Theoretical Computer Science

Theoretical Computer Science 228 (1999) 77–103

www.elsevier.com/locate/tcs

Closed action calculi ☆

Philippa Gardner *

Computer Science Laboratory, University of Cambridge, New Museums Site, Pembroke Site, Cambridge, CB2 3QG, UK

Abstract

Action calculi provide a framework for capturing many kinds of interactive behaviour by focussing on the primitive notion of names. We introduce a name-free account of action calculi, called the closed action calculi, and show that there is a strong correspondence between the original presentation and the name-free presentation. We also add free names plus natural axioms to the closed world, and show that the abstraction operator can be constructed as a derived operator. Our results show that in some sense names are inessential. However, the purpose of action calculi is to understand formalisms which mimic the behaviour of interactive systems. Perhaps more significantly therefore, these results highlight the important presentational role that names play. © 1999 Published by Elsevier Science B.V. All rights reserved.

Keywords: Action calculi; Names; Combinatoric account

1. Introduction

Action calculi arose directly from the π-calculus and were introduced by Milner in 1993. They give a notation for capturing many kinds of interactive behaviour by focussing on the primitive notion of names. Names describe communication channels (or pointers or identifiers or locations) between agents, concepts fundamental to interactive systems. There are many calculi which use names for describing interactive behaviour, including the π-calculus [21], the λ-calculus, several models of distributed migratory systems [6, 25], the spi-calculus used for describing security protocols [2] and the object calculus [1]. Action calculi provide a framework for investigating all these calculi in a unified setting. Such a unification is necessary for the analysis of the similarities and differences between the many possible and existing calculi, and

☆ The author acknowledges support of a Royal Society of Edinburgh BP Fellowship and an EPSRC Advanced Fellowship.

* E-mail: philippa.gardner@cl.cam.ac.uk.

PII: S0304-3975(98)00355-7

to allow the common study of metatheory, such as behavioural congruences between agents.

An action calculus consists of a set of actions, constructed from constants which determine the specific action calculus under consideration, and a reaction relation which describes the interactive behaviour of these constants. Unlike the π-calculus, action calculi have two simple constructs for naming – $\langle x\rangle$ for using name x and $\mathbf{ab}_x(a)$ for binding x to the action a. This paper shows that we can give a name-free presentation of action calculi, called the closed action calculi. First, we show that there is a strong correspondence between closed action calculi and Milner's original presentation, which develops the results presented in [9]. Second, we add free names plus natural axioms to the closed world, and show that the abstraction operator can be constructed as a derived operator. This second result is analogous to the standard connection between the λ-calculus and combinatory logic, given for example in [13]. Just after this result was proved, Dusko Pavlovic [23] independently added free names to his models of closed action calculi and showed a similar result. In addition, he pointed out that the results are analogues of the standard categorical notion of functional completeness for cartesian categories.

Our results show that in some sense names are inessential. However, the purpose of action calculi is to understand formalisms which mimic the behaviour of interactive systems. Perhaps more significantly therefore, we shall see that these results highlight the important presentational role that names play.

In Section 2, we give an introduction to action calculi to make the paper self-contained. Further details can be found in [20]. In Section 3, we introduce the closed action calculi. Section 4 contains the translations and results; the main proofs are given in the appendix. Section 5 extends the closed action calculi with names and shows that the abstraction operator is derivable. In Section 6 we extend our results to the reflexive and higher-order action calculi, and in Section 7 we assess our results and discuss related research.

2. Action calculi

There are three presentations of the original definition of action calculi: the *algebraic* presentation, where actions are equivalence classes arising from a set of terms and an equational theory on terms, the *graphical* presentation, which gives an intuitive account using pictures, and the *molecular form* presentation, which give (weak) normal forms for the algebraic terms and a direct syntax for the graphs. We concentrate on the algebraic presentation in this paper; details of the other presentations are given in [20].

In all the presentations, an action calculus is specified by a *signature* $\mathbb{K}=(P,\mathscr{K})$, which consists of a set P of basic types, called *primes* and denoted by $p,q,\ldots$, and a set $\mathscr{K}$ of constants, called *controls*. Each control K in $\mathscr{K}$ has an associated arity $((m_1,n_1),\ldots,(m_r,n_r))\rightarrow(m,n)$, where the m's and n's are finite sequences of primes, called *tensor arities*; we write ε for the empty sequence, $\otimes$ for concatenation using infix notation, and write M for the set of tensor arities. We usually refer to the tensor arities as just arities, when the meaning is clear. We assume a fixed denumerable set X of *names*, each of which has a prime arity. We let $x, y, \ldots$ range over names, and sometimes write x^p to indicate that x has the prime arity p.

Definition 2.1 (*Terms*). The set of *terms* over signature $\mathbb{K}$, denoted by $T(\mathbb{K})$, is constructed from the basic operators: *identity* $\mathbf{id}_m$, *composition* $\cdot$, *tensor* $\otimes$, *discard* $\boldsymbol{\omega}_p$, *datum* $\langle x^p \rangle$, *abstraction* $\mathbf{ab}_{x^p}$, and the *controls* K. A term t is assigned an arity $t : m \to n$, for arities m and n, using the following rules:

$$\mathbf{id}_m : m \to m$$

$$\frac{s : k \to l \qquad t : l \to m}{s \cdot t : k \to m}$$

$$\frac{s : k \to m \qquad t : l \to n}{s \otimes t : k \otimes l \to m \otimes n}$$

$$\boldsymbol{\omega}_p : p \to \varepsilon$$

$$\langle x^p \rangle : \varepsilon \to p$$

$$\frac{t : m \to n}{\mathbf{ab}_{x^p}(t) : p \otimes m \to p \otimes n}$$

$$\frac{t_1 : m_1 \to n_1 \quad \cdots \quad t_r : m_r \to n_r}{K(t_1, \ldots, t_r) : m \to n}$$

where, in the *control term* $K(t_1, \ldots, t_r)$, the arity of K is $((m_1, n_1), \ldots, (m_r, n_r)) \to (m, n)$.

If a term contains no control terms, we call it a *wiring term*. We write $T_X(\mathbb{K})$ to emphasise the underlying set of names X, and $T_{\boldsymbol{x}}(\mathbb{K})$ to denote the set of terms whose free names are contained in $\{\boldsymbol{x}\}$. We omit the arity subscripts on the basic operators when apparent. The notions of *free* and *bound name* are standard: $\mathbf{ab}_x$ binds x and $\langle x \rangle$ represents a free occurrence of x. We write $s\{t/\langle x \rangle\}$ to denote the usual capture-avoiding substitution. The set of names free in $s, t, \ldots$ is denoted by $fn(s, t, \ldots)$. Given a possibly empty sequence of names $\boldsymbol{x} = x_1^{p_1}, \ldots, x_r^{p_r}$, we write $|\boldsymbol{x}|$ for $p_1 \otimes \cdots \otimes p_r$. All terms and expressions used are well formed, and all equations are between terms of the same arity.

Definition 2.2 (*Derived operations*). To help us define the equational theory, we give an alternative form of *abstraction* $(x)t$, the *permutations* $\mathbf{p}_{m,n}$, and some other standard abbreviations as follows:

$$(x)t \stackrel{\text{def}}{=} \mathbf{ab}_x(t)\cdot(\omega\otimes\mathbf{id})$$

$$(\boldsymbol{x})t \stackrel{\text{def}}{=} (x_1)\cdots(x_r)t, \qquad (\boldsymbol{x}=[x_1,\ldots,x_r],\ \text{all distinct},\ r\geqslant 1)$$

$$\langle\boldsymbol{x}\rangle \stackrel{\text{def}}{=} \langle x_1\rangle\otimes\cdots\otimes\langle x_r\rangle, \quad (\boldsymbol{x}=[x_1,\ldots,x_r],\ r\geqslant 1)$$

$$\mathbf{p}_{m,n} \stackrel{\text{def}}{=} (\boldsymbol{x},\boldsymbol{y})\langle\boldsymbol{y},\boldsymbol{x}\rangle, \qquad (|\boldsymbol{x}|=m,\ |\boldsymbol{y}|=n)$$

We assume that $(\ )t$ denotes the term t and $\langle\ \rangle$ denotes the term $\mathbf{id}_\varepsilon$. Notice that $\mathbf{p}_{m,n}$ is defined using *particular* names; with α-conversion, we shall be justified in choosing these names at will. Throughout this paper we shall adopt the convention that all names appearing in a vector within round brackets are distinct.

The equational theory for action calculi consists of a set of equations upon terms generated by the *action structure* axioms and the *concrete* axioms, given in Definition 2.3. The action structure axioms, introduced in [17], state that an action calculus is a strict monoidal category whose objects are given by arities and whose morphisms are defined by terms, with an endofunctor given by the $\mathbf{ab}_x$ operator. The concrete axioms describes the interplay between the free and bound names.

Definition 2.3 (*The theory AC*). The *equational theory* AC is the set of equations upon terms generated by the following axioms:

1. *The action structure axioms*:

A1: $s\cdot\boldsymbol{id}=s=\boldsymbol{id}\cdot s$

A2: $s\otimes\boldsymbol{id}_\varepsilon=s=\boldsymbol{id}_\varepsilon\otimes s$

A3: $\boldsymbol{id}\otimes\boldsymbol{id}=\boldsymbol{id}$

A4: $s\cdot(t\cdot u)=(s\cdot t)\cdot u$

A5: $s\otimes(t\otimes u)=(s\otimes t)\otimes u$

A6: $(s\cdot t)\otimes(u\cdot v)=(s\otimes u)\cdot(t\otimes v)$

A7: $\boldsymbol{ab}_x(\boldsymbol{id})=\boldsymbol{id}$

A8: $\boldsymbol{ab}_x(s\cdot t)=\boldsymbol{ab}_x(s)\cdot\boldsymbol{ab}_x(t)$

2. *The concrete axioms*:

γ: $(x)t=\omega\otimes t$ $\quad(x\notin fn(t))$

δ: $(x)(\langle x\rangle\otimes\mathbf{id}_m)=\mathbf{id}_{p\otimes m}$ $\quad(x^p)$

ζ: $\mathbf{p}_{k,m}\cdot(t\otimes s)=(s\otimes t)\cdot\mathbf{p}_{l,n}$ $\quad(s:k\to l,\ t:m\to n)$

σ: $(\langle y\rangle\otimes\mathbf{id}_m)\cdot(x)t=t\{\langle y\rangle/\langle x\rangle\}$ $\quad(x^p, y^p)$

Remark 2.4. For historical reasons, we have chosen to consider the operator $\mathbf{ab}_x$ as primitive, and define the operator (x) in terms of $\mathbf{ab}_x$. An alternative approach is to treat (x) as primitive, and let $\mathbf{ab}_x$ be defined by $\mathbf{ab}_x(t)\stackrel{\text{def}}{=}(x)(\langle x\rangle\otimes t)$. In fact, there is a slightly simpler presentation using the alternative binding. Hasegawa [10] observed that the equational theory AC can be generated by the axioms of a symmetric monoidal category, the σ-axiom and a stronger δ'-axiom

$$(x)(\langle x\rangle \otimes \mathbf{id}) \cdot t) = t, \quad x \notin fn(t)$$

Definition 2.5. The *static* part of an action calculus AC($\mathbb{K}$) consists of the equivalence classes, called *actions* and denoted by $a, b, c, \ldots$, obtained by quotienting the terms in Definition 2.1 by the equational theory. We sometimes write $\mathrm{AC}_X(\mathbb{K})$ to indicate that the names have come from the set X. The *dynamic* part of an action calculus AC($\mathbb{K}$), or the *reaction relation*, is a transitive relation between terms with the same arity which is preserved under tensor, composition, abstraction, and equality, and such that **id** does not react: that is, there is no s with $\mathbf{id} \searrow s$.

The definition of the dynamics has purposely been kept general, since it is on-going research to fully understand which dynamic relations describe interesting behaviour. It is typically generated from a set of rewrite rules. Notice that, since the reaction relation is preserved by the equational theory, an equivalent way of stating the dynamics is to define a relation on the actions as in [20]. Also notice that since **id** does not react, it follows for arbitrary wiring term u that u does not react. This fact is a corollary of the following lemma.

Proposition 2.6. *Given wiring term $u \in T(\mathbb{K})$ with arity $m \to n$, there exists wiring terms v_1, v_2 such that $v_1 \cdot u \cdot v_2 = \mathbf{id}_\varepsilon$.*

Proof. It is easy to prove that $u = (\boldsymbol{x})\langle \boldsymbol{y}\rangle$ in AC for $|\boldsymbol{x}| = m$ and $|\boldsymbol{y}| = n$, by induction on the structure of wiring term u. Let $v_1 = \langle \boldsymbol{x}\rangle$ and $v_2 = (\boldsymbol{y})\langle\ \rangle$ to obtain the result. □

Example 2.7. We use a simple version of the asynchronous π-calculus [5, 14] as a running example throughout the paper. The set of processes Proc are given by the abstract grammar

$$P ::= 0 \mid P \mid Q \mid \bar{x}\langle y\rangle \mid x(z).P \mid v(z)P \tag{1}$$

These represent respectively the null process, parallel composition, the output of a name y on channel name x, the process which can input a name along x, bind it to z and become P, and a declaration of a new private channel which binds z in P. The act of passing a name to a process is described by the rule

$$\bar{x}\langle y\rangle \mid x(z).P \to P\{y/z\} \tag{2}$$

The action calculus PIC for the asynchronous π-calculus is specified by the signature $\mathbb{K} = (\{1\}, \{\mathbf{out}, \mathbf{in}, \mathbf{new}\})$, where the controls have arity rules

$$\mathbf{out} : 1 \otimes 1 \to \varepsilon \qquad \frac{t : 1 \to 1}{\mathbf{in}(t) : 1 \to 1} \qquad \mathbf{new} : \varepsilon \to 1$$

Each control corresponds to a construct in (1), as shown by the following translation $(\_)' : \mathrm{Proc} \to T(\mathbb{K})$:

$$0' = \mathbf{id}_\varepsilon$$

$$(P \mid Q)' = P' \otimes Q'$$

$$(\bar{x}\langle y \rangle)' = \langle x, y \rangle \cdot \mathbf{out}$$

$$(x(z).P)' = \langle x \rangle.\mathbf{in}((z)P')$$

$$(\nu(z)P)' = \mathbf{new} \cdot (z)P'$$

The rule generating the reaction relation for PIC is

$$\langle x, y \rangle \cdot \mathbf{out} \otimes \langle x \rangle \cdot \mathbf{in}(t) \searrow \langle y \rangle \cdot t. \tag{3}$$

Notice that when t is $(z)t'$, we have $\langle y \rangle \cdot (z)t' = t'\{\langle y \rangle / \langle z \rangle\}$. A full account of the connection between this version of the asynchronous π-calculus and PIC is given in [20], including some extensions to the basic version given here.

Example 2.8. We also give the action calculus LAMB, introduced by Milner in [19], as a step towards defining the higher-order action calculi given in Section 6. Gardner and Hasegawa [10] have shown that it is related to a simply-typed call-by-value λ-calculus arising from Moggi's computational λ-calculus [22]. Given a set of basic primes P, we define the sets of higher-order primes $P^{\Rightarrow}$ and higher-order arities $M^{\Rightarrow}$ by the abstract grammars

$$p ::= p' \in P \mid m \Rightarrow n$$

$$m ::= p \mid m \otimes n \mid \varepsilon$$

The action calculus LAMB is specified by the signature $\mathbb{K} = \{P^{\Rightarrow}, \{\boldsymbol{\lambda}, \mathbf{ap}\}\}$, where the controls have arity rules

$$\frac{t : m \to n}{\boldsymbol{\lambda}(t) : \varepsilon \to (m \Rightarrow n)} \qquad \mathbf{ap} : (m \Rightarrow n) \otimes m \to n$$

The reaction relation is generated from the rules

$$\sigma' : \quad (\boldsymbol{\lambda}(s) \otimes \mathbf{id}) \cdot (x)t \searrow t\{\boldsymbol{\lambda}(s)/\langle x \rangle\}$$

$$\beta : \quad (\boldsymbol{\lambda}(t) \otimes \mathbf{id}) \cdot \mathbf{ap} \searrow t$$

$$\eta : \quad \boldsymbol{\lambda}((\langle x \rangle \otimes \mathbf{id}) \cdot \mathbf{ap}) \searrow \langle x \rangle$$

where σ' can be viewed as explicit substitution.

3. Closed action calculi

We define a closed action calculus using a signature in a similar fashion to the definition of an action calculus. We shall see however that, given an action calculus

AC($\mathbb{K}$), the corresponding closed action calculus is generated from a signature $\mathbb{K}'$ constructed from $\mathbb{K}$.

Definition 3.1 (*Closed terms*). The set of *closed terms* over signature $\mathbb{K}=(P,\mathscr{K})$, denoted by CT($\mathbb{K}$), is constructed from the basic operators: *identity* $\mathbf{id}_m$, *composition* $\cdot$, *tensor* $\otimes$, *permutation* $\mathbf{p}$, *copy* Δ, *discard* $\boldsymbol{\omega}$, and the *controls* K. A closed term t is assigned an arity $t:m\rightarrow n$, for arities m and n, using the following rules:

$$\mathbf{id}_m : m \rightarrow m$$

$$\frac{s:k\rightarrow l \qquad t:l\rightarrow m}{s\cdot t:k\rightarrow m}$$

$$\frac{s:k\rightarrow m \qquad t:l\rightarrow n}{s\otimes t:k\otimes l\rightarrow m\otimes n}$$

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

$$\Delta_m : m \rightarrow m\otimes m$$

$$\boldsymbol{\omega}_m : m \rightarrow \varepsilon$$

$$\frac{t_1:m_1\rightarrow n_1 \quad \dots \quad t_r:m_r\rightarrow n_r}{K(t_1,\dots,t_r):m\rightarrow n}$$

where, in the *control term* $K(t_1,\dots,t_r)$, the arity of K is $((m_1,n_1),\dots,(m_r,n_r))\rightarrow(m,n)$. If the closed term contains no control terms, we call it a *closed wiring term*. As before, we shall omit the arity subscripts on the basic operators when they are apparent.

Definition 3.2 (*The theory CAC′*). The equational theory CAC′ is the set of equations upon terms generated by the action structure axioms A1–A6 from Definition 2.3, and the following:

B1: $\Delta_m\cdot(\boldsymbol{\omega}_m\otimes\mathbf{id})=\mathbf{id}$

B2: $\Delta_m\cdot\mathbf{p}_{m,m}=\Delta_m$

B3: $\mathbf{p}_{k,m}\cdot(s\otimes t)=(t\otimes s)\cdot\mathbf{p}_{l,n}, \quad s:m\rightarrow n, \quad t:k\rightarrow l$

B4: $\mathbf{p}_{m,n}\cdot\mathbf{p}_{n,m}=\mathbf{id}$

B5: $\mathbf{p}_{m\otimes n,k}=(\mathbf{id}\otimes\mathbf{p}_{n,k})\cdot(\mathbf{p}_{m,k}\otimes\mathbf{id})$

B6: $\boldsymbol{\omega}_{m\otimes n}=\boldsymbol{\omega}_m\otimes\boldsymbol{\omega}_n$

B7: $\Delta_{m\otimes n}=(\Delta_m\otimes\Delta_n)\cdot(\mathbf{id}\otimes\mathbf{p}_{m,n}\otimes\mathbf{id})$

B8: $\Delta_m\cdot(\Delta_m\otimes\mathbf{id})=\Delta_m\cdot(\mathbf{id}\otimes\Delta_m)$

Remark 3.3. We have chosen to define $\mathbf{id}_m$, $\boldsymbol{\omega}_m$, Δ_m and $\mathbf{p}_{m,n}$ for arbitrary arities and include the axioms B5–B7. Since arities can be uniquely factorized into primes, an alternative approach is to restrict the definitions to prime arities, remove B5–B7 and define the composite cases in terms of the prime cases and the other operators. This alternative approach is used in the definition of action calculi, since names are forced to have prime arity.

Remark 3.4. The equational theory CAC′ corresponds to the equational theory of a *ps-monoidal category*, recently studied by Corradini and Gadducci [7] in their work on graph rewriting.

Given an action calculus AC($\mathbb{K}$) where $\mathbb{K} = (P, \mathscr{K})$, we still need to identify the corresponding closed action calculus. We cannot just use the same signature $\mathbb{K}$ in the closed world. In the open world, we have free names occurring inside controls which are bound outside the controls. For example, using the action calculus PIC given in Example 2.7, we have the term

$$(x, y)\mathbf{in}(\langle x, y\rangle)$$

In order to express this term in the closed world, we declare a family of controls $\mathbf{in}_m$ for every $m \in M$. The purpose of the index m is to record the fact that terms inside the control $\mathbf{in}_m$ have been closed with respect to some sequence of names $\boldsymbol{x}$, where $|\boldsymbol{x}| = m$. For example, if we close the term $\mathbf{in}(\langle x, y\rangle)$ using sequence $[x^p, y^q]$ we obtain the closed term $\mathbf{in}_{p\otimes q}(\mathbf{id}_{p,q})$. If however we close the same term using sequence $[y^q, x^p]$, we obtain the closed term $\mathbf{in}_{q\otimes p}(\mathbf{p}_{q,p})$. Intuitively, these two closed terms should be connected since they have come from the same term $\mathbf{in}(\langle x, y\rangle)$. This intuition is captured by adding extra equalities to link controls with related indexing. For example, the controls $\mathbf{in}_{p\otimes q}$ and $\mathbf{in}_{q\otimes p}$ are connected by the equality

$$(\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot \mathbf{in}_{q\otimes p}(t) = \mathbf{in}_{p\otimes q}((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot t)$$

which results in $\mathbf{in}_{p\otimes q}(\mathbf{id})$ and $(\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot \mathbf{in}_{q\otimes p}(\mathbf{p}_{q,p})$ being equal. Using these extra equalities on the indexed controls, we obtain the tight correspondence we are seeking.

Definition 3.5. Given signature $\mathbb{K} = (P, \mathscr{K})$, the corresponding *closed signature* $\mathbb{K}'$ has the same set of primes P and the control set

$$\mathscr{K}' = \{K_l\colon\ l \in M \text{ and } K \in \mathscr{K}\}$$

such that, if the arity of K is $((m_1, n_1), \ldots, (m_r, n_r)) \to (m, n)$, then the arity of K_l is $((l \otimes m_1, n_1), \ldots, (l \otimes m_r, n_r)) \to (l \otimes m, n)$.

Definition 3.6. The *static* part of a closed action calculus CAC($\mathbb{K}'$) consists of the equivalence classes, called *closed actions* and denoted by $a, b, c, \ldots$, obtained by quotienting the closed terms in Definition 3.1, generated by the closed signature $\mathbb{K}'$, by the equational theory CAC generated by the axioms in Definition 3.2 plus the *control axioms*

D1: $\omega_n \otimes K_m(t_1, \dots, t_r) = K_{n\otimes m}(\omega_n \otimes t_1, \dots, \omega_n \otimes t_r)$

D2: $(\mathbf{id} \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}) \cdot K_{k\otimes n\otimes m}(t_1, \dots, t_r)$
$= K_{k\otimes m\otimes n}((\mathbf{id} \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}) \cdot t_1, \dots, (\mathbf{id} \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}) \cdot t_r)$

D3: $(\mathbf{id} \otimes \Delta_m \otimes \mathbf{id}) \cdot K_{k\otimes m\otimes m}(t_1, \dots, t_r)$
$= K_{k\otimes m}((\mathbf{id} \otimes \Delta_m \otimes \mathbf{id}) \cdot t_1, \dots, (\mathbf{id} \otimes \Delta_m \otimes \mathbf{id}) \cdot t_r)$

Before defining the reaction relation for CAC($\mathbb{K}'$), we first prove a property of the equational theory CAC which does not hold for AC. The impact of this result is that we must include an extra condition for the reaction relation of a closed action calculus. We require a few preliminary definitions. We write $p \in m$ if $m = p_1 \otimes \cdots \otimes p_r$ and $p = p_i$, and say that m is *contained* in n if $p \in m$ implies $p \in n$. A *closed context* is a term with a hole in it. Formally, it is described by the abstract grammar

$$C ::= [\_] \mid s \otimes C \mid C \otimes s \mid s \cdot C \mid C \cdot s \mid K(\dots, C, \dots)$$

where $s \in \mathrm{CT}(\mathbb{K})$. A *closed wiring context* W is a closed context which contains no controls. Given closed wiring term u and closed term s, we cannot always find a context C such that $C[u \otimes s] = s$, as the following proposition shows.

Proposition 3.7. *Given a closed wiring term $u : m \to n$ and closed term $s : k \to l$, where m is not contained in k, there is no context C such that $C[u \otimes s] = s$ in* CAC.

Proof. It is easy to show, for a closed wiring term $u : m \to n$, that n is contained in m using induction on the structure of u. Using this result, we can show for $W[s] : k \to l$, where $s : m \to n$ is a closed term and W is a closed wiring context, that m is contained in k. To prove the main result that $C[u \otimes s] \neq s$, it is enough to prove the result for all closed wiring contexts W, since if C contained a control the inequality would be automatic. We know that m is contained in the domain arity of $W[u \otimes s]$, and so the result holds. $\square$

Definition 3.8. The *dynamic* part of the closed action calculus CAC($\mathbb{K}'$) is a transitive relation between closed terms with the same arity which is preserved under tensor, composition and equality, and such that
1. $\mathbf{id}$ does not react;
2. for closed terms s and t, if $\mathbf{id} \otimes s \searrow \mathbf{id} \otimes t$ then $s \searrow t$.

By Proposition 3.7, we know that condition 2 cannot follow from the other closure properties. We shall see in Section 4 that the corresponding property for AC($\mathbb{K}$) is admissible, and so condition 2 is a necessary property of the reaction relation for closed action calculi. Conditions 1 and 2 of Definition 3.8 imply, for an arbitrary closed wiring term u, that u does not react and that $u \otimes s \searrow u \otimes t$ implies $s \searrow t$. This last fact is a corollary of the following proposition.

Proposition 3.9. *Given closed wiring term $u : k \to l$, there exist closed wiring terms s and t such that $s \cdot u \cdot t = \mathbf{id}_k$ in* CAC.

Proof (*Sketch*). First, we define the basic terms basic_p^n inductively on $n \geqslant 0$, by

$$\text{basic}_p^0 = \omega_p$$

$$\text{basic}_p^{n+1} = \Delta_p \cdot (\text{basic}_p^n \otimes \mathbf{id}_p)$$

For arbitrary closed wiring term u, we have $u = (u_1 \otimes \cdots \otimes u_n) \cdot \text{perm}$ in CAC, where the u_i are basic terms, and perm is a permutation term, defined by the grammar

$$t :: \mathbf{id} \mid \mathbf{p} \mid t \cdot t \mid t \otimes t$$

This fact is proved by induction on u. The only interesting case is when $u = v \cdot w$ for closed wiring terms v and w. By the induction hypothesis, $v = (v_1 \otimes \cdots \otimes v_l) \cdot \text{perm}_1$ and $w = (w_1 \otimes \cdots \otimes w_k) \cdot \text{perm}_2$. Using axiom B3 in Definition 3.2 for permuting tensors, we have

$$v \cdot w = (v_1 \otimes \cdots \otimes v_l) \cdot (w_1' \otimes \cdots \otimes w_k') \cdot \text{perm}'$$

where $w_1' \otimes \cdots \otimes w_k'$ is a permutation of $w_1 \otimes \cdots \otimes w_k$ and perm' is a permutation term. The proof follows by a secondary induction on l.

To prove the result, let $u = (u_1 \otimes \cdots \otimes u_n) \cdot \text{perm}$ for basic terms u_i and permutation term perm. By axiom B1 of Definition 3.2, observe that for all $n \geqslant 1$ there exists terms t^n such that $\text{basic}_p^n \cdot t^n = \mathbf{id}_p$, and that $\Delta_p \cdot (\text{basic}_p^0 \otimes \mathbf{id}_p) = \mathbf{id}_p$. Using this observation and the fact that the permutation terms have inverses, it is easy to construct s and t such that $s \cdot u \cdot t = \mathbf{id}_k$ in CAC. □

Remark 3.10. Notice that, unlike Proposition 2.6 for AC($\mathbb{K}$), we do not have the stronger result that there exists closed wiring terms s and t such that $s \cdot u \cdot t = \mathbf{id}_\varepsilon$ in CAC, since it would contradict Proposition 3.7.

In the next section, we shall see that a reaction relation for AC($\mathbb{K}$) determines a corresponding relation for CAC($\mathbb{K}'$), and vice versa.

4. Translations

We give the equality-preserving translations between action calculi and their corresponding closed action calculi, to provide a formal justification for the closed action calculi. These translations are also used to relate the reaction relations.

4.1. Action calculi to closed action calculi

We define a family of functions $[\![\_]\!]_{\boldsymbol{x}} : T_{\boldsymbol{x}}(\mathbb{K}) \to \text{CT}(\mathbb{K}')$ indexed by the sequence of names $\boldsymbol{x}$. We call these functions *closure functions*.

Definition 4.1. The *closure functions* $[\![\_]\!]_{\boldsymbol{x}} : T_{\boldsymbol{x}}(\mathbb{K}) \to \mathrm{CT}(\mathbb{K}')$, for each list of distinct names $\boldsymbol{x} = [x_1^{p_1}, \ldots, x_r^{p_r}]$, are defined inductively on the structure of terms in $T_{\boldsymbol{x}}(\mathbb{K})$ as follows:

$$[\![\mathbf{id}]\!]_{\boldsymbol{x}} = \omega_{|\boldsymbol{x}|} \otimes \mathbf{id}$$

$$[\![s \cdot t]\!]_{\boldsymbol{x}} = (\Delta_{|\boldsymbol{x}|} \otimes \mathbf{id}) \cdot (\mathbf{id}_{|\boldsymbol{x}|} \otimes [\![s]\!]_{\boldsymbol{x}}) \cdot [\![t]\!]_{\boldsymbol{x}}$$

$$[\![s \otimes t]\!]_{\boldsymbol{x}} = (\Delta_{\boldsymbol{x}} \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \mathbf{p}_{|\boldsymbol{x}|,k} \otimes \mathbf{id}) \cdot ([\![s]\!]_{\boldsymbol{x}} \otimes [\![t]\!]_{\boldsymbol{x}})$$

$$[\![\mathbf{ab}_x(t)]\!]_{\boldsymbol{x}} = (\mathbf{id} \otimes \Delta_p \otimes \mathbf{id}) \cdot (\mathbf{p}_{|\boldsymbol{x}|,p} \otimes \mathbf{id} \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes [\![t\{y/x\}]\!]_{\boldsymbol{x},y}), \quad y^p \notin \{\boldsymbol{x}\}$$

$$[\![\langle x \rangle]\!]_{\boldsymbol{x}} = \omega_{p_i \otimes \cdots \otimes p_{i-1}} \otimes \mathbf{id}_{p_i} \otimes \omega_{p_{i+1} \otimes \cdots \otimes p_r}, \quad x = x_i,\ i \in \{1, \ldots, r\}$$

$$[\![\omega_p]\!]_{\boldsymbol{x}} = \omega_{|\boldsymbol{x}|} \otimes \omega_p$$

$$[\![K(t_1, \ldots, t_r)]\!]_{\boldsymbol{x}} = K_{|\boldsymbol{x}|}([\![t_1]\!]_{\boldsymbol{x}}, \ldots, [\![t_r]\!]_{\boldsymbol{x}})$$

Whenever we write $[\![\_]\!]_{\boldsymbol{x}}$, we assume that $\boldsymbol{x}$ is a list of distinct names. We shall often wish to distinguish a particular name in such a list. We therefore write $\boldsymbol{x}, y, \boldsymbol{z}$ to denote a sequence of distinct names with the name y distinguished. In the definition of $[\![\_]\!]_{\boldsymbol{x}}$, the abstraction case is perhaps the most confusing. The idea of viewing the behaviour of $[\![\_]\!]_{\boldsymbol{x}}$ as the closure of a term using $\boldsymbol{x}$ becomes clearer when we use the alternative form of abstraction $(x)t$, as the following proposition shows.

Proposition 4.2. *We have the equality* $[\![(x)t]\!]_{\boldsymbol{x}} = [\![t\{\langle y \rangle / \langle x \rangle\}]\!]_{\boldsymbol{x},y}$ *in* CAC *for some* $y \notin \{\boldsymbol{x}\}$ *with the same arity as* x.

Proof. The proof follows by straightforward equational reasoning. □

Also, notice that $[\![\mathbf{ab}_x(t)]\!]_{\boldsymbol{x}}$ and $[\![(x)t]\!]_{\boldsymbol{x}}$ are defined using a chosen $y \notin \{\boldsymbol{x}\}$. The next lemma shows that this choice of y is not important.

Lemma 4.3. *We have the equality* $[\![t]\!]_{\boldsymbol{x},u,\boldsymbol{y}} = [\![t\{v/u\}]\!]_{\boldsymbol{x},v,\boldsymbol{y}}$ *in* CAC, *if* u *and* v *have the same arity.*

Proof. The proof is by easy induction on the structure of t. □

The following three lemmas illustrate the connection between the closure functions $[\![\_]\!]_{\boldsymbol{x}}$ and $[\![\_]\!]_{\boldsymbol{y}}$, when $\{\boldsymbol{x}\} \subseteq \{\boldsymbol{y}\}$. They are proved by induction on the structure of term t. In each proof, the interesting case is when t has the form $K(t_1, \ldots, t_r)$, since this case shows that the proofs rely directly on the control axioms D1–D3 introduced in Section 3. The details are not difficult and can be found in [8].

Lemma 4.4. 1. $[\![t]\!]_{y,\boldsymbol{x}} = \omega_p \otimes [\![t]\!]_{\boldsymbol{x}}$ *in* CAC, *when* $y^p \notin fn(t)$.
2. $[\![t]\!]_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} = (\mathbf{id} \otimes \mathbf{p}_{|\boldsymbol{y}|,|\boldsymbol{z}|} \otimes \mathbf{id}) \cdot [\![t]\!]_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{y}}$ *in* CAC.
3. $(\mathbf{id} \otimes \Delta_p \otimes \mathbf{id} \otimes \mathbf{id}) \cdot [\![t]\!]_{\boldsymbol{x},u,v,\boldsymbol{y}} = [\![t\{u/v\}]\!]_{\boldsymbol{x},u,\boldsymbol{y}}$ *in* CAC *if* u *and* v *have the same arity.*

The results above illustrate that the closure functions behave as expected. Using these results, we prove that these functions preserve equality. The proof uses the technical device of working with judgements of the form $\{\boldsymbol{x}\} \vdash s = t$, which denotes that $s = t$ in AC and that $fn(s,t) \subseteq \{\boldsymbol{x}\}$, in order to give precise control of names in actions. The details are given in the appendix.

Theorem 4.5. *Given* $s, t, \in T_{\boldsymbol{x}}(\mathbb{K})$, *if* $s = t$ *in the equational theory* AC *then* $[\![s]\!]_{\boldsymbol{x}} = [\![t]\!]_{\boldsymbol{x}}$ *in* CAC.

4.2. Closed action calculi to action calculi

There is also an equality-preserving translation from the closed action calculus to the corresponding action calculus. This translation, together with the closure functions defined in the previous section, yields a tight correspondence between the static parts of AC($\mathbb{K}$) and CAC($\mathbb{K}'$). Recall that the indexing on the controls is used to record the information that the terms inside the controls have been closed using a sequence of names of the appropriate arity. We use this information during the translation in an essential way to incorporate free names inside the controls.

Definition 4.6. The translation $\langle \_ \rangle : \mathrm{CT}(\mathbb{K}') \to T_{\emptyset}(\mathbb{K})$ is defined inductively on the structure of closed terms as follows:

$$\langle \mathbf{id} \rangle = \mathbf{id}$$

$$\langle s \cdot t \rangle = \langle s \rangle \cdot \langle t \rangle$$

$$\langle s \otimes t \rangle = \langle s \rangle \otimes \langle t \rangle$$

$$\langle \Delta_m \rangle = (\boldsymbol{x})\langle \boldsymbol{x}, \boldsymbol{x} \rangle, \quad |\boldsymbol{x}| = m$$

$$\langle \mathbf{p}_{m,n} \rangle = \mathbf{p}_{m,n}, \quad |\boldsymbol{x}| = m, |\boldsymbol{y}| = n$$

$$\langle \omega_m \rangle = \omega_m$$

$$\langle K_m(t_1, \ldots, t_r) \rangle = (\boldsymbol{x})K((\langle \boldsymbol{x} \rangle \otimes \mathbf{id}) \cdot \langle t_1 \rangle, \ldots, (\langle \boldsymbol{x} \rangle \otimes \mathbf{id}) \cdot \langle t_r \rangle), \quad |\boldsymbol{x}| = m$$

The proof that $\langle \_ \rangle$ preserves equality is easier than the corresponding proof for $[\![\_]\!]$, and again involves working with judgements of the form $\{\boldsymbol{x}\} \vdash s = t$ where $fn(s,t) \subseteq \{\boldsymbol{x}\}$. The proof is given in the appendix.

Theorem 4.7. *Given closed terms* $s, t \in \mathrm{CT}(\mathbb{K})$, *if* $s = t$ *in the equational theory* CAC *then* $\langle s \rangle = \langle t \rangle$ *in* AC.

In the appendix we also give the proof of the theorem below, which states how the translations $[\![\_]\!]_{\boldsymbol{x}}$ and $\langle \_ \rangle$ are connected.

Theorem 4.8. 1. *Given* $t \in T_{\boldsymbol{x}}(\mathbb{K})$, *we have* $\langle [\![t]\!]_{\boldsymbol{x}} \rangle = (\boldsymbol{x})t$ *in* AC.
2. *Given* $s \in \mathrm{CT}(\mathbb{K})$, *we have* $[\![\langle s \rangle]\!]_{\emptyset} = s$ *in* CAC.

4.3. Relating the reaction relations

We relate the reaction relations for action calculi and their corresponding closed action calculi. In particular, we give general results which show that a reaction relation for an action calculus generates a reaction relation in the corresponding closed world, and vice versa. We look at the examples PIC and LAMB given in Section 2, whose reaction relations are generated by a finite set of rules. The LAMB example shows that we cannot generate the reaction relations in the closed world by simply translating the rules from the open world. We shall see in Section 5 that such a translation is possible when we add free names to closed action calculi.

Definition 4.9. Let $\searrow$ be a relation for AC($\mathbb{K}$). Define a binary relation R on closed terms in CT($\mathbb{K}'$) by $s\,R\,t$ if and only if $\langle s\rangle\searrow\langle t\rangle$ in AC($\mathbb{K}$).

We show that the relation R is a reaction relation for CAC($\mathbb{K}'$). The interesting part of the proof is to show that condition 2 in Definition 3.8 holds. It relies on a property of the equational theory AC that, for wiring term u and term s, there is a wiring context W such that $W[u\otimes s]=s$, which is independent of the structure of s. This property follows directly from Proposition 2.6.

Proposition 4.10. *The relation R given in Definition* 4.9 *is a reaction relation for* CAC($\mathbb{K}'$).

Proof. It is easy to prove that R is transitive, and is closed under composition, tensor and the equational theory. It is also easy to prove, given closed wiring term u, that uRs does not occur for any closed term s using the fact that $\langle u\rangle$ is a wiring term. To prove condition 2 of Definition 3.8, assume that $(u\otimes s)R(u\otimes t)$, and hence that $\langle u\rangle\otimes\langle s\rangle\searrow\langle u\rangle\otimes\langle t\rangle$. Since $\langle u\rangle$ is a wiring term, by Proposition 2.6 we have wiring terms w_1 and w_2 such that $w_1\cdot u\cdot w_2=\mathbf{id}_\varepsilon$. We therefore have

$$(w_1\otimes\mathbf{id}_m)\cdot(\langle u\rangle\otimes\langle s\rangle)\cdot(w_2\otimes\mathbf{id}_n)=\langle s\rangle\searrow(w_1\otimes\mathbf{id}_m)\cdot(\langle u\rangle\otimes\langle t\rangle)\cdot(w_2\otimes\mathbf{id}_n)=\langle t\rangle$$

where $s:m\to n$, and hence $s\,R\,t$. □

We call R in Definition 4.9 the reaction relation for CAC($\mathbb{K}'$) generated by $\searrow$.

We can also generate a reaction relation for AC($\mathbb{K}$) from a reaction relation in the corresponding closed world.

Definition 4.11. Let $\searrow$ be a reaction relation for CAC($\mathbb{K}'$). Define a binary relation S on terms in $T(\mathbb{K})$ by $s\,S\,t$ if and only if $[\![s]\!]_{\boldsymbol{x}}\searrow[\![t]\!]_{\boldsymbol{x}}$ in CAC($\mathbb{K}'$) whenever $fn(s)\subseteq\{\boldsymbol{x}\}$.

Remark 4.12. It is easy to prove that $[\![s]\!]_{\boldsymbol{x}}=u\otimes[\![s]\!]_{fn(s)}$, where u is a wiring term. We can therefore just consider the case when $\{\boldsymbol{x}\}=fn(s)$ in the above definition.

Proposition 4.13. *The relation S in Definition* 4.11 *is a reaction relation for* AC($\mathbb{K}$).

Proof. To prove transitivity of S, assume that $s\,S\,t$ and $t\,S\,u$, and let $\{\boldsymbol{x}\} = fn(s,u)$ and $\{\boldsymbol{y}\} = fn(t)\backslash\{\boldsymbol{x}\}$. We have $[\![s]\!]_{\boldsymbol{y},\boldsymbol{x}} \searrow [\![t]\!]_{\boldsymbol{y},\boldsymbol{x}} \searrow [\![u]\!]_{\boldsymbol{y},\boldsymbol{x}}$. By the transitivity of $\searrow$ and Lemma 4.4, we have

$$\boldsymbol{\omega}_{|\boldsymbol{y}|} \otimes [\![s]\!]_{\boldsymbol{x}} \searrow \boldsymbol{\omega}_{|\boldsymbol{y}|} \otimes [\![u]\!]_{\boldsymbol{x}}$$

Using condition 2 of Definition 3.8, we have $[\![s]\!]_{\boldsymbol{x}} \searrow [\![u]\!]_{\boldsymbol{x}}$, and hence R is transitive. The relation S is closed under tensor, composition and abstraction. It is easy to prove that, for wiring term u, the relation $u\,S\,s$ does not occur for any s. To prove closure under equality, assume that $s = s'$ and $t = t'$ in AC, and $s'\,S\,t'$, and let $\{\boldsymbol{x}\} = fn(s,t)$ and $\{\boldsymbol{y}\} = fn(s',t')\backslash\{\boldsymbol{x}\}$. By Theorem 4.5 and the definition of R, we have $[\![s]\!]_{\boldsymbol{y},\boldsymbol{x}} = [\![s']\!]_{\boldsymbol{y},\boldsymbol{x}}$ and $[\![t]\!]_{\boldsymbol{y},\boldsymbol{x}} = [\![t']\!]_{\boldsymbol{y},\boldsymbol{x}}$ in CAC, and $[\![s]\!]_{\boldsymbol{y},\boldsymbol{x}} \searrow [\![t]\!]_{\boldsymbol{y},\boldsymbol{x}}$. By Lemma 4.4, it follows that $\boldsymbol{\omega}_{|\boldsymbol{y}|} \otimes [\![s]\!]_{\boldsymbol{x}} \searrow \boldsymbol{\omega}_{|\boldsymbol{y}|} \otimes [\![t]\!]_{\boldsymbol{x}}$. Using condition 2 of Definition 3.8, we have $[\![s]\!]_{\boldsymbol{x}} \searrow [\![t]\!]_{\boldsymbol{x}}$, and hence $s\,S\,t$. □

We call S in Definition 4.11 the reaction relation for AC($\mathbb{K}$) generated by $\searrow$.

Proposition 4.14. *Let* $\searrow$ *be a reaction relation for* AC($\mathbb{K}$), *let* R *be the reaction relation for* CAC($\mathbb{K}'$) *generated from* $\searrow$, *and let* S *be the reaction relation for* AC($\mathbb{K}$) *generated from* R. *The relations* S *and* $\searrow$ *are equal. The analogous result holds if we start from a reaction relation for* CAC($\mathbb{K}$).

Proof. The proof follows easily from Theorem 4.8. □

A reaction relation is typically generated from a set of rules. We would like a simple connection between such rules for action calculi and closed action calculi, but the LAMB example in Section 2 shows that this is not straightforward. The PIC example is a simple case, in that one reaction rule in the open world corresponds to a reaction rule in the closed world. The LAMB example requires more care, since the number of reaction rules in the open and closed world are not the same.

Example 4.15. The closed action calculus corresponding to PIC in Example 2.7 has signature $(\{1\}, \{\mathbf{out}_m, \mathbf{in}_m, \mathbf{new}_m\colon\ m \in M\})$, and a reaction relation generated by the rule

$$(\Delta_{k\otimes 1} \otimes \mathbf{id}_1) \cdot (\mathbf{id} \otimes \mathbf{p}_{k\otimes 1,1}) \cdot (\mathbf{out}_k \otimes \mathbf{in}_k(t)) \searrow (\mathbf{id} \otimes \boldsymbol{\omega}_1 \otimes \mathbf{id}) \cdot t$$

Example 4.16. The closed action calculus corresponding to LAMB in Example 2.8 has signature $(P^{\Rightarrow}, \{\boldsymbol{\lambda}_m, \mathbf{ap}_m\colon\ m \in M^{\Rightarrow}\})$, where $P^{\Rightarrow}$ and $M^{\Rightarrow}$ are given in Example 2.8, the arity rules for $\boldsymbol{\lambda}_k$ and $\mathbf{ap}_k$ are

$$\frac{t : k \otimes m \to k \otimes n}{\boldsymbol{\lambda}_k(t) : k \to (m \Rightarrow n)} \qquad \mathbf{ap}_k : k \otimes (m \Rightarrow n) \otimes m \to n$$

and the reaction relation is generated by the rules

$$\boldsymbol{\lambda}_k(t)\cdot \Delta_{m\Rightarrow n} \searrow \Delta_k \cdot (\boldsymbol{\lambda}_k(t)\otimes \boldsymbol{\lambda}_k(t))$$

$$\boldsymbol{\lambda}_k(t)\cdot \boldsymbol{\omega}_{m\Rightarrow n} \searrow \boldsymbol{\omega}_k$$

$$\Delta_k \cdot (\mathbf{id}_k \otimes \boldsymbol{\lambda}_k(s))\cdot \boldsymbol{\lambda}_{k\otimes(m\Rightarrow n)}(t) \searrow \boldsymbol{\lambda}_k((\Delta_k \otimes \mathbf{id})\cdot(\mathbf{id}\otimes \boldsymbol{\lambda}_k(s)\otimes \mathbf{id})\cdot t)$$

$$(\Delta_k \otimes \mathbf{id})\cdot(\mathbf{id}\otimes \boldsymbol{\lambda}_k(s)\otimes \mathbf{id})\cdot \mathbf{ap}_k \searrow s$$

$$\boldsymbol{\lambda}_k((\Delta \otimes \mathbf{id})\cdot \mathbf{ap}_k) \searrow \mathbf{id}$$

The last two rules correspond to the β- and η-rule, respectively, in Example 2.8. The first three rules provide the individual cases necessary to mimic in the closed world the substitution of a term t for a name in the open world. In general, we can have an arbitrary number of reaction rules in the closed world corresponding to one reaction rule in the open world. For example, if we extend the control set of LAMB by an arbitrary control set $\mathscr{K}$, then the corresponding closed action calculus would contain a reaction rule

$$(\Delta_k \otimes \mathbf{id})\cdot(\mathbf{id}\otimes \boldsymbol{\lambda}_k(s)\otimes \mathbf{id})\cdot K_{k\otimes(m\Rightarrow n)}(t_1,\dots,t_r) \searrow$$

$$K_k((\Delta_k \otimes \mathbf{id})\cdot(\mathbf{id}\otimes \boldsymbol{\lambda}_k(s)\otimes \mathbf{id})\cdot t_1,\dots,(\Delta_k \otimes \mathbf{id})\cdot(\mathbf{id}\otimes \boldsymbol{\lambda}_k(s)\otimes \mathbf{id})\cdot t_r)$$

for each control $K \in \mathscr{K}$.

5. Closed action calculi with names

We extend the closed action calculi with free names, and show that abstraction can be defined as a derived operator. In independent work, Pavlovic has given similar results to those presented in this section [23].

Definition 5.1 (*Extended terms*). The set of *extended terms* over closed signature $\mathbb{K}'$ and name set X, denoted by $\mathrm{CT}_X(\mathbb{K}')$, is generated from the rules in Definition 3.1, plus the rule

$$\langle x\rangle : \varepsilon \to p, \quad x^p \in X$$

The name x is free in any extended term containing $\langle x\rangle$. In the action calculi setting, we have axiom σ which allows the movement of names through the sequential composition. We mimic this movement of names, by incorporating three natural axioms which allow names to be copied, discarded and to move inside controls.

Definition 5.2 (*The extended theory* CAC_X). The equational theory CAC_X is the set of equations upon the extended terms generated by the axioms in Definition 3.2, plus the axioms

$$\langle x\rangle\cdot\Delta=\langle x\rangle\otimes\langle x\rangle$$

$$\langle x\rangle\cdot\boldsymbol{\omega}=\mathbf{id}_\varepsilon$$

$$(\langle x^p\rangle\otimes\mathbf{id})\cdot K_{p\otimes m}(t_1,\dots,t_n)=K_m((\langle x^p\rangle\otimes\mathbf{id})\cdot t_1,\dots,(\langle x^p\rangle\otimes\mathbf{id})\cdot t_n)$$

The definition of a reaction relation for $\mathrm{CAC}_X(\mathbb{K}')$ is the same as the one given in Definition 3.8.

Using the free names, we can derive an abstraction for the extended terms. This definition is similar to the standard way of defining abstraction in combinatory logic, which leads to the well-known connection with the λ-calculus (see for example [13]).

Definition 5.3 (*Abstraction*). Given the extended term $t:m\to n$ in $\mathrm{CAC}_X(\mathbb{K}')$ and name $x^p\in X$, the *abstraction* $[x]t:p\otimes m\to n$ is defined by induction on the structure of t:

$$[x]\langle x\rangle=\mathbf{id}_p$$

$$[x]\langle y\rangle=\boldsymbol{\omega}_p\otimes\langle y\rangle$$

$$[x]\Delta=\boldsymbol{\omega}_p\otimes\Delta$$

$$[x]\mathbf{id}=\boldsymbol{\omega}_p\otimes\mathbf{id}$$

$$[x]\boldsymbol{\omega}=\boldsymbol{\omega}_p\otimes\boldsymbol{\omega}$$

$$[x]\mathbf{p}=\boldsymbol{\omega}_p\otimes\mathbf{p}$$

$$[x](t_1\cdot t_2)=(\Delta_p\otimes\mathbf{id})\cdot(\mathbf{id}\otimes[x]t_1)\cdot[x]t_2$$

$$[x](t_1\otimes t_2)=(\Delta_p\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\mathbf{p}_{p,m}\otimes\mathbf{id})\cdot([x]t_1\otimes[x]t_2)$$

$$[x](K_m(t_1,\dots,t_n))=K_{p\otimes m}([x]t_1,\dots,[x]t_r)$$

The following results show that we have equalities corresponding to the axioms γ,σ and δ from Definition 2.3, and that equality is preserved by this derived abstraction.

Lemma 5.4. 1. $[x]t=\boldsymbol{\omega}\otimes t$ *when* $x\notin fn(t)$;
2. $(\langle x\rangle\otimes\mathbf{id})\cdot[y]t=t\{\langle x\rangle/\langle y\rangle\}$;
3. $[x]((\langle x\rangle\otimes\mathbf{id})\cdot t)=t,\quad x\notin fn(t)$;
4. $s=t$ *in* CAC_X *implies* $[x]s=[x]t$ *in* CAC_X.

Proof. The details are straightforward. In part 4, the following easy technical result was helpful: if $x\notin fn(s)$, then
1. $[x]s\cdot t=(\mathbf{id}_p\otimes s)\cdot[x]t$;
2. $[x](t\cdot s)=[x]t\cdot s$;
3. $[x](s\otimes t)=(\mathbf{p}_{p,m}\otimes\mathbf{id})\cdot(s\otimes[x]t)$;
4. $[x](t\otimes s)=[x]t\otimes s$.

Now that we have defined the abstraction $[x]s$, the translations between $\mathrm{AC}_X(\mathbb{K})$ and $\mathrm{CAC}_X(\mathbb{K}')$ are simple. Using Lemma 5.4, it is straightforward to check that these translations are equality-preserving and inverse to each other.

Definition 5.5 (*Translation*). The translation $(\_)' : T_X(\mathbb{K}) \to \mathrm{CT}_X(\mathbb{K})$ is defined inductively by

$$(\mathbf{id})' = \mathbf{id}$$

$$(\mathbf{p}_{m,n})' = \mathbf{p}_{m,n}$$

$$(\langle x\rangle)' = \langle x\rangle$$

$$(s \cdot t)' = s' \cdot t'$$

$$(s \otimes t)' = s' \otimes t'$$

$$((x)t)' = [x]t'$$

$$(K(t_1, \ldots, t_r))' = K_0(t_1', \ldots, t_r')$$

Proposition 5.6. 1. *If* $t \in T_X(\mathbb{K})$ *with arity* $m \to n$, *then* $t' \in \mathrm{CT}_X(\mathbb{K}')$ *with the same arity, and* $fn(t) = fn(t')$.

2. $s = t$ *in* AC *implies* $s' = t'$ *in* CAC_X.

Definition 5.7 (*Translation*). The translation $(\_)^\circ : \mathrm{CT}_X(\mathbb{K}') \to T_X(\mathbb{K})$ is defined inductively on the structure of closed terms by

$$(\mathbf{id}_m)^\circ = \mathbf{id}_m$$

$$(\mathbf{p}_{m,n})^\circ = \mathbf{p}_{m,n}$$

$$(\langle x\rangle)^\circ = \langle x\rangle$$

$$(s \cdot t)^\circ = s^\circ \cdot t^\circ$$

$$(s \otimes t)^\circ = s^\circ \otimes t^\circ$$

$$(\Delta_m)^\circ = (\boldsymbol{x})(\langle \boldsymbol{x}\rangle \otimes \langle \boldsymbol{x}\rangle), \quad |\boldsymbol{x}| = m$$

$$(\omega_m)^\circ = (\boldsymbol{x})\mathbf{id}_\varepsilon, \quad |\boldsymbol{x}| = m$$

$$(K_m(t_1, \ldots, t_r))^\circ = (\boldsymbol{x})K((\langle \boldsymbol{x}\rangle \otimes \mathbf{id}) \cdot t_1^\circ, \ldots, (\langle \boldsymbol{x}\rangle \otimes \mathbf{id}) \cdot t_r^\circ),$$
$$|\boldsymbol{x}| = m, \;\; \boldsymbol{x} \notin fn(t_1, \ldots, t_n).$$

Proposition 5.8. 1. *If* $t \in \mathrm{CT}_X(\mathbb{K})$ *with arity* $m \to n$, *then* $t^\circ \in T_X(\mathbb{K})$ *with the same arity, and* $fn(t) = fn(t^\circ)$.

2. $s = t$ *in* CAC_X *implies* $s^\circ = t^\circ$ *in* AC.

Proposition 5.9. 1. *Given* $t \in T_X(\mathbb{K})$, *we have* $(t')^\circ = t$ *in* AC.

2. *Given* $t \in \mathrm{CT}_X(\mathbb{K}')$, *we have* $(t^\circ)' = t$ *in* CAC_X.

Proof. The proof of 1 depends on showing that $([x]t)^\circ = (x)t^\circ$ in AC. The proof of 2 depends on Lemma 5.4. $\square$

Since we have a derived abstraction in $\mathrm{CAC}_X(\mathbb{K}')$, the rules for generating the reaction relation for $\mathrm{AC}(\mathbb{K})$ simply translate to rules for generating the corresponding reaction relation for $\mathrm{CAC}_X(\mathbb{K}')$. We illustrate this with the LAMB example.

Example 5.10. The closed action calculus extended with names, and corresponding to LAMB in Example 2.8, has the same signature as in the closed action calculus in Example 4.15, and a reaction relation generated by the rules

$$(\lambda_0(s) \otimes \mathbf{id}) \cdot [x]t \searrow t\{\lambda_0(s)/\langle x\rangle\}$$

$$(\lambda_0(t) \otimes \mathbf{id}) \cdot \mathbf{ap}_0 \searrow t$$

$$\lambda_0((\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{ap}_0) \searrow \langle x\rangle$$

These rules are the $(\_)'$-translations of the original rules for LAMB.

6. Extensions of action calculi

Milner has introduced two extensions of action calculi: the *higher-order action calculi* [19], which allow the substitution of actions as well as names for names, and the *reflexive action calculi* [18], which in the presence of higher-order features gives recursion. We extend closed action calculi to include higher-order and reflexive features, and obtain results analogous to those given in Section 4.

6.1. Higher-order action calculi

Recall the action calculus LAMB given in Example 2.8. The controls $\boldsymbol{\lambda}$ and **ap**, and their accompanying reaction rules, describe a uniform way of packing up a term t using $\boldsymbol{\lambda}$, substituting the resulting term for names, and unpacking the term using **ap**. These controls therefore describe a way of moving terms around, which is a natural extension to the basic structure of action calculi. Higher-order action calculi capture this extension explicitly, by viewing $\boldsymbol{\lambda}$ and **ap** as basic operators rather than controls, and by extending the equational theory by equalities corresponding to the σ', β and η axioms.

Definition 6.1 (*Higher-order action calculi*). The *higher-order action calculus* $\mathrm{HAC}(\mathbb{K})$ is given by extending the definition of action calculi as follows:
1. the sets of higher-order primes and arities are the same as in Definition 2.8;
2. the set of higher-order terms $\mathrm{HT}(\mathbb{K})$ is generated by the rules in Definition 2.1 using a name set X ranging over $P^{\Rightarrow}$, plus the rules

$$\frac{t : m \to n}{\boldsymbol{\lambda}(t) : \varepsilon \to (m \Rightarrow n)} \qquad \mathbf{ap} : (m \Rightarrow n) \otimes m \to n$$

3. the equational theory HAC is generated from the axioms in Definition 2.3, plus the axioms

$$(\lambda(s)\otimes \mathbf{id})\cdot(x)t = t\{\boldsymbol{\lambda}(s)/\langle x\rangle\}$$
$$(\lambda(s)\otimes \mathbf{id}_m)\cdot \mathbf{ap} = s$$
$$\lambda((\langle x\rangle\otimes \mathbf{id}_m)\cdot \mathbf{ap}) = \langle x\rangle$$

4. a reaction relation for HAC($\mathbb{K}$) is the same as that given in Definition 2.5.

The closed higher-order action calculi similarly arise from the closed version of LAMB given in Example 4.15. The basic operators are extended by the family of operators $\boldsymbol{\lambda}_k$ and $\mathbf{ap}_k$, and the equational theory is extended by equalities corresponding to the rewrite rules.

Definition 6.2. The *closed higher-order action calculus* CHAC($\mathbb{K}'$) is given by extending the definition of the closed action calculus CAC($\mathbb{K}'$) as follows:
1. the sets of higher-order primes and arities are the same as in Definition 2.8;
2. the set of closed higher-order terms CHT($\mathbb{K}'$) is generated by the rules in Definition 2.1, together with rules for $\boldsymbol{\lambda}_k$ and $\mathbf{ap}_k$ given in Example 4.16;
3. the equational theory CHAC is generated from the axioms in Definition 2.3, the higher-order axioms

$$\boldsymbol{\lambda}_k(t)\cdot \Delta_{m\Rightarrow n} = \Delta_k\cdot(\boldsymbol{\lambda}_k(t)\otimes\boldsymbol{\lambda}_k(t))$$
$$\boldsymbol{\lambda}_k(t)\cdot \omega_{m\Rightarrow n} = \omega_k$$
$$\Delta_k\cdot(\mathbf{id}_k\otimes\boldsymbol{\lambda}_k(s))\cdot\boldsymbol{\lambda}_{k\otimes(m\Rightarrow n)}(t) = \boldsymbol{\lambda}_k((\Delta_k\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\boldsymbol{\lambda}_k(s)\otimes\mathbf{id})\cdot t)$$
$$(\Delta_k\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\boldsymbol{\lambda}_k(s)\otimes\mathbf{id})\cdot\mathbf{ap}_k = s$$
$$\boldsymbol{\lambda}_{m\Rightarrow n}((\Delta\otimes\mathbf{id})\cdot\mathbf{ap}_{m\Rightarrow n}) = \mathbf{id}$$

the control axioms D1–D3 from Definition 3.6 and an extra control axiom

$$(\Delta_k\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\boldsymbol{\lambda}_k(s)\otimes\mathbf{id})\cdot K_{k\otimes(m\Rightarrow n)}(t_1,\dots,t_r)$$
$$= K_k((\Delta_k\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\lambda_k(s)\otimes\mathbf{id})\cdot t_1,\dots,(\Delta_k\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\boldsymbol{\lambda}_k(s)\otimes\mathbf{id})\cdot t_r)$$

for each control K_k;
4. a reaction relation on closed higher-order terms is defined similarly to the reaction relation for CAC($\mathbb{K}'$) in Definition 3.6.

The functions $[\![\_]\!]_{\boldsymbol{x}}: T_{\boldsymbol{x}}(\mathbb{K})\to \mathrm{CT}(\mathbb{K}')$ and $\langle\_\rangle:\mathrm{CT}(\mathbb{K}')\to T(\mathbb{K})$ in Definitions 4.1 and 4.6 are easily extended to account for the higher-order setting:

$$[\![\boldsymbol{\lambda}(t)]\!]_{\boldsymbol{x}} = \boldsymbol{\lambda}_{|\boldsymbol{x}|}([\![t]\!]_{\boldsymbol{x}})$$
$$[\![\mathbf{ap}]\!]_{\boldsymbol{x}} = \mathbf{ap}_{|\boldsymbol{x}|}$$
$$\langle\boldsymbol{\lambda}_k(t)\rangle = (\boldsymbol{x})\boldsymbol{\lambda}((\langle\boldsymbol{x}\rangle\otimes\mathbf{id})\cdot(\langle t\rangle), \quad |\boldsymbol{x}| = k$$
$$\langle\mathbf{ap}_k\rangle = (\boldsymbol{x})\mathbf{ap}, \quad |\boldsymbol{x}| = k$$

We have analogous results to those given in Theorems 4.5, 4.7 and 4.8.

6.2. Reflexive action calculi

The reflexive action calculi [18] are action calculi extended by an additional operator $\uparrow_p$, called the *reflexion operator*, which constructs a term $\uparrow_p t : m \to n$ from $t : p \otimes m \to p \otimes n$. This operator provides a notion of feedback and, together with the higher-order features described in the previous section, is enough to capture recursion. Mifsud [16] and Hasegawa [12] have shown that it corresponds to the trace operator of Joyal et al. [15].

Definition 6.3. The *reflexive action calculus* RAC($\mathbb{K}$) is given by extending the definition of the action calculus AC($\mathbb{K}$) as follows:
1. the set of reflexive terms RT($\mathbb{K}$) is generated by the rules in Definition 2.1, plus a rule for the *reflexion* operator
$$\frac{t : p \otimes m \to p \otimes n}{\uparrow_p(t) : m \to n}$$
2. the equational theory RAC is generated from the axioms in Definition 2.3, plus the reflexive axioms
$$\mathbf{id} = \ \uparrow_p \mathbf{id}_p$$
$$\mathbf{id} = \ \uparrow_p \mathbf{p}_{p,p}$$
$$\uparrow_p t \otimes \mathbf{id} = \ \uparrow_p (t \otimes \mathbf{id})$$
$$\uparrow_p s \cdot t = \ \uparrow_p (s \cdot (\mathbf{id}_p \otimes t))$$
$$s \cdot \ \uparrow_p t = \ \uparrow_p ((\mathbf{id}_p \otimes s) \cdot t)$$
$$\uparrow_q \uparrow_p t = \ \uparrow_p \uparrow_q ((\mathbf{p}_{q,p} \otimes \mathbf{id}) \cdot t \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}))$$
3. a reaction relation on reflexive terms defined similarly to the reaction relation for AC($\mathbb{K}$) in Definition 2.5.

Remark 6.4. The original definition of reflexive action calculi [18] also has the axiom
$$(x) \uparrow_p t = \ \uparrow_p ((\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot (x)t)$$
Hasegawa [12] observed that this axiom follows from the other axioms.

Remark 6.5. In [18], the first axiom is not included in the original definition of reflexive action calculi, although it is discussed as a natural extension. We chose to include it here, although our result does not depend on it. One reason for our choice is that this axiom is necessary to prove the analogous result to Proposition 2.6.

It is easy to extend the closed action calculi to account for reflexion, using the same reflexive operator and axioms.

Definition 6.6. The *closed reflexive action calculus* CRAC($\mathbb{K}'$) is given by extending the definition of the closed action calculus CAC($\mathbb{K}'$) as follows:
1. the set of closed reflexive terms CRT($\mathbb{K}'$) is generated by the rules in Definition 2.1, plus the rule for the reflexive operator given Definition 6.3;
2. the equational theory CRAC is generated from the axioms in Definition 2.3, together with the reflexive axioms in Definition 6.3 and the control axioms D1–D3 in Definition 3.6;
3. a reaction relation on closed reflexive terms defined similarly to the reaction relation for CAC($\mathbb{K}'$) in Definition 3.6.

The functions $[\![\_]\!]_{\boldsymbol{x}} : T_{\boldsymbol{x}}(\mathbb{K}) \to \mathrm{CT}(\mathbb{K}')$ and $\langle \_ \rangle : \mathrm{CT}(\mathbb{K}') \to T(\mathbb{K})$ in Definitions 4.1 and 4.6 are easily extended to account for the reflexive setting:

$$[\![\uparrow_p t]\!]_{\boldsymbol{x}} = \uparrow_p ((\mathbf{p}_{p,|\boldsymbol{x}|} \otimes \mathbf{id}) \cdot [\![t]\!]_{\boldsymbol{x}})$$
$$\langle \uparrow_p t \rangle = \uparrow_p (\langle t \rangle)$$

We have analogous results to those given in Theorems 4.5, 4.7 and 4.8; the details can be found in [8].

7. Conclusions and related work

We have introduced the closed action calculi, and shown that they are as expressive as the corresponding action calculi. We have also shown that our ideas simply extend to the higher-order and reflexive extensions of action calculi. The price we pay is one of presentation. The term $(x, y)(L(\langle x, y\rangle) \otimes K(\langle y \rangle))$ in the action calculi setting has the corresponding closed term $(\varDelta_{|x,y|} \otimes \mathbf{id}_{m\otimes n}) \cdot (\mathbf{id} \otimes \mathbf{p}_{|y|,m} \otimes \mathbf{id}_n) \cdot (L_{|x,y|}(\mathbf{id}) \otimes \boldsymbol{\omega} \otimes K_{|y|}(\mathbf{id}))$. In the first term, it is completely apparent how the actions contained in L and K are related. In the second term, we require a global analysis of the term to understand the relationships. The names and abstraction provide a local way of describing these connections.

Power has given the categorical models of the closed action calculi [24]. In [8], Gardner observed that the wiring terms yield a strict cartesian category. Power has taken this further, by showing the full categorical structure of closed action calculi. His models are constructed from a triple $(\mathscr{C}, \mathscr{S}, F)$, where $\mathscr{C}$ is a strict cartesian category which models the wiring terms, $\mathscr{S}$ is a strict symmetric monoidal category which models arbitrary terms, and F is an strict symmetric monoidal functor which embeds the cartesian structure in the symmetric monoidal structure. The controls correspond to natural transformations in $\mathscr{S}$, which are natural with respect to $\mathscr{C}$. This naturality condition corresponds to the control axioms D1–D3 given in Definition 3.6. Finally, there is a local preorder between morphisms in $\mathscr{S}$, which corresponds to the dynamics.

Pavlovic has also explored related categorical models for closed action calculi. We have already observed that he has similar results to those in Section 5, which adds free names to the closed action calculi. In Power's setting, this amounts to freely adding indeterminants $\langle x\rangle : \varepsilon \to p$ to the cartesian category $\mathscr{C}$ in such a way that the relevant structure is preserved. Pavlovic points out that the results are an extension of the standard notion of functional completeness for cartesian closed categories.

As well as the categorical models, we also have a type-theoretic presentation. Gardner and Hasegawa [10] show that closed action calculi can be described using known ideas from type theory, with sequents of the form $\boldsymbol{x},\boldsymbol{y} \vdash t : n$ where $|\boldsymbol{y}|$ is the domain arity m, the domain arity, and $\boldsymbol{x}$ contains the free names. Their results use an observation of Plotkin that the controls K correspond to the general binding operators of Aczel. They extend their results to the closed higher-order action calculi, and extend Power's models to capture the higher-order features. Their higher-order models relate to Moggi's semantic framework, which he calls 'notions of computation'. Hasegawa also extends the results to account for reflexion [12]. In particular, he shows that the reflexion operator corresponds to adding a trace operator, due to Joyal, Street and Verity [15], to the symmetric monoidal category $\mathscr{S}$.

Barber, Gardner, Hasegawa and Plotkin [3] have also given a direct type-theoretic presentation of action calculi, with sequents of the form $\boldsymbol{x};\boldsymbol{y} \vdash t : n$ where the names $\boldsymbol{x}$ and $\boldsymbol{y}$ are kept separate: the $\boldsymbol{x}$ behave in an intuitionistic fashion, and the $\boldsymbol{y}$ in a linear fashion. This type theory has a sound translation in Benton's type theory of intuitionistic linear logic [4], corresponding to the relation of Benton's models of linear logic to Power's models of action calculi. The conservativity of the syntactic translation is proved by a model-embedding construction using the Yoneda lemma.

In summary, the work on closed action calculi has led to a good understanding of the static part of action calculi: in particular, the presentational role that names play, and the connections with known ideas from type theory and category theory. It remains on-going research to fully understand the dynamics of action calculi. In particular, we hope that the type-theoretic and categorical presentations of action calculi will provide useful criteria for assessing which dynamic relations describe interesting interactive behaviour.

Appendix

In this appendix, we prove the key results of the paper, which relate action calculi and their corresponding closed action calculi. Our proofs involve the technical device of working with judgements of the form $\{\boldsymbol{x}\} \vdash s = t$, which denote that $s = t$ in AC and that $fn(s,t) \subseteq \{\boldsymbol{x}\}$.

Definition A.1. The *equational theory with names*, denoted by AC_n, is defined by the following rules, where $\{\boldsymbol{x}\}$ denotes a set of distinct names and the arity information is omitted since it is apparent: [1]

$$\{\boldsymbol{x}\} \vdash s = t, \qquad s = t \text{ an axiom of AC, } fn(s,t) \subseteq \{\boldsymbol{x}\}$$

$$\{\boldsymbol{x}\} \vdash s = s, \qquad fn(s) \subseteq \{\boldsymbol{x}\}$$

$$\frac{\{\boldsymbol{x}\} \vdash s = t}{\{\boldsymbol{x}\} \vdash t = s}$$

$$\frac{\{\boldsymbol{x}\} \vdash s = t \quad \{\boldsymbol{x}\} \vdash t = u}{\{\boldsymbol{x}\} \vdash s = u}$$

$$\frac{\{\boldsymbol{x}, y\} \vdash s = t}{\{\boldsymbol{x}\} \vdash (y)s = (y)t} \qquad y \notin \{\boldsymbol{x}\}$$

$$\frac{\{\boldsymbol{x}\} \vdash s = t}{\{\boldsymbol{x}\} \vdash u \cdot s = u \cdot t \quad \{\boldsymbol{x}\} \vdash s \cdot u = t \cdot u} \qquad fn(u) \subseteq \{\boldsymbol{x}\}$$

$$\frac{\{\boldsymbol{x}\} \vdash s = t}{\{\boldsymbol{x}\} \vdash u \otimes s = u \otimes t \quad \{\boldsymbol{x}\} \vdash s \otimes u = t \otimes u} \qquad fn(u) \subseteq \{\boldsymbol{x}\}$$

$$\frac{\{\boldsymbol{x}\} \vdash s_i = t_i, \quad i = 1, \ldots, r}{\{\boldsymbol{x}\} \vdash K(s_1, \ldots, s_r) = K(t_1 \ldots, t_r)}$$

Proposition A.2. 1. $\{\boldsymbol{x}\} \vdash s = t$ *in* AC_n *implies* $fn(s,t) \subseteq \{\boldsymbol{x}\}$.
2. $\{\boldsymbol{x}, y\} \vdash s = t$ *in* AC_n *and* $z \notin \{\boldsymbol{x}\}$ *imply* $\{\boldsymbol{x}, z\} \vdash s\{z/y\} = t\{z/y\}$ *in* AC_n.
3. (*weakening*) $\{\boldsymbol{x}\} \vdash s = t$ *in* AC_n *and* $y \notin \{\boldsymbol{x}\}$ *imply* $\{\boldsymbol{x}, y\} \vdash s = t$ *in* AC_n.
4. (*strengthening*) $\{\boldsymbol{x}, y\} \vdash s = t \in \mathrm{AC}_n$ *and* $y \notin fn(s,t)$ *imply* $\{\boldsymbol{x}\} \vdash s = t \in \mathrm{AC}_n$.

Proof. The proofs of parts (1)–(3) are easy. The proof of part (4) is less straightforward. It relies on the connection between AC_n and the alternative presentation of actions using the molecular forms. See [11] for a detailed proof. □

Proposition A.3. $s = t$ *in* AC *if and only if* $fn(s,t) \vdash s = t$ *in* AC_n.

Proof. Both implications are easy. The implication from left to right requires Proposition A.2. □

We have shown the connection between AC and AC_n. It remains to prove the connection between AC_n and CAC. First, we state some technical results about the translation $[\![\_]\!]_{\boldsymbol{x}}$ used to simplify the proof that the translation preserves equality.

Lemma A.4. *The following hold in* CAC:
1. $[\![s \otimes t]\!]_{\boldsymbol{x}} = [\![s]\!]_{\boldsymbol{x}} \otimes [\![t]\!]_{[\,]}$, *if* $fn(t) = \emptyset$;
2. $[\![t \otimes s]\!]_{\boldsymbol{x}} = (\mathbf{p}_{|\boldsymbol{x}|,m} \otimes \mathbf{id}) \cdot ([\![t]\!]_{[\,]} \otimes [\![s]\!]_{\boldsymbol{x}})$, *if* $fn(t) = \emptyset$ *and* $t : m \to n$;
3. $[\![s \cdot t]\!]_{\boldsymbol{x}} = [\![s]\!]_{\boldsymbol{x}} \cdot [\![t]\!]_{[\,]}$, *if* $fn(t) = \emptyset$;

[1] We use a rule with two conclusions as shorthand for two rules with the same premises and one conclusion each.

4. $[\![t\cdot s]\!]_{\boldsymbol{x}} = (\mathbf{id}\otimes[\![t]\!]_{[\,]})\cdot[\![s]\!]_{\boldsymbol{x}}$, *if* $fn(t)=\emptyset$;
5. $[\![\mathbf{p}_{m,n}]\!]_{\boldsymbol{x}} = \boldsymbol{\omega}_{|\boldsymbol{x}|}\otimes\mathbf{p}_{m,n}$;
6. $[\![\langle x\rangle_{\boldsymbol{x}} = \mathbf{id}_{|\boldsymbol{x}|}$.

Proof. The proof involves straightforward equational reasoning. □

Theorem A.5. $\{\boldsymbol{x}\}\vdash s=t$ *in* AC *implies* $[\![s]\!]_{\boldsymbol{x}}=[\![t]\!]_{\boldsymbol{x}}$ *in* CAC.

Proof. The proof that the translation $[\![\_]\!]_{\boldsymbol{x}}$ preserves the basic axioms involves simple equational reasoning using the axioms A1–A6 and B1–B8. It is easy to prove that the reflexive, symmetric and transitive rules, and the structural rules are preserved under translation; the structural rule for abstraction requires Lemma 4.3. Here we prove that the concrete axioms are preserved.

The proof that the axiom γ is preserved requires Lemmas 4.4 and A.4:

$$\begin{aligned}
[\![\boldsymbol{\omega}_p\otimes s]\!]_{\boldsymbol{x}} &\overset{\text{A.4}}{=} (\mathbf{p}_{|\boldsymbol{x}|,p}\otimes\mathbf{id})\cdot(\boldsymbol{\omega}_p\otimes[\![s]\!]_{\boldsymbol{x}})\\
&\overset{4.4}{=} (\mathbf{p}_{|\boldsymbol{x}|,p}\otimes\mathbf{id})\cdot[\![s]\!]_{\boldsymbol{y},\boldsymbol{x}}, \quad \boldsymbol{y}\notin\{\boldsymbol{x}\}\\
&\overset{4.4}{=} [\![s]\!]_{\boldsymbol{x},\boldsymbol{y}}\\
&= [\![(\boldsymbol{y})s]\!]_{\boldsymbol{x}}
\end{aligned}$$

The proof that the axiom δ is preserved is easy:

$$\begin{aligned}
[\![(x)(\langle x\rangle\otimes\mathbf{id})]\!]_{\boldsymbol{x}} &= [\![\langle y\rangle\otimes\mathbf{id}]\!]_{\boldsymbol{x},y} \quad y\notin\{\boldsymbol{x}\}\\
&\overset{\text{A.4}}{=} \boldsymbol{\omega}_{|\boldsymbol{x}|}\otimes\mathbf{id}\otimes\mathbf{id}\\
&= [\![\mathbf{id}]\!]_{\boldsymbol{x}}
\end{aligned}$$

Proving that σ is preserved requires Lemma 4.4. Let $\boldsymbol{x}$ be $\boldsymbol{y},u:p,\boldsymbol{z}$ such that $|\boldsymbol{y}|=k$ and $|\boldsymbol{z}|=l$:

$$\begin{aligned}
&[\![(\langle u\rangle\otimes\mathbf{id})\cdot(x)s]\!]_{\boldsymbol{y},u,\boldsymbol{z}}\\
&\overset{\text{A.4}}{=} (\varDelta_{k\otimes p\otimes l}\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\boldsymbol{\omega}_k\otimes\mathbf{id}\otimes\boldsymbol{\omega}_l\otimes\mathbf{id})\cdot[\![s\{v/x\}]\!]_{\boldsymbol{y},u,\boldsymbol{z},v}\\
&= (\mathbf{id}\otimes\varDelta_p\otimes\mathbf{id}\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\mathbf{id}\otimes\mathbf{p}_{p,l}\otimes\mathbf{id})\cdot[\![s\{v/x\}]\!]_{\boldsymbol{y},u,\boldsymbol{z},v}\\
&= (\mathbf{id}\otimes\varDelta_p\otimes\mathbf{id}\otimes\mathbf{id})\cdot[\![s\{v/x\}]\!]_{\boldsymbol{y},u,v,\boldsymbol{z}}\\
&\overset{\text{A.4}}{=} [\![s\{v/x\}\{u/v\}]\!]_{\boldsymbol{y},u,\boldsymbol{z}}\\
&= [\![s\{u/x\}]\!]_{\boldsymbol{y},u,\boldsymbol{z}}
\end{aligned}$$

The proof that ζ is preserved by the translation involves simple equational reasoning:

$$\begin{aligned}
&[\![\mathbf{p}_{k,m}\cdot(t\otimes s)]\!]_{\boldsymbol{x}}\\
&\overset{\text{A.4}}{=} (\mathbf{id}\otimes\mathbf{p}_{k,m})\cdot(\varDelta_{|\boldsymbol{x}|}\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\mathbf{p}_{|\boldsymbol{x}|,m}\otimes\mathbf{id})\cdot([\![t]\!]_{\boldsymbol{x}}\otimes[\![s]\!]_{\boldsymbol{x}})\\
&\overset{\text{B.5, B.4}}{=} (\varDelta_{|\boldsymbol{x}|}\otimes\mathbf{id})\cdot(\mathbf{id}\otimes\mathbf{p}_{|\boldsymbol{x}|\otimes k,m})\cdot([\![t]\!]_{\boldsymbol{x}}\otimes[\![s]\!]_{\boldsymbol{x}})\cdot\mathbf{p}_{n,l}\cdot\mathbf{p}_{l,n}
\end{aligned}$$

$$\overset{\text{B.3}}{=} (\Delta_{|\boldsymbol{x}|} \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \mathbf{p}_{|\boldsymbol{x}| \otimes k, m}) \cdot \mathbf{p}_{|\boldsymbol{x}| \otimes m, |\boldsymbol{x}| \otimes k} \cdot (\llbracket s \rrbracket_{\boldsymbol{x}} \otimes \llbracket t \rrbracket_{\boldsymbol{x}}) \cdot \mathbf{p}_{l,n}$$

$$\overset{\text{B.5,B.4}}{=} (\Delta_{|\boldsymbol{x}|} \otimes \mathbf{id}) \cdot (\mathbf{p}_{|\boldsymbol{x}|, |\boldsymbol{x}| \otimes k} \otimes \mathbf{id}) \cdot (\llbracket s \rrbracket_{\boldsymbol{x}} \otimes \llbracket t \rrbracket_{\boldsymbol{x}}) \cdot \mathbf{p}_{l,n}$$

$$= (\Delta_{|\boldsymbol{x}|} \otimes \mathbf{id}) \cdot (\mathbf{p}_{|\boldsymbol{x}|, |\boldsymbol{x}|} \otimes \mathbf{id} \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \mathbf{p}_{|\boldsymbol{x}|, k} \otimes \mathbf{id}) \cdot (\llbracket s \rrbracket_{\boldsymbol{x}} \otimes \llbracket t \rrbracket_{\boldsymbol{x}}) \cdot \mathbf{p}_{l,n}$$

$$\overset{\text{B.2}}{=} \llbracket s \otimes t \rrbracket_{\boldsymbol{x}} \cdot \mathbf{p}_{l,n}$$

$$\overset{\text{A.4}}{=} \llbracket (s \otimes t) \cdot \mathbf{p}_{l,n} \rrbracket_{\boldsymbol{x}}$$

Corollary A.6. *$s = t$ in* AC *implies* $\llbracket s \rrbracket_{fn(s,t)} = \llbracket t \rrbracket_{fn(s,t)}$ *in* CAC.

Proof. The proof follows immediately from Proposition A.3 and Theorem A.5. □

Theorem A.7. *$s = t$ in* CAC *implies* $\emptyset \vdash \langle s \rangle = \langle t \rangle$ *in* AC_n.

Proof. The proof that the axioms A1–A6 and B1–B8 are preserved under translation is easy, since $\langle \_ \rangle$ preserves the structure of these axioms and the corresponding equalities hold in AC_n. The proof that the structural rules, and the reflexive, symmetric and transitive rules are preserved is trivial using the induction hypothesis. The interesting cases are the axioms D1–D3:

$$\emptyset \vdash \langle K_{n \otimes m}(\boldsymbol{\omega}_n \otimes t_1, \ldots, \boldsymbol{\omega}_n \otimes t_r) \rangle$$

$$= (\boldsymbol{y}, \boldsymbol{x}) K((\langle \boldsymbol{y}, \boldsymbol{x} \rangle \otimes \boldsymbol{id}) \cdot (\boldsymbol{\omega} \otimes \langle t_1 \rangle), \ldots, (\langle \boldsymbol{y}, \boldsymbol{x} \rangle \otimes \boldsymbol{id}) \cdot (\boldsymbol{\omega} \otimes \langle t_r \rangle)), |\boldsymbol{y}| = n, |\boldsymbol{x}| = m$$

$$\overset{\gamma,\sigma}{=} (\boldsymbol{y}, \boldsymbol{x}) K((\langle \boldsymbol{x} \rangle \otimes \mathbf{id}) \cdot \langle t_1 \rangle, \ldots, (\langle \boldsymbol{x} \rangle \otimes \mathbf{id}) \cdot \langle t_r \rangle)$$

$$= (\boldsymbol{y}) \langle K_m(t_1, \ldots, t_r) \rangle$$

$$\overset{\gamma}{=} \boldsymbol{\omega}_n \otimes \langle K_m \rangle (t_1, \ldots, t_r) \rangle$$

$$= \langle \boldsymbol{\omega}_n \otimes K_m(t_1, \ldots, t_r) \rangle$$

$$\emptyset \vdash \langle K_{k \otimes m \otimes n}((\mathbf{id} \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}) \cdot t_1, \ldots, (\mathbf{id} \otimes \mathbf{p}_{m,n} \otimes \mathbf{id}) \cdot t_r) \rangle$$

$$= (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) K((\langle \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \rangle \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \boldsymbol{p}_{m,n} \otimes \mathbf{id}) \cdot \langle t_1 \rangle, \ldots, (\langle \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \rangle \otimes \mathbf{id}) \cdot$$

$$(\mathbf{id} \otimes \boldsymbol{p}_{m,n} \otimes \mathbf{id}) \cdot \langle t_r \rangle)), \quad |\boldsymbol{x}| = k, |\boldsymbol{y}| = m, |\boldsymbol{z}| = n$$

$$\overset{\sigma}{=} (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) K((\langle \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y} \rangle \otimes \mathbf{id}) \cdot \langle t_1 \rangle, \ldots, (\langle \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y} \rangle \otimes \mathbf{id}) \cdot \langle t_r \rangle)$$

$$\overset{\sigma}{=} (\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w})((\langle \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v} \rangle \otimes \mathbf{id}) \cdot (\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y}) K((\langle \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y} \rangle \otimes \mathbf{id}) \cdot$$

$$\langle t_1 \rangle, \ldots, (\langle \boldsymbol{x}, \boldsymbol{z}, \boldsymbol{y} \rangle \otimes \mathbf{id}) \cdot \langle t_r \rangle)$$

$$\overset{\sigma}{=} (\mathbf{id} \otimes \boldsymbol{p}_{m,n} \otimes \mathbf{id}) \cdot \langle K_{k \otimes n \otimes m}(t_1, \ldots, t_r) \rangle$$

$$= \langle (\mathbf{id} \otimes \boldsymbol{p}_{m,n} \otimes \mathbf{id}) \cdot K_{k \otimes n \otimes m}(t_1, \ldots, t_r) \rangle$$

$$\emptyset \vdash \langle K_{k\otimes m}((\mathbf{id}\otimes \Delta_m \otimes \mathbf{id})\cdot t_1,\ldots,(\mathbf{id}\otimes \Delta_m \otimes \mathbf{id})\cdot t_r)\rangle$$

$$= (\boldsymbol{x},\boldsymbol{y})K((\langle \boldsymbol{x},\boldsymbol{y}\otimes \mathbf{id})\cdot(\mathbf{id}\otimes \mathbf{copy}_m \otimes \mathbf{id})\cdot\langle t_1\rangle,\ldots,(\langle \boldsymbol{x},\boldsymbol{y}\rangle\otimes \mathbf{id})\cdot$$

$$(\mathbf{id}\otimes \mathbf{copy}_m\otimes \mathbf{id})\cdot\langle t_r\rangle),\quad |\boldsymbol{x}|=k, |\boldsymbol{y}|=m$$

$$\overset{\sigma}{=} (\boldsymbol{x},\boldsymbol{y})K((\langle \boldsymbol{x},\boldsymbol{y},\boldsymbol{y}\rangle\otimes \mathbf{id})\cdot\langle t_1\rangle,\ldots,(\langle \boldsymbol{x},\boldsymbol{y},\boldsymbol{y}\rangle\otimes \mathbf{id})\cdot\langle t_r\rangle)$$

$$\overset{\sigma}{=} (\boldsymbol{x},\boldsymbol{y})((\langle \boldsymbol{x},\boldsymbol{y},\boldsymbol{y}\rangle\otimes \mathbf{id})\cdot(\boldsymbol{u},\boldsymbol{v},\boldsymbol{w})K((\langle \boldsymbol{u},\boldsymbol{v},\boldsymbol{w}\rangle\otimes \mathbf{id})\cdot$$

$$\langle t_1\rangle,\ldots,(\langle \boldsymbol{u},\boldsymbol{v},\boldsymbol{w}\rangle\otimes \mathbf{id})\cdot\langle t_r\rangle))$$

$$\overset{\sigma}{=} (\mathbf{id}\otimes \mathbf{copy}_m\otimes \mathbf{id})\cdot\langle K_{k\otimes m\otimes m}(t_1,\ldots,t_r)\rangle$$

$$= \langle(\mathbf{id}\otimes \Delta_m\otimes \mathbf{id})\cdot K_{k\otimes m\otimes m}(t_1,\ldots,t_r)\rangle$$

Corollary A.8. *$s=t$ in CAC implies $\langle s\rangle=\langle t\rangle$ in AC.*

Theorem A.9. 1. *Given $t\in T_{\boldsymbol{x}}(\mathbb{K})$, we have $\emptyset\vdash\langle[\![t]\!]_{\boldsymbol{x}}\rangle=(\boldsymbol{x})t$ in AC_n.*
2. *Given $s\in \mathrm{CT}(\mathbb{K})$, we have $[\![\langle s\rangle]\!]_{\boldsymbol{x}}=\boldsymbol{\omega}_{|\boldsymbol{x}|}\otimes s$ in CAC.*

Proof. Both parts are proved by induction on the structure of t. We just give the control case for each part, to illustrate the movement of names through the controls. The other cases involve simple equational reasoning:

$$\emptyset\vdash\langle[\![K(t_1,\ldots,t_r)]\!]_{\boldsymbol{x}}\rangle = \langle K_{|\boldsymbol{x}|}([\![t_1]\!]_{\boldsymbol{x}},\ldots,[\![t_r]\!]_{\boldsymbol{x}})\rangle$$

$$\overset{\mathrm{IH}}{=} (\boldsymbol{x})K((\langle\boldsymbol{x}\rangle\otimes \mathbf{id})\cdot(\boldsymbol{x})t_1,\ldots,(\langle\boldsymbol{x}\rangle\otimes \mathbf{id})\cdot(\boldsymbol{x})t_r)$$

$$\overset{\sigma}{=} (\boldsymbol{x})K(t_1,\ldots,t_r)$$

$$[\![\langle K_m(s_1,\ldots,s_r)\rangle]\!]_{\boldsymbol{x}} = [\![(\boldsymbol{y})K((\langle\boldsymbol{y}\rangle\otimes \mathbf{id})\cdot\langle s_1\rangle,\ldots,(\langle\boldsymbol{y}\rangle\otimes \mathbf{id})\cdot\langle s_r\rangle)]\!]_{\boldsymbol{x}},\quad |\boldsymbol{y}|=m$$

$$= [\![K((\langle\boldsymbol{z}\rangle\otimes \mathbf{id})\cdot\langle s_1\rangle,\ldots,(\langle\boldsymbol{z}\rangle\otimes \mathbf{id})\cdot\langle s_r\rangle)]\!]_{\boldsymbol{x},\boldsymbol{z}}$$

$$= K_{|\boldsymbol{x}|\otimes m}([\![(\langle z\rangle\otimes \mathbf{id})\cdot\langle s_1\rangle]\!]_{\boldsymbol{x},z},\ldots,[\![(\langle z\rangle\otimes \mathbf{id})\cdot\langle s_r\rangle]\!]_{\boldsymbol{x},z})$$

$$\overset{\mathrm{A.4}}{=} K_{|\boldsymbol{x}|\otimes m}([\![\langle\boldsymbol{z}\rangle\otimes \mathbf{id}]\!]_{\boldsymbol{x},\boldsymbol{z}}\cdot s_1,\ldots,[\![\langle\boldsymbol{z}\rangle\otimes \mathbf{id}]\!]_{\boldsymbol{x},\boldsymbol{z}}\cdot s_r)$$

$$\overset{\mathrm{IH}}{=} K_{|\boldsymbol{x}|\otimes k}(\boldsymbol{\omega}_{|\boldsymbol{x}|}\otimes s_1,\ldots,\boldsymbol{\omega}_{|\boldsymbol{x}|}\otimes s_r)$$

$$\overset{\mathrm{D1}}{=} \boldsymbol{\omega}_{|\boldsymbol{x}|}\otimes K_m(s_1,\ldots,s_r)\qquad\square$$

Corollary A.10. 1. *Given $t\in T_{\boldsymbol{x}}(\mathbb{K})$, we have $\langle[\![t]\!]_{\boldsymbol{x}}\langle=(\boldsymbol{x})t$ in AC.*
2. *Given $s\in\mathrm{CT}(\mathbb{K})$, we have $[\![\langle s\rangle]\!]_{\emptyset}=s$ in CAC.*

Acknowledgements

My thanks go to Robin Milner for his encouragement and many interesting discussions regarding this work, and to Masahito Hasegawa for his comments on a previous draft.

References

[1] M. Abadi, L. Cardelli, A theory of objects, Springer, Berlin, 1996.
[2] M. Abadi, A. Gordon, A calculus for cryptgraphic protocols, Proc. 4th ACM Conf. of Computer and Communications Security, 1997.
[3] A. Barber, P. Gardner, M. Hasegawa, G. Plotkin, From action calculi to linear logic, Selected papers from the annual conference of the European Association for Computer Science Logic (CSL'97), Lecture Notes in Computer Science, vol. 1414, Springer, Berlin, 1998, pp. 78–97.
[4] N. Benton, A mixed linear and non-linear logic; proofs, terms and models, Proc. Computer Science Logic, Lecture Notes in Computer Science, vol. 933, Springer, Berlin, 1995.
[5] G. Boudol, Asynchrony and the π-calculus (note), Rapport de Recherche 1702, INRIA Sofia-Antipolis, 1992.
[6] L. Cardelli, A. Gordon, Mobile ambients, Foundations of Software Science and Computational Structures, M. Nivat (Ed.), Lecture Notes in Computer Science, vol. 1378, Springer, Berlin, 1998, pp. 140–155.
[7] A. Corradini, F. Gadducci, An algebraic presentation of term graphs via ps-monoidal categories, Applied Categorical Structures, to appear, 1999.
[8] P.A. Gardner, A name-free account of action calculi, Full manuscript, 1994.
[9] P.A. Gardner, A name-free account of action calculi, in: S. Brookes, M. Main, A. Melton, M. Mislove (Eds.), Mathematical foundations of Programming Semantics, ENTCS 1. Elsevier, Amsterdam, 1995.
[10] P.A. Gardner, M. Hasegawa, Types and models for higher-order action calculi, Theoretical Aspects of Computer Software, Sendai, 1997.
[11] P.A. Gardner, R. Milner, Action calculi VIII: Contextual Action Calculi, Manuscript, 1994.
[12] M. Hasegawa, Models of sharing graphs (A categorial semantics of let and lectrec), Ph.D. Thesis, LFCS, University of Edinburgh, 1997.
[13] J.R. Hindley, J. Seldin, Introduction to Combinators and Lambda-Calculus, Cambridge University Press, Cambridge, 1986.
[14] K. Honda, M. Tokoro, On asynchronous communication semantics, Object-based Concurrent Computing, Lecture Notes in Computer Science, vol. 612, Springer, Berlin, 1992, 21–51.
[15] A. Joyal, R. Street, D. Verity, Traced monoidal categories, Math. Proc. Cambridge Philos. Soc. 119(3) (1996) 447–468.
[16] A. Mifsud, Control structures, Ph.D. Thesis, LFCS, University of Edinburgh, 1996.
[17] R. Milner, Action calculi, or syntactic action structures, in: A.M. Borzyszkowski, S. Sokolowski (Eds.), Mathematical Foundations of Computer Science, Lecture Notes in Computer Science, vol. 711, Springer, Berlin, 1993, 105–121.
[18] R. Milner, Action calculi V: Reflexive Action Calculi, Manuscript, 1994.
[19] R. Milner, Higher-order calculi, Proc. Computer Science Logic, Lecture Notes in Computer Science, vol. 832, Springer, Berlin, 1994.
[20] R. Milner, Calculi for interaction, Acta Inform. 33(8) (1996) 707–737.
[21] R. Milner, J. Parrow, D. Walker, A calculus of mobile processes, Inform. and Control 100 (1992) 1–77.
[22] E. Moggi, Notions of computation and monads, Inform. and Comput. 93(1) (1991) 55–92.
[23] D. Pavlovic, Categorical logic of names and abstraction in action calculi, Math. Struct. Comput. Sci. 7(6) (1997) 619–637.
[24] A.J. Power, Elementary control structures, Proc. CONCUR '96, Lecture Notes in Computer Science, vol. 1119, Springer, Berlin, 1996.
[25] P. Sewell, Global/local subtyping and capability inference for a distributed π-calculus, Proc. International Colloquium an Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 1443, Springer, Berlin, 1998, pp. 695–706.