \otimes \mathbf{id})\cdot t) = t$.

result in Section 4. (Milner's original presentation [Mil96] has different primitive operators and axioms. Our presentation is equivalent, and relates more simply to category theory.)

Quotienting terms $\mathsf{RT}(\mathbb{K})$ by the theory RAC gives rise to equivalence classes. We call these equivalence classes *reflexive actions*. The set of these actions is called the *reflexive action calculus* specified by $\mathbb{K}$.

The purpose of the *reaction relation* $\searrow$ on actions is to describe their interactive behaviour. It relates actions of the same arity and is preserved under tensor, composition, reflexion and abstraction. In addition, we intend that the interactive behaviour of the calculus be expressed through the reaction of its controls. Accordingly we impose the minimality condition that there is no s with $\mathbf{id} \searrow s$, which implies that wiring actions do not react. It may be the case that further conditions should be imposed, perhaps in relation to the connectedness and locality of graphs.

Observational properties of the reaction relation are currently under investigation. Leifer and Milner have a contextual bisimulation result for non-nested action calculi: that is, for action calculi specified by controls with arity $() \to (m, n)$. The nesting case appears significantly more difficult.

Definition 2.4 [Derived Operators] Let $\vec{x}$ denote the list of names $x_1, \ldots, x_r$ and $|\vec{x}| = m$. The following derived operators are used throughout the paper:

$$\begin{aligned}
(\vec{x})t &\stackrel{\text{def}}{=} (x_1)\ldots(x_r)t, \qquad x_i \text{ distinct} \\
\langle\vec{x}\rangle &\stackrel{\text{def}}{=} \langle x_1\rangle \otimes \ldots \otimes \langle x_r\rangle \\
\mathbf{copy}_m &\stackrel{\text{def}}{=} (\vec{x})\langle\vec{x},\vec{x}\rangle \\
\mathbf{disc}_m &\stackrel{\text{def}}{=} (\vec{x})\mathbf{id}_\epsilon
\end{aligned}$$

Let $(\,)t$ denote the term t and $\langle\,\rangle$ denote the term $\mathbf{id}_\varepsilon$. (Notice that $\mathbf{copy}_m$ and $\mathbf{disc}_m$ are defined using particular names. With α-conversion, which can be deduced from the σ- and δ-axioms, we are justified in choosing these names at will.)

Remark 2.5 It is possible to reformulate the action calculus by writing $(\vec{x},\vec{y})\langle\vec{y},\vec{x}\rangle$ for $\mathbf{p}_{m,n}$ and $(\vec{x})\langle\vec{x}\rangle$ for $\mathbf{id}_m$, with $|\vec{x}| = m$ and $|\vec{y}| = m$. The term $\mathbf{id}_\epsilon$ is however still necessary. This reformulation is closer in syntax to the π-calculus, while our actual choice of presentation is more natural from the category-theory perspective.

Example 2.6 We present the synchronous π-calculus [MPW92] and give the corresponding action calculus. The set of π-processes is defined by the abstract grammar

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

and is subject to the standard rules for structural congruence. It has binding input processes $x(\vec{z}).P$ and non-binding output processes $\overline{x}\langle\vec{z}\rangle.P$. The reaction relation is generated by the rule

$$\overline{x}\langle\vec{y}\rangle.Q|x(\vec{z}).P \searrow Q|P\{\vec{y}/\vec{z}\}$$

and is closed under composition, restriction and the structural congruence.

The corresponding action calculus PIC is specified by the signature $\mathbb{K} = (\{1\},\{\mathbf{in},\mathbf{out}\})$ with the arity rules

$$\frac{t : \epsilon \to m}{\mathbf{out}(t) : 1 \to \epsilon} \qquad \frac{t : m \to \epsilon}{\mathbf{in}(t) : 1 \to \epsilon}$$

and the reaction relation

$$\langle x\rangle \cdot \mathbf{out}(s) \ \otimes \ \langle x\rangle \cdot \mathbf{in}(t) \ \searrow \ s \cdot t.$$

Reaction is not permitted inside **out** or **in**. The reaction relation is illustrated in figure 3

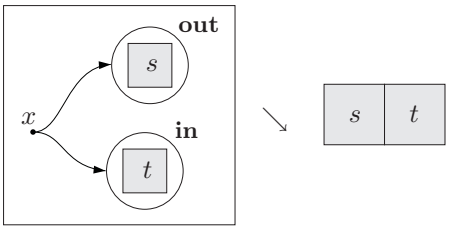


Figure 3: The reaction relation for PIC. The contents of the controls **out** and **in** correspond to concretions and abstractions in the π-calculus. (We shade s and t to indicate that they are being used as parameters).

The translation $(\_)'$, which maps π-processes to action terms of arity $\epsilon \to \epsilon$, is defined inductively by

$$\begin{array}{rcl}
(\mathbf{nil})' & = & \mathbf{id}_\epsilon \\
(P|Q)' & = & P' \otimes Q' \\
(\nu.(z)P)' & = & \uparrow((z)(\langle z\rangle \otimes P')) \\
(\overline{x}\langle\vec{y}\rangle.Q)' & = & \langle x\rangle \cdot \mathbf{out}(\langle\vec{y}\rangle \otimes Q') \\
(x(\vec{z}).P)' & = & \langle x\rangle \cdot \mathbf{in}((\vec{z})P')
\end{array}$$

Remark 2.7 In [Mil96], Milner gives an example action calculus corresponding to the asynchronous π-calculus [HT92, Bou92]. He chooses to interpret the π-process $\overline{x}\langle\vec{y}\rangle$ by the action calculus term $\langle x, \vec{y}\rangle \cdot \mathbf{out}$. In our presentation we instead incorporate the $\vec{y}$ s within the body of the **out** control, interpreting the π-processes $\overline{x}\langle\vec{y}\rangle.Q$ by the action calculus term $\langle x\rangle \cdot \mathbf{out}(\langle\vec{y}\rangle \otimes Q')$. In our presentation the body of the control corresponds to *concretion*, presented in [Mil99]. Since Milner is working in the non-reflexive case, he also declares an extra control $\mathbf{new} : \epsilon \to 1$ and translates $(\nu x)P$ by $\mathbf{new} \cdot (x)P'$.

Notice the asymmetry in the operators of action calculi: datum $\langle x\rangle : \epsilon \to p$ is a primitive naming operator, but discard $\mathbf{disc} : p \to \epsilon$ is a derived anonymous operator. The following section explores a symmetric version.

3 Symmetric Action Calculi

In this section we introduce the *symmetric action calculi*. These incorporate two datums: normal datum $x^+ : \epsilon \to p$ and a *co-datum* $x^- : p \to \epsilon$. With this choice, we are able to introduce a *name-hiding* operator $(t)_x$ which does not affect the interface of the graph (as is the case with name-abstraction). We show that several variants of the π-calculus can be naturally expressed as symmetric action calculi. In particular, restriction in these π-calculi correspond directly to name-hiding in the symmetric action calculi.

The plan is the same as for the reflexive action calculi. We construct terms and quotient them by an equivalence relation. We call the equivalence classes so generated *symmetric actions*, and we call the set of symmetric actions the *symmetric action calculus*.

Definition 3.1 [Terms] The set of *symmetric terms* over a signature $\mathbb{K}$, denoted by $\mathsf{ST}(\mathbb{K})$, is constructed from the basic operators: identity $\mathbf{id}_m$, composition $\cdot$, tensor $\otimes$, permutation $\mathbf{p}_{m,n}$, datum x^+, co-datum x^-, name-hiding $(\_)_x$ and the controls K. A term t is assigned an arity $t : m \to n$ using the following rules:

$$\mathbf{id}_m : m \to m \qquad \frac{s : k \to l \qquad t : l \to m}{s \cdot t : k \to m} \qquad \frac{s : k \to m \qquad t : l \to n}{s \otimes t : \ k \otimes l \ \to \ m \otimes n}$$

$$\mathbf{p}_{m,n} : m \otimes n \to n \otimes m$$

$$x^+ : \epsilon \to p, \quad x{:}p \qquad x^- : p \to \epsilon, \quad x{:}p \qquad \frac{t : m \to n}{(t)_x : m \to n}$$

$$\frac{s_1 : m_1 \to n_1 \quad \cdots \quad s_r : m_r \to n_r}{K(s_1, \ldots, s_r) : k \to l} \quad K : ((m_1, n_1), \ldots, (m_r, n_r)) \to (k, l) \text{ in } \mathbb{K}.$$

Figure 4 illustrates the graphical notation for these operators.

The notion of *free and local names* is standard: x is local inside $(\_)_x$, and both x^+ and x^- represent free occurences of x. We write $t\{y/x\}$ for the standard capture-avoiding substitution. All equations are between terms of the same arity.

The equational theory upon symmetric terms is generated by the axioms of a strict symmetric monoidal category, and some additional naming axioms.

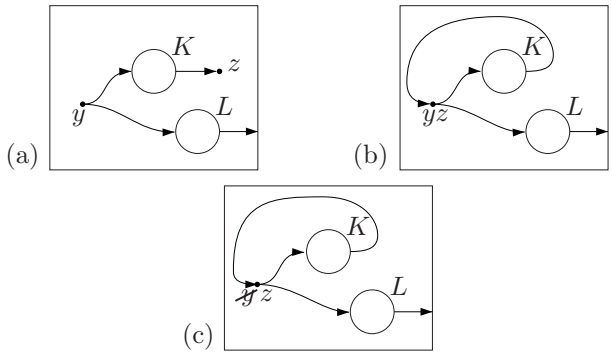


Figure 4: (a) Nodes marked • are called *naming nodes*. (b) In general naming nodes are labelled by *sets* of names indicating that the names are joined, or equal: algebraically this is written $y^+ \cdot z^-$. (c) We draw a line through a name when it has been hidden by the name-hiding operator $(\_)_y$. Notice that the name-hiding does not change the interface of the graph. The symmetric terms for these graphs are: (a) $y^+ \cdot K \cdot z^- \otimes y^+ \cdot L$; (b) $y^+ \cdot z^- \otimes y^+ \cdot K \cdot z^- \otimes y^+ \cdot L$; (c) $(y^+ \cdot z^- \otimes y^+ \cdot K \cdot z^- \otimes y^+ \cdot L)_y$.

Definition 3.2 [The theory SAC] The *equational theory* SAC is the set of equations upon symmetric terms generated by the following axioms:

1. the axioms of a strict symmetric monoidal category, given in Definition 2.2,
2. the naming axioms, where $x \notin fn(s)$ in $N1$-$N6$ (σ and δ are illustrated in figure 5)

$$\begin{array}{ll} \boldsymbol{\sigma}. & (y^+ \cdot x^- \otimes t)_x = (x^+ \cdot y^- \otimes t)_x = (t\{y/x\})_x \\ \boldsymbol{\delta}. & (x^+ \otimes x^-)_x = \mathbf{id}_p, \qquad x : p \end{array}$$

$$\begin{array}{ll} N1. & ((t)_x)_y = ((t)_y)_x \\ N2. & (s)_x = s \\ N3. & (t \cdot s)_x = (t)_x \cdot s \\ N4. & (s \cdot t)_x = s \cdot (t)_x \\ N5. & (t \otimes s)_x = (t)_x \otimes s \\ N6. & (s \otimes t)_x = s \otimes (t)_x \end{array}$$

We intuitively think of $y^+ \cdot x^-$ as joining the nodes y and x in the graph. The σ-axiom states that if x is local then y can be substituted for x. Lemma 3.3 part 3 shows that this property is also true in the non-local setting, provided that we retain the information that x and y are joined. Thus, although the algebraic structure gives an inherent directionality to the graphs, the connections between names are non-directional.

Lemma 3.3

1. [α-conversion] $(t)_x = (t\{y/x\})_y, \qquad y \notin fn(t)$
2. $x^+ \cdot x^- = \mathbf{id}_\epsilon$
3. [free substitution] $y^+ \cdot x^- \otimes t = y^+ \cdot x^- \otimes t\{y/x\} = y^+ \cdot x^- \otimes t\{x/y\}$.

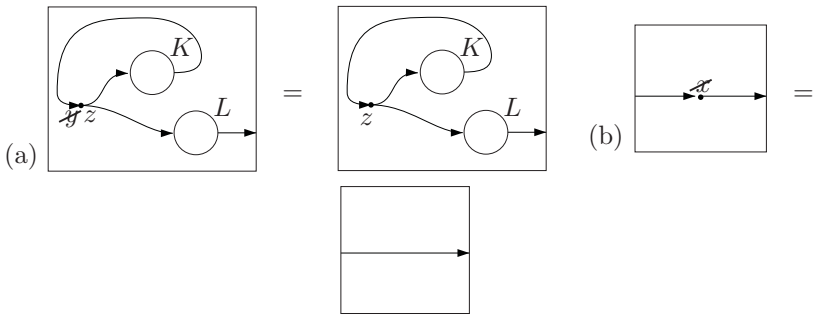


Figure 5: Graph (a) shows an application of the σ-axiom corresponding to the symmetric term equation $(y^+ \cdot z^- \otimes y^+ \cdot K \cdot z^- \otimes y^+ \cdot L)_y = (z^+ \cdot K \cdot z^- \otimes z^+ \cdot L)$. Graph (b) shows the δ-axiom.

The *symmetric actions* are the equivalence classes obtained by quotienting terms $\mathsf{ST}(\mathbb{K})$ by the theory SAC. The set of these actions is called the *symmetric action calculus* specified by $\mathbb{K}$.

Definition 3.4 [Derived Operators] Let $\vec{x}$ denote the list of names $x_1, \ldots, x_r$ with $|\vec{x}| = m$. Define

$$
\begin{array}{rcl}
\vec{x}^+ & \stackrel{\text{def}}{=} & x_1{}^+ \otimes \ldots \otimes x_r{}^+ \\
\vec{x}^- & \stackrel{\text{def}}{=} & x_1{}^- \otimes \ldots \otimes x_r{}^- \\
(t)_{\vec{x}} & \stackrel{\text{def}}{=} & (\ldots((t)_{x_1})_{x_2}\ldots)_{x_r} \\
\mathbf{new}_m & \stackrel{\text{def}}{=} & (\vec{x}^+)_{\vec{x}} \\
\mathbf{disc}_m & \stackrel{\text{def}}{=} & (\vec{x}^-)_{\vec{x}} \\
(\vec{x})t & \stackrel{\text{def}}{=} & (\vec{x}^- \otimes t)_{\vec{x}} \\
\uparrow_m t & \stackrel{\text{def}}{=} & (\ (\vec{x}^+ \otimes \mathbf{id}) \cdot t \cdot (\vec{x}^- \otimes \mathbf{id})\ )_{\vec{x}} \\
\mathbf{copy}_m & \stackrel{\text{def}}{=} & (\vec{x}^- \otimes \vec{x}^+ \otimes \vec{x}^+)_{\vec{x}} \\
\mathbf{merge}_m & \stackrel{\text{def}}{=} & (\vec{x}^- \otimes \vec{x}^- \otimes \vec{x}^+)_{\vec{x}} \\
\rceil_m & \stackrel{\text{def}}{=} & \mathbf{merge}_m \cdot \mathbf{disc}_m \\
\lfloor_m & \stackrel{\text{def}}{=} & \mathbf{new}_m \cdot \mathbf{copy}_m
\end{array}
$$

Remark 3.5 Using the derived operator for reflexion, it is not difficult to prove the equalities corresponding to the reflexive axioms $R1$ to $R6$ in Definition 2.2. We are also able to prove analogous equalities to the axioms $B1$-$B8$ given in Definition 5.2. Also, notice that a simple wiring context can turn an input wire into an output wire and vice versa: given $s : m \to p \otimes n$ we have $(\mathbf{id}_p \otimes s) \cdot (\lfloor_p \otimes \mathbf{id}_m) : p \otimes m \to n$. This demonstrates that the difference between the input and output arities is present for algebraic convenience rather than necessity.

Example 3.6 We now give symmetric action calculi into which can be embedded three variants of the π-calculus. The first variant is the synchronous π-calculus mentioned in the previous section (however we now embed it into a symmetric action calculus rather than a reflexive one). The second is the π_I-calculus of Sangiorgi [San96], and the third is a variant of the fusion calculus of Victor and Parrow [VP98] which we call the π_F-calculus.

The three calculi differ in the binding of their input and output processes: the π-calculus binds on the input process, the π_I-calculus binds on both input and output processes, and the π_F-calculus binds on neither. Processes in the three algebras are generated by the abstract grammars

$$\begin{array}{lrcl} (\pi) & P & ::= & \mathbf{nil} \mid P|P \mid (\nu x)P \mid \overline{x}\langle\vec{y}\rangle.P \mid x(\vec{z}).P \\ (\pi_I) & P & ::= & \mathbf{nil} \mid P|P \mid (\nu x)P \mid \overline{x}(\vec{z}).P \mid x(\vec{z}).P \\ (\pi_F) & P & ::= & \mathbf{nil} \mid P|P \mid (\nu x)P \mid \overline{x}\langle\vec{y}\rangle.P \mid x\langle\vec{y}\rangle.P \end{array}$$

where, in the output and input processes, the round brackets $(\vec{z})$ denote distinct bound names and the angle brackets $\langle\vec{y}\rangle$ denote free names. The processes are subject to the standard structural congruence.

The reaction relations for the three calculi are generated by the rules

$$\begin{array}{lrcll} (\pi) & \overline{x}\langle\vec{y}\rangle.P|x(\vec{z}).Q & \searrow & P|Q\{\vec{y}/\vec{z}\} & \\ (\pi_I) & \overline{x}(\vec{z_1}).P|x(\vec{z_2}).Q & \searrow & (\nu\vec{z})(P\{\vec{z}/\vec{z_1}\}|Q\{\vec{z}/\vec{z_2}\}), & \vec{z} \text{ fresh} \\ (\pi_F) & (\nu\vec{u})(\overline{x}\langle\vec{y}\rangle.P|x\langle\vec{z}\rangle.Q|R) & \searrow & (\nu\vec{u})((P|Q|R)\sigma), & \text{side-conditions (see below)} \end{array}$$

Notice that the π_I reaction relation is symmetric in the behaviour of the output and input processes. The π_F reaction relation is also symmetric, but its binding encompasses R and occurs outside the output and input processes. The side-conditions express the following constraints: that the $\vec{u}$ are distinct and provide witnesses for the equivalence relation S generated by $\{\vec{y} = \vec{z}\}$; and that σ is a substitution with domain $\{\vec{u}\}$ and range $\{\vec{y}, \vec{z}\}$ such that σ *agrees with* S (that is, vSw iff $\sigma v = \sigma w$). The π_F reaction rule is similar to that for the fusion calculus, except that the fusion calculus additionally requires that the domain and range of σ do not intersect.

We have one symmetric action calculus SPIC for interpreting both the π-calculus and the π_I-calculus. It is specified by the signature $\mathbb{K} = (\{1\}, \{\mathbf{in}, \mathbf{out}\})$ with the arity rules

$$\frac{t : \epsilon \to m}{\mathbf{out}(t) : \epsilon \to 1} \qquad \frac{t : m \to \epsilon}{\mathbf{in}(t) : 1 \to \epsilon}$$

Given a process P we translate it to a SPIC term P'. The process **nil** is translated to the SPIC term $\mathbf{id}_\epsilon$. Parallel composition $|$ is translated to tensor $\otimes$. Restriction $(\nu x)P$ is translated to name-hiding $(P')_x$. The various different forms of input and output action are translated as follows:

$$\begin{array}{lcll} (\overline{x}\langle\vec{y}\rangle.P)' & = & \mathbf{out}(\vec{y}^+ \otimes P') \cdot x^- & \text{(non-binding output)} \\ (\overline{x}(\vec{z}).P)' & = & \mathbf{out}((\vec{z}^+ \otimes P')_{\vec{z}}) \cdot x^- & \text{(binding output)} \\ (x(\vec{z}).P)' & = & x^+ \cdot \mathbf{in}((\vec{z}^- \otimes P')_{\vec{z}}) & \text{(binding input)} \end{array}$$

The reaction relation for SPIC, illustrated in figure 6, is generated the rule

$$\mathbf{out}(s) \cdot x^- \otimes x^+ \cdot \mathbf{in}(t) \searrow s \cdot t$$

Reaction is not permitted inside **in** or **out**. The π-calculus and the π_I-calculus have different images in SPIC.

For π-processes P and Q we have the results that $P \equiv Q$ iff $P' = Q'$, and that $P \searrow Q$ iff $P' \searrow Q'$. The proofs for the reverse direction require weak normal forms for the symmetric action calculi, introduced in the next section. We believe that the results also hold for π_I processes.

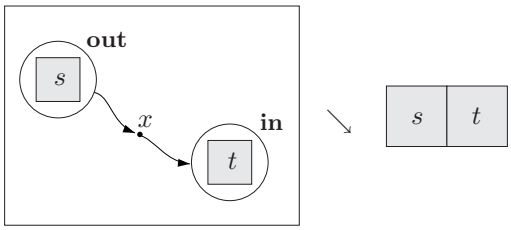


Figure 6: The reaction relation for SPIC.

To interpret the π_F-calculus we use a symmetric action calculus with the same controls as SPIC, but with the different reaction relation generated by the rule

$$(\mathbf{out}(\vec{y}^+ \otimes s) \cdot x^- \otimes x^+ \cdot \mathbf{in}(\vec{z}^- \otimes t) \otimes r)_{\vec{u}} \searrow (\vec{y}^+ \cdot \vec{z}^- \otimes s \otimes t \otimes r)_{\vec{u}}$$

where the $\vec{u}$ are distinct and provide witnesses for the equivalence relation generated by $\{\vec{y} = \vec{z}\}$. The translation of π_F-processes is as above, with the non-binding input translated by

$$(x\langle\vec{y}\rangle.P)' = x^+ \cdot \mathbf{in}(\vec{y}^- \otimes P') \quad \text{(non-binding input)}$$

The structural congruence is preserved by the translation. We are currently working on the connection between the reaction relations.

These examples indicate that the symmetric action calculi provide a natural framework for presenting at least some variants of the π-calculus. We are currently studying the presentation of other calculi including the ambient, spi, λ- and ν-calculi, referenced in the introduction. We believe that the restriction operator is an important part of any calculus which involves names, and that it is a positive feature that the symmetric action calculi interpret restriction directly.

4 Symmetric molecular forms

In this section we introduce *symmetric molecular forms*. Definitions 4.1 to 4.4 describe molecular terms and their equational theory; quotienting gives equivalence classes, called *molecular forms*. Proposition 4.9 shows that symmetric molecular forms are (weak) normal forms for the symmetric action calculus. The remainder of this section uses the molecular forms to prove that the symmetric action calculi conservatively extend the reflexive action calculi.

Definition 4.1 [Symmetric molecular terms] The set of symmetric molecular terms $\mathsf{SM}(\mathbb{K})$ over a signature $\mathbb{K}$ consist of terms a defined by the abstract grammar

$$\begin{array}{rcll} a & ::= & (\vec{x};\vec{\mu};\vec{y})_{R;\vec{z}} & a : |\vec{x}| \to |\vec{y}| \\ \mu_i & ::= & \vec{u}_i \cdot K\vec{a}_i \cdot \vec{v}_i & K\vec{a}_i : |\vec{u}_i| \to |\vec{v}_i| \end{array}$$

where $\vec{x}, \vec{y}, \vec{u}_i, \vec{v}_i$ are lists of names, $\vec{\mu}$ is a multiset $\mu_1, \ldots, \mu_r$ of *symmetric molecules*, $\vec{z}$ is a set of names, $K\vec{a}_i$ respects the arity of K given in $\mathbb{K}$, and R is an equivalence relation on names which respects arities and which is finitely generated (that is, R can be generated from a finite set of axioms).

An example molecular form and its corresponding graph are shown in figure 7. Notice that the syntax for molecular terms is closely related to the graphical presentation.

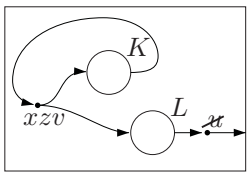


Figure 7: Graph corresponding to the molecular term $(\_; x \cdot K \cdot z, x \cdot L \cdot u; u)_{\{x=z, z=v\};u}$

In the molecular term $a = (\vec{x};\vec{\mu};\vec{y})_{R;\vec{z}}$, all names in $\vec{z}$ are *local*. The *free names* of a are

$$fn(a) = (\{\vec{x}, \vec{y}\} \cup fn(\vec{\mu}) \cup fn(R)) \backslash \{\vec{z}\}$$

where $fn(\vec{u}_i \cdot K\vec{a}_i \cdot \vec{v}_i) = \{\vec{u}_i, \vec{v}_i\} \cup fn(\vec{a}_i)$ and $fn(R)$ is the set of all names in R which are not in singleton equivalence classes. We write $\mathrm{Clo}(R)$ for the reflexive, symmetric and transitive closure of a relation R and write $\{x_1 = y_1, \ldots, x_r = y_r\}$ as short-hand for $\mathrm{Clo}(\{x_1 = y_1, \ldots, x_r = y_r\})$.

We wish to define substitution on molecular terms. To do this it is first necessary to define substitution on the equivalence relation R.

Definition 4.2 Let R and S be finitely-generated equivalence relations on names which respect arities. We define the following operations:

$$\begin{array}{rcl} R + S & \stackrel{\text{def}}{=} & \mathrm{Clo}\ (R \cup S) \\ R - v & \stackrel{\text{def}}{=} & \mathrm{Clo}\ \{(x,y) \text{ s.t. } xRy \text{ and } x \neq v \text{ and } y \neq v\} \\ \{u/v\}R & \stackrel{\text{def}}{=} & (R + \{u = v\}) - v \end{array}$$

It is a standard result that these relations are themselves finitely-generated equivalence relations which respect arities. The expected properties hold: $R + S = S + R$; $\sigma(R + S) = \sigma R + \sigma S$; and $(\{y/x\}R) + \{x = y\} = R + \{x = y\}$.

Substitution on molecular terms is now easy to define.

Definition 4.3 [Substitution] Given $\vec{\mu} = \mu_1, \ldots, \mu_r$, $a = (\vec{x}; \vec{\mu}; \vec{y})_{R;\vec{z}}$ and $\sigma = \{u/v\}$, we define $\sigma(a)$ by

$$\sigma(a) = \begin{cases} a & \text{if} \quad v \in \{\vec{z}\} \\ (\sigma\tau\vec{x}; \sigma\tau\vec{\mu}; \sigma\tau\vec{y})_{\sigma\tau R;\sigma\tau\vec{z}} & \text{if} \quad v \notin \{\vec{z}\} \end{cases}$$

where $\tau = \{u'/u\}$ for some fresh u', and $\sigma\mu_i = \sigma\vec{u}_i \cdot K\sigma\vec{a}_i \cdot \sigma\vec{v}_i$ for $\mu_i = \vec{u}_i \cdot K\vec{a}_i \cdot \vec{v}_i$.

The equational theory on symmetric molecular terms is generated by two simple axioms: if a local name is not used in the body it can be removed (or added), and if two names x and y are related then one can be substituted for the other.

Definition 4.4 [The theory SMAC] The *equational theory* SMAC is the set of equations upon terms generated by the axioms

$$\begin{array}{lrcll} \nu. & (\vec{x}; \vec{\mu}; \vec{y})_{R-v;\vec{z}} & = & (\vec{x}; \vec{\mu}; \vec{y})_{R;v,\vec{z}} & v \notin \{\vec{x}, \vec{y}, \vec{z}\} \cup fn(\vec{\mu}) \\ \sigma. & (\vec{x}; \vec{\mu}; \vec{y})_{R;\vec{z}} & = & (\sigma\vec{x}; \sigma\vec{\mu}; \sigma\vec{y})_{R;\vec{z}} & \sigma = \{u/v\}, uRv \end{array}$$

and the structural rules

$$\frac{\vec{a} = \vec{b}}{\vec{u} \cdot K\vec{a} \cdot \vec{v} = \vec{u} \cdot K\vec{b} \cdot \vec{v}} \qquad \frac{\vec{\mu} = \vec{\eta}}{(\vec{x}; \vec{\mu}; \vec{y})_{R;\vec{z}} = (\vec{x}; \vec{\eta}; \vec{y})_{R;\vec{z}}}$$

The *symmetric molecular forms* of a particular symmetric action calculus are the equivalence classes obtained by quotienting the molecular terms in $\mathsf{SM}(\mathbb{K})$ by the equational theory SMAC. The next step is to show that molecular forms correspond to actions in the symmetric action calculi. This is done in Proposition 4.9. First we need some definitions and lemmas. Their proofs are straightforward and have been omitted.

Lemma 4.5 [α-conversion] If $a = (\vec{x}; \vec{\mu}; \vec{y})_{R;v,\vec{z}}$, $u \notin fn(a)$ and $\sigma = \{u/v\}$, then

$$(\vec{x}; \vec{\mu}; \vec{y})_{R;v,\vec{z}} = (\sigma\vec{x}; \sigma\vec{\mu}; \sigma\vec{y})_{\sigma R;u,\vec{z}}$$

Definition 4.6 [Derived operators] Let $a = (\vec{x_1}; \vec{\mu}; \vec{x_2})_{R;\vec{x}}, b = (\vec{y_1}; \vec{\eta}; \vec{y_2})_{S;\vec{y}}, c = (\vec{x_1}\vec{y_1}; \vec{\mu}; \vec{x_2}\vec{y_2})_{R;\vec{z}}$. By α-conversion we may assume that the names in each term are disjoint from the local names in the others. Define

$$\begin{array}{rcll} \mathbf{id}_m & \stackrel{\text{def}}{=} & (\vec{x}; \_; \vec{x})_{\_;\vec{x}} & |\vec{x}| = m \\ a \cdot b & \stackrel{\text{def}}{=} & (\vec{x_1}; \vec{\mu}, \vec{\eta}; \vec{y_2})_{R+S+\{\vec{x_2}=\vec{y_1}\};\vec{x},\vec{y}} & |\vec{x_2}| = \vec{y_1}| \\ a \otimes b & \stackrel{\text{def}}{=} & (\vec{x_1}, \vec{y_1}; \vec{\mu}, \vec{\eta}; \vec{x_2}, \vec{y_2})_{R+S;\vec{x},\vec{y}} & \\ \mathbf{p}_{m,n} & \stackrel{\text{def}}{=} & (\vec{x}, \vec{y}; \_; \vec{y}, \vec{x})_{\_;\vec{x},\vec{y}} & |\vec{x}| = m, |\vec{y}| = n \\ x^- & \stackrel{\text{def}}{=} & (x; \_; \_)_{\_;\_} & \\ x^+ & \stackrel{\text{def}}{=} & (\_; \_; x)_{\_;\_} & \\ (a)_y & \stackrel{\text{def}}{=} & (\vec{x_1}; \vec{\mu}; \vec{x_2})_{R;\vec{x},y} & y \notin \{\vec{x}\} \\ \uparrow_m c & \stackrel{\text{def}}{=} & (\vec{y_1}; \vec{\mu}; \vec{y_2})_{R+\{\vec{x_1}=\vec{x_2}\};\vec{z}} & |\vec{x_1}| = |\vec{x_2}| = m \\ K(\vec{s}) & \stackrel{\text{def}}{=} & (\vec{x}; \vec{x} \cdot K(\vec{s}) \cdot \vec{y}; \vec{y})_{\_;\vec{x},\vec{y}} & K : |\vec{x}| \to |\vec{y}|, \vec{x}, \vec{y} \notin fn(\vec{s}) \end{array}$$

Using these derived operators we define a equality-preserving translations between the symmetric term algebra and the symmetric molecular forms. These translations are mutually inverse.

Definition 4.7 [Translation] The translation $[\![\cdot]\!] : \mathsf{ST}(\mathbb{K}) \to \mathsf{SM}(\mathbb{K})$ is defined inductively on the structure of symmetric terms using the operators in definition 4.6.

Definition 4.8 [Translation] The translation $\widehat{(\_)} : \mathsf{SM}(\mathbb{K}) \to \mathsf{ST}(\mathbb{K})$ is defined inductively on the structure of molecular forms by

$$(\vec{x}; \widehat{\mu_1, \ldots, \mu_r}; \vec{y})_{R;\vec{z}} = (\vec{x}^- \otimes \vec{y}^+ \otimes \bigotimes \widehat{\mu}_i \otimes \widehat{R})_{\vec{z}}$$

where $\widehat{\mu_i} = \vec{u_i}^+ \cdot K_i(\widehat{\vec{a}_i}) \cdot \vec{v_i}^-$ and $\widehat{R}$ is the tensor $\bigotimes_{x,y} x^+ \cdot y^-$ for each pair of different names x, y related in R.

Proposition 4.9

1. $s, t \in \mathsf{ST}(\mathbb{K})$ and $s = t$ in SAC implies $[\![s]\!] = [\![t]\!]$ in SMAC
2. $s, t \in \mathsf{SM}(\mathbb{K})$ and $a = b$ in SMAC implies $\widehat{a} = \widehat{b}$ in SAC
3. $\widehat{[\![s]\!]} = s$ in SAC
4. $[\![\widehat{a}]\!] = a$ in SMAC

It is also possible to show that the derived molecular operators satisfy the axioms for an action calculus given in Definition 2.2. The translation $\widehat{(\_)}$ moreover preserves those derived operators given in Definition 4.6.

Conservativity

We will use molecular forms to demonstrate that the symmetric action calculi conservatively extend the reflexive action calculi. First, we present an overview of reflexive molecular forms. A full description is given in [Mil94].

Definition 4.10 [Reflexive molecular terms] The reflexive molecular terms $\mathsf{RM}(\mathbb{K})$ over a signature $\mathbb{K}$ consist of terms a defined by the abstract grammar

$$\begin{array}{rcll} a & ::= & (\vec{x})[\mu_1 \cdots \mu_r]\langle \vec{y} \rangle & a : |\vec{x}| \to |\vec{y}| \\ \mu_i & ::= & \langle \vec{u}_i \rangle K \vec{a}_i(\vec{v}_i) & K\vec{a}_i : |\vec{u}_i| \to |\vec{v}_i| \end{array}$$

where $\vec{x}, \vec{y}, \vec{u}_i, \vec{v}_i$ are lists of names, $\vec{\mu}$ is a multiset $\mu_1, \ldots, \mu_r$ of *reflexive molecules* and $K \in \mathcal{K} \cup \{\boldsymbol{\nu}\}$ with $\boldsymbol{\nu} : \epsilon \to p$. The intention is that $\boldsymbol{\nu}$ generates a new name; graphically it is represented by a closed loop, as in Figure 8. An term is *well-formed* if the arities are respected and the names in $\vec{x}$ and each $\vec{v}_i$ are all mutually distinct; we will consider only well-formed terms. The equational theory RMAC is generated by α-conversion, and the removal or addition of superfluous $\boldsymbol{\nu}$s given by the axiom

$$(\vec{x})[\boldsymbol{\nu}(w), \vec{\mu}]\langle \vec{y} \rangle = (\vec{x})[\vec{\mu}]\langle \vec{y} \rangle \qquad w \text{ not free in } ()[\vec{\mu}]\langle \vec{y} \rangle$$

The *reflexive molecular forms* are the equivalence classes obtained by quotienting terms $\mathsf{RM}(\mathbb{K})$ by the theory RMAC.

The definition of a derived reflexion operator for molecular terms, used in Proposition 4.12 and in the conservativity result, is complicated. It requires a *reflexive substitution* which has the ability to introduce new names. This reflexive substitution $\{u/v\}^*$ is defined in terms of two other substitutions: the standard capture-avoiding substitution $\sigma = \{u/v\}$, and a syntactic substutition ${}^{u/v}$ which is not capture-avoiding at the top level.

Definition 4.11 [Reflexive substitution] Let $a = (\vec{x})[\vec{\mu}]\langle\vec{y}\rangle$, $\vec{\mu} = \mu_1, \ldots, \mu_r$ and $\mu_i = \vec{u}_i \cdot K_i \vec{a}_i \cdot \vec{v}_i$. Then

$$\{u/v\}^* a = \begin{cases} (\vec{x})[\boldsymbol{\nu}(v), \vec{\mu}]\langle\vec{y}\rangle & \text{if} \quad u = v, v \notin \{\vec{x}, \vec{v}_i\} \\ {}^{u/v}a & \text{if} \quad u \neq v, v \notin \{\vec{x}, \vec{v}_i\} \\ \text{undefined} & \text{if} \quad v \in \{\vec{x}, \vec{v}_i\} \end{cases}$$

where ${}^{u/v}a = (\sigma\vec{x})[{}^{u/v}\vec{\mu}]\langle\sigma\vec{y}\rangle$ and ${}^{u/v}\mu_i = \langle\sigma\vec{u}_i\rangle K_i(\sigma\vec{a}_i)(\sigma\vec{v}_i)$

The derived reflexion operator is now given by

$$\uparrow\ ((v, \vec{x})[\vec{\mu}]\langle u, \vec{y}\rangle) \stackrel{\text{def}}{=} \{u/v\}^*(\vec{x})[\vec{\mu}]\langle\vec{u}\rangle$$

The other operators are defined in [Mil94].

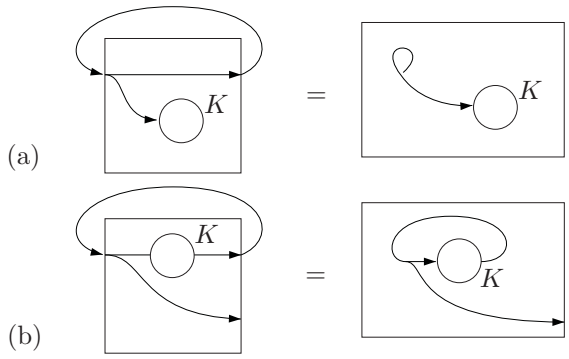


Figure 8: Two cases of reflexion in reflexive molecular forms. (a) In reflecting $\uparrow (x)[\langle x\rangle K]\langle x\rangle$, note that the reflexive substitution on the same name $\{x/x\}^*$ introduces a special ν control to give $()[\nu(x), \langle x\rangle K]\langle x\rangle$, and that this control corresponds to a closed loop in the graph. (b) Reflexive substitution on different names $\{y/x\}^*$ does not introduce new controls, as in $\uparrow (x)[\langle x\rangle K(y)]\langle yx\rangle = ()[\langle x\rangle K(x)]\langle x\rangle$.

Proposition 4.12 Just as in Proposition 4.9, we can define translations between reflexive molecular terms $\mathsf{RM}(\mathbb{K})$ and the reflexive terms $\mathsf{RT}(\mathbb{K})$ which are equality-preserving, mutually inverse and which preserve the operators.

The reflexive molecular forms are somewhat awkward: they use an extra control $\boldsymbol{\nu}$, they use the complicated reflexive substitution to derive the reflexion operator, and they require the bound names in each molecule to be distinct. In contrast our symmetric molecular forms do not require an extra control, there

is a simpler derived reflexion operator, and there are no constraints upon the local names. In this section we provide a conservative translation from reflexive molecular forms to symmetric molecular forms.

Definition 4.13 [Translation] Let $a = (\vec{x})[\mu_1, \ldots, \mu_r]\langle\vec{y}\rangle$ be an arbitrary reflexive molecular term, where $\mu_i = \langle\vec{u}_i\rangle K_i\vec{a}_i(\vec{v}_i)$. The translation $[\![\cdot]\!] : \mathsf{RM}(\mathbb{K}) \to \mathsf{SM}(\mathbb{K})$ is defined by

$$[\![a]\!] = (\vec{x}; [\![\mu_1]\!], \ldots, [\![\mu_r]\!]; \vec{y})_{I;\vec{x},\vec{v}_1,\ldots,\vec{v}_r}$$

where

$$[\![\mu_i]\!] = \begin{cases} \vec{u}_i \cdot K_i[\![\vec{a}_i]\!] \cdot \vec{v}_i & \text{if} \quad K \in \mathcal{K} \\ \_ & \text{if} \quad K = \boldsymbol{\nu} \end{cases}$$

Proposition 4.14 If $a, b \in \mathsf{RM}(\mathbb{K})$ and $a = b$ in $\mathsf{RMAC}(\mathbb{K})$ then $[\![a]\!] = [\![b]\!]$ in $\mathsf{SMAC}(\mathbb{K})$.

To prove the converse, we define a subset of the symmetric molecular terms which turns out to be the image of $[\![\_]\!]$. Let $\mathsf{SM}_R(\mathbb{K})$ denote the subset of $\mathsf{SM}(\mathbb{K})$ in which all symmetric terms have the form $(\vec{x}; \mu_1, \ldots, \mu_r; \vec{y})_{I;\vec{z}}$, where

- $\mu_i = \vec{u}_i \cdot K_i\vec{a}_i \cdot \vec{v}_i$ for $\vec{a}_i \in \mathsf{SM}_R(\mathbb{K})$,
- all the names in $\vec{x}$ and each $\vec{v}_i$ are distinct,
- and $\{\vec{x}, \vec{v}_1, \ldots, \vec{v}_r\} \subseteq \{\vec{z}\}$.

A key property of this subset is that if $a, b \in \mathsf{SM}_R(\mathbb{K})$ and $a = b$ in SMAC then a and b differ only by α-conversion and some superfluous local names in $\{\vec{z}\}$.

Definition 4.15 [Translation] Let $a = (\vec{x}; \mu_1, \ldots, \mu_r; \vec{y})_{I;\vec{x},\vec{v}_1,\ldots,\vec{v}_r,\vec{z}}$ with $a \in \mathsf{SM}_R(\mathbb{K})$ and $\mu_i = \vec{u}_i \cdot K_i\vec{a}_i \cdot \vec{v}_i$. The translation $|\cdot| : \mathsf{SM}_R(\mathbb{K}) \to \mathsf{RM}(\mathbb{K})$ is defined by

$$|a| = (\vec{x})\ [\boldsymbol{\nu}\vec{z}, \langle\vec{u}_1\rangle K_1|\vec{a}_1|(\vec{v}_1), \ldots, \langle\vec{u}_r\rangle K_r|\vec{a}_r|(\vec{v}_r)]\ \langle\vec{y}\rangle$$

Proposition 4.16 [Conservativity]

1. $a, b \in \mathsf{SM}_R(\mathbb{K})$ and $a = b$ in SMAC implies $|a| = |b|$ in RMAC.
2. $a \in \mathsf{RM}(\mathbb{K})$ implies $|[\![a]\!]| = a$ in RMAC.
3. $a \in \mathsf{SM}_R(\mathbb{K})$ implies $[\![|a|]\!] = a$ in SMAC.

Propositions 4.16 and 4.14, together with the fact that the translations preserve the operators, imply that the symmetric action calculi conservatively extend the reflexive action calculi.

5 Combinatoric Presentation

Ultimately we would like to develop categorical models for the symmetric action calculi. A significant step towards this end is the combinatoric presentation given in this section. This involves expressing name-hiding as a derived operator rather than as a primitive. Our results illustrate the expressive power of the name-hiding operator. They are analogous to the results given in [Gar99b] for non-symmetric action calculi.

To express name-hiding as a derived operator, we push it progressively inside terms until it can be dismissed. For example, consider the symmetric term $(x^- \otimes x^+ \otimes K(x^+))_x : p \otimes m \to p \otimes n$ corresponding to graph (a) of figure 9. Our technique is to declare a family of controls K_m indexed by arity information. This index records the arity of those names that have been pushed inside. Applying this technique to (a) gives $K_p((x^- \otimes x^+ \otimes x^+)_x) : p \otimes m \to p \otimes n$, which is shown in graph (b). Finally the name-hiding can be dismissed since the contents of this control are simply a **copy** operator, as can be seen in graph (c). By dismissing name-hiding in this way, we can express every symmetric term using just the operators $\mathbf{id}, \mathbf{p}, x^+, x^-, \mathbf{copy}, \mathbf{merge}, \mathbf{disc}$ and $\mathbf{new}$, along with indexed controls. We therefore choose these as the combinators for the symmetric action calculi.

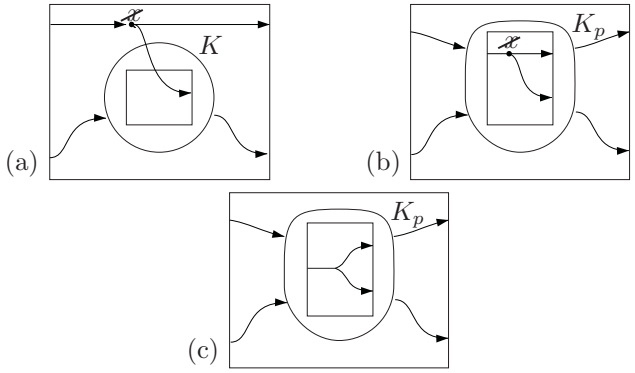


Figure 9: In this sequence of graphs we push name-restriction progressively inside controls until it can be dismissed. We use this technique in defining a combinatoric presentation for symmetric action calculi.

Definition 5.1 [Combinators] The set of *combinatory terms* over signature $\mathbb{K}$, denoted by $\mathsf{CT}(\mathbb{K})$, is constructed from the basic operators: identity $\mathbf{id}_m$, composition $\cdot$, tensor $\otimes$, permutation $\mathbf{p}_{m,n}$, datum x^+, co-datum x^-, copy $\lhd_m$, merge $\rhd_m$, discard $\boldsymbol{\omega}_m$, new $\boldsymbol{\nu}_m$ and the indexed controls K_m. A combinator t

is assigned arity $t : m \to n$, for arities m and n, using the rules

$$\mathbf{id}_m : m \to m \qquad \frac{s : k \to l \quad t : l \to m}{s \cdot t : k \to m} \qquad \frac{s : k \to m \quad t : l \to n}{s \otimes t : k \otimes l \to m \otimes n}$$

$$\mathbf{p}_{m,n} : m \otimes n \to n \otimes m$$

$$x^+ : \epsilon \to p, \quad x{:}p \qquad x^- : p \to \epsilon, \quad x{:}p$$

$$\lhd_m : m \to m \otimes m \qquad \rhd_m : m \otimes m \to m$$

$$\boldsymbol{\omega}_m : m \to \epsilon \qquad \boldsymbol{\nu}_m : \epsilon \to m$$

$$\frac{t_1 : m_1 \to n_1 \quad \dots \quad t_r : m_r \to n_r}{K_m(t_1, \dots, t_r) : k \to l} \quad K : ((m_1, n_1), \dots, (m_r, n_r)) \to (k, l) \text{ in } \mathbb{K}$$

We choose to assign prime arities to names, since it makes the connection with the original presentation clearer. It would also be possible to allow names to have arbitrary arities.

The axioms for the wiring terms are intuitive. The axioms for the controls are necessary to permit the movemement of wiring terms inside indexed controls. They are illustrated in figure 10.

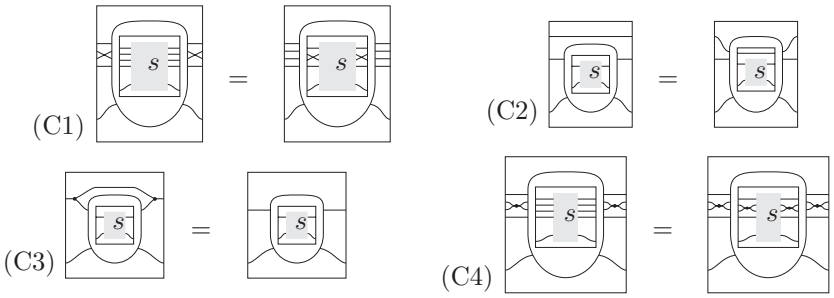


Figure 10: Four of the combinator axioms. They are given algebraically in Definition 5.2.

Definition 5.2 [The theory SC] The equational theory SC is the set of equations upon terms generated by the axioms of a strict symmetric monoidal category given in definition 2.2 and the following axioms:
1. the axioms for copy, merge, new and discard (we write $\lhd_{mn}$ as shorthand for

$\lhd_{m\otimes n}$)

$B1.$ $\lhd_{mn} = (\lhd_m \otimes \lhd_n) \cdot (\mathbf{id}_m \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}_n)$
$\rhd_{mn} = (\mathbf{id}_m \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}_n) \cdot (\rhd_m \otimes \rhd_n)$

$B2.$ $\boldsymbol{\omega}_{mn} = \boldsymbol{\omega}_m \otimes \boldsymbol{\omega}_n$
$\boldsymbol{\nu}_{mn} = \boldsymbol{\nu}_m \otimes \boldsymbol{\nu}_n$

$B3.$ $\lhd_m \cdot (\boldsymbol{\omega}_m \otimes \mathbf{id}_m) = \mathbf{id}_m$
$(\boldsymbol{\nu}_m \otimes \mathbf{id}_m) \cdot \rhd_m = \mathbf{id}_m$

$B4.$ $\lhd_m \cdot \mathbf{p}_{m,m} = \lhd_m$
$\mathbf{p}_{m,m} \cdot \rhd_m = \rhd_m$

$B5.$ $\lhd_m \cdot (\lhd_m \otimes \mathbf{id}_m) = \lhd_m \cdot (\mathbf{id}_m \otimes \lhd_m)$
$(\rhd_m \otimes \mathbf{id}_m) \cdot \rhd_m = (\mathbf{id}_m \otimes \rhd_m) \cdot \rhd_m$

$B6.$ $\boldsymbol{\nu}_m \cdot \boldsymbol{\omega}_m = \mathbf{id}_\epsilon$

$B7.$ $\rhd_m \cdot \lhd_m = (\lhd_m \otimes \mathbf{id}_m) \cdot (\mathbf{id}_m \otimes \rhd_m) = (\mathbf{id}_m \otimes \lhd_m) \cdot (\rhd_m \otimes \mathbf{id}_m)$

$B8.$ $\lhd_m \cdot \rhd_m = \mathbf{id}_m$

2. the axioms for naming, where $x : p$

$B9.$ $x^+ \cdot x^- = \mathbf{id}_\epsilon$

$B10.$ $x^+ \cdot \lhd_p = x^+ \otimes x^+$
$\rhd_p \cdot x^- = x^- \otimes x^-$

$B11.$ $x^+ \cdot \boldsymbol{\omega}_p = \mathbf{id}_\epsilon$
$\boldsymbol{\nu}_p \cdot x^- = \mathbf{id}_\epsilon$

3. the control axioms (again, K_{kmnl} is shorthand for $K_{k\otimes m\otimes n\otimes l}$)

$C1.$ $(\mathbf{id}_k \otimes \mathbf{p}_{n,m} \otimes \mathbf{id}) \cdot K_{kmnl}(\vec{s}) \cdot (\mathbf{id}_k \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}) = K_{knml}((\mathbf{id}_k \otimes \mathbf{p}_{n,m} \otimes \mathbf{id}) \cdot \vec{s} \cdot (\mathbf{id}_k \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}))$

$C2.$ $\mathbf{id}_m \otimes K_n(\vec{s}) = K_{m,n}(\mathbf{id}_m \otimes \vec{s})$

$C3.$ $(\lhd_m \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes K_m(\vec{s})) \cdot (\rhd_m \otimes \mathbf{id}) = K_m(\vec{s})$

$C4.$ $(\mathbf{id}_k \otimes \rhd_m \cdot \lhd_m \otimes \mathbf{id}) \cdot K_{kmml}(\vec{s}) \cdot (\mathbf{id}_k \otimes \rhd_m \cdot \lhd_m \otimes \mathbf{id}) = (\mathbf{id}_k \otimes \rhd_m \cdot \lhd_m \otimes \mathbf{id}) \cdot$
$K_{kmml}((\mathbf{id}_k \otimes \rhd_m \cdot \lhd_m \otimes \mathbf{id}) \cdot \vec{s} \cdot (\mathbf{id}_k \otimes \rhd_m \cdot \lhd_m \otimes \mathbf{id})) \cdot (\mathbf{id}_k \otimes \rhd_m \cdot \lhd_m \otimes \mathbf{id})$

$C5.$ $(x^+ \otimes \mathbf{id} \otimes \mathbf{id}) \cdot K_{km}(\vec{s}) \cdot (x^- \otimes \mathbf{id} \otimes \mathbf{id}) = K_m((x^+ \otimes \mathbf{id}) \cdot \vec{s} \cdot (x^- \otimes \mathbf{id}))$

Remark 5.3 The axioms $B1$-$B7$ correspond to a known categorical structure called a *Frobenius algebra*. This structure combined with the additional axiom $B8$ has a free characterisation previously discovered by Hyland. Finding a categorical interpretation of the control axioms remains the subject of ongoing discussion.

Definition 5.4 [Name-hiding] Let $s : m \to n$ be a combinator in $\mathsf{CT}(\mathbb{K})$ and let $x : p$. The derived *name-hiding operator* $(s)_x : m \to n$ is given by

$$(s)_x \stackrel{\text{def}}{=} (\boldsymbol{\nu}_p \otimes \mathbf{id}) \cdot [x^- \otimes x^+ \otimes s]_x \cdot (\mathbf{disc}_p \otimes \mathbf{id})$$

where $[x^- \otimes x^+ \otimes s]_x$ is defined inductively on the structure of s as follows:

$$\begin{array}{rcl}
[x^- \otimes x^+ \otimes s]_x & \stackrel{\text{def}}{=} & \mathbf{id}_p \otimes s, \qquad x \notin fn(s) \\
[x^- \otimes x^+ \otimes x^+]_x & \stackrel{\text{def}}{=} & \lhd_p \\
[x^- \otimes x^+ \otimes x^-]_x & \stackrel{\text{def}}{=} & \rhd_p \\
[x^- \otimes x^+ \otimes (s_1 \cdot s_2)]_x & \stackrel{\text{def}}{=} & [x^- \otimes x^+ \otimes s_1]_x \cdot [x^- \otimes x^+ \otimes s_2]_x \\
[x^- \otimes x^+ \otimes (s_1 \otimes s_2)]_x & \stackrel{\text{def}}{=} & (\lhd_p \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \mathbf{p}_{p,m} \otimes \mathbf{id}) \cdot ([x^- \otimes x^+ \otimes s_1]_x \otimes [x^- \otimes x^+ \otimes s_2]_x) \cdot \\
 & & (\mathbf{id} \otimes \mathbf{p}_{k,p} \otimes \mathbf{id}) \cdot (\rhd_p \otimes \mathbf{id}) \\
[x^- \otimes x^+ \otimes K_m(\vec{s})]_x & \stackrel{\text{def}}{=} & K_{p \otimes m}([x^- \otimes x^+ \otimes \vec{s}]_x)
\end{array}$$

Using this name-hiding operator, it is simple to define translations $[\![\cdot]\!] : \mathsf{ST}(\mathbb{K}) \to \mathsf{CT}(\mathbb{K})$ and $|\cdot| : \mathsf{CT}(\mathbb{K}) \to \mathsf{ST}(\mathbb{K})$ by induction on the structure of the terms. The only interesting cases are those of controls, namely

$$[\![K(t_1, \ldots, t_r)]\!] = K_\epsilon([\![t_1]\!], \ldots, [\![t_r]\!])$$

which has an empty index since no name-hiding operator has yet been pushed inside, and

$$|K_m(t_1, \ldots, t_r)| = (\vec{x}^- \otimes \vec{x}^+ \otimes K((\vec{x}^+ \otimes \mathbf{id}) \cdot |\vec{t_i}| \cdot (\vec{x}^- \otimes \mathbf{id})))_{\vec{x}}$$

with $\vec{x}$ fresh and $|\vec{x}| = m$.

Proposition 5.5

1. $s, t \in \mathsf{ST}(\mathbb{K})$ and $s = t$ in SAC implies $[\![s]\!] = [\![t]\!]$ in SC.
2. $s, t \in \mathsf{CT}(\mathbb{K})$ and $s = t$ in SC implies $|s| = |t|$ in SAC.
3. Given $t \in \mathsf{ST}(\mathbb{K})$, we have $|[\![t]\!]| = t$ in SAC.
4. Given $t \in \mathsf{CT}(\mathbb{K})$, we have $[\![|t|]\!] = t$ in CS.

6 Conclusion and additional work

To conclude, we briefly describe the connection between Yoshida's process graphs [Yos95] and non-nested symmetric action calculi. Process graphs extend Lafont's interaction nets [Laf97] to allow for non-determinism. Using these graphs, Yoshida shows that a minimal set of combinators can be used to represent the π-calculus [Yos98]. Figure 11 illustrates some process graphs. The nodes K, L, M are the same as nodes in interaction nets. As with interaction nets, the nodes have distinguished principle ports. The joins in the wires are named; these names may be free or hidden. Reaction occurs between two nodes that are joined on their principal ports.

There are three operations on process graphs, also illustrated in Figure 11:

juxtaposition, which allows two graphs to be placed side-by-side if they have distinct free names;

connection which, given a graph G and a partition Σ of the free names in G, allows the names equal in Σ to be joined together and renamed; and

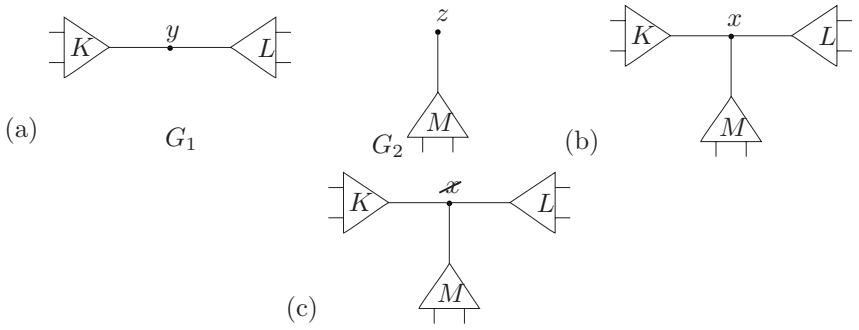


Figure 11: (a) Juxtaposition of graphs G_1 and G_2 is simply their tensor composition, in the case where their free names are distinct. (b) We may connect the graphs with a partition $y = z$ and then rename the join, in this case by the name x. The partition corresponds to the equivalence relation R in our symmetric molecular forms, and relabelling is given by the σ- and ν-axioms. (c) Hiding in the process graphs corresponds directly to name-hiding in the symmetric action calculi, and is illustrated by crossing out the name in question.

hiding, which removes some of the names.

Using an argument similar to that which connects interaction nets with simple directed graphs [Gar99a], we can show that Yoshida's graphs corresponds to non-nested symmetric molecular forms with the shape $(\_; \vec{\mu}; \_)_{I;\vec{z}}$ for $\{\vec{z}\} \subseteq fn(\vec{\mu})$.

In this paper we have introduced the symmetric action calculi. Our framework differs from the action calculus framework in its use of name-hiding as a primitive operator. We have shown that the π-calculus, the π_I-calculus and a variant of the fusion calculus all have natural presentations in our framework. We have proved that the symmetric action calculi conservatively extend the reflexive action calculi. Finally, we have provided a combinatory version of the symmetric action calculi: this is a step towards a categorical formulation of the graphs. We are currently looking at the presentation of other calculi based on names and name-restriction.

Acknowledgements We would like to thank J. Leifer and D. Wischik for their help and patience.

References

[AG97] Martín Abadi and Andrew D. Gordon. A calculus for cryptographic protocols: The spi calculus. In *Proceedings of the Fourth ACM Conference on Computer and Communications Security, Zürich*, pages 36–47. ACM Press, April 1997.

[Bou92] Gérard Boudol. Asynchrony and the π-calculus (note). Rapport de Recherche 1702, INRIA Sofia-Antipolis, May 1992.

[CG98] Luca Cardelli and Andrew D. Gordon. Mobile ambients. In *Proc. of Foundations of Software Science and Computation Structures (FoSSaCS), ETAPS'98, LNCS 1378*, pages 140–155, March 1998.

[Gar99a] P. Gardner. Graphical presentations of interactive systems. Submitted for publication; available at `http://www.cl.cam.ac.uk/~pag20/`, January 1999.

[Gar99b] Philippa Gardner. Closed action calculi. *Theoretical Computer Science*, 1999. To appear.

[GH97] Philippa Gardner and Masahito Hasegawa. Types and models in higher-order action calculi. In *Proceedings of TACS 97, Sendai, Japan*, 1997.

[HT92] Kohei Honda and Mario Tokoro. On asynchronous communication semantics. *Lecture Notes in Computer Science*, 612, 1992.

[Jen99] O. Jensen. Ph.D. thesis, University of Cambridge. 1999. In preparation.

[JSV96] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3), 1996.

[Laf97] Yves Lafont. Interaction combinators. *Information and Computation*, 137(1):69–101, 1997.

[Mil94] R. Milner. Action calculi V: Reflexive Action Calculi. Manuscript, 1994.

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

[Mil99] R. Milner. *Communicating and Mobile Systems: The Pi Calculus*. Cambridge University Press, 1999. To appear.

[MPW92] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes. *Information and Control*, 100:1–77, 1992.

[PS93] A. M. Pitts and I. D. B. Stark. Observable properties of higher order functions that dynamically create local names, or: What's new? *Lecture Notes in Computer Science*, 711, 1993.

[San96] D. Sangiorgi. Pi-calculus, internal mobility and agent-passing calculi. *Theoretical Computer Science*, 167(2), 1996.

[Sew97] Peter Sewell. Global/local subtyping for a distributed π-calculus. Technical Report 435, University of Cambridge, August 1997. Available from `http://www.cl.cam.ac.uk/users/pes20/`.

[VP98] B. Victor and J. Parrow. Concurrent constraints in the fusion calculus. In *Proceedings of ICALP'98*, number 1443 in Lecture Notes in Computer Science, pages 455–469. Springer-Verlag, 1998.

[Yos95] N. Yoshida. Graph notation for concurrent combinators. *Lecture Notes in Computer Science*, 907:393–412, May 1995.

[Yos98] N. Yoshida. Minimality and separation results on asynchronous mobile processes: Representability theorems by concurrent combinators. In *Proceedings of CONCUR '98, LNCS 1466*, pages 131–146, Springer-Verlag, 1998.