Dynamic Disciplines in the Action Calculus

James J. Leifer

Supervised by:

Professor Robin Milner

30 July 96

Cambridge University Computing Laboratory
New Museums Site
Pembroke St.
Cambridge, CB2 3QG
UK

Part I
First Year Report

1 Introduction

The Action Calculus (AC), originally developed to explore the neighbourhood of systems related to the π-calculus, provides a uniform means of encoding the syntax and semantics of diverse operational calculi, including the π-calculus, join-calculus, Petri-nets, λ-calculus, and fragments of ML. The AC is young and the definitions for much of the supporting theory surrounding the AC are in flux — changing often to accommodate our desire to prove theorems that, "morally" at least, ought to be true. Thus, the following discussion represents a combination of my work done on the AC over the past two terms and my *current* understanding of the subject — where the two are sometimes very different.

My goal has been to construct a Control Structure (CS, a model of the AC) called IMGRAPH that characterises those calculi that do not have the property of *mobility*. A process algebra is *mobile* if (informally) it can describe processes that are not connected by a communication channel at one point in their interaction, but via the intervention of some other process that "introduces" the first two, they then become connected later. While we believe that the π-calculus (Milner, 1992) is mobile (its distinguishing feature) but CCS (Milner, 1989) and the λ-calculus (Barendregt, 1992) are not, we cannot compare these calculi directly in any precise way on the basis of mobility because their syntax and semantics are stated differently. My plan was to embed the three into the AC —which provides a uniform presentation of their syntax and dynamics (semantics)— and then to define IMGRAPH so that its dynamics included CCS's and the λ-calculus's but excluded all "mobility" (or, at least, excluded the dynamics of the π-calculus). Thus, we might *define* mobile action calculi as those from which there is no CS morphism to IMGRAPH, i.e., for which IMGRAPH is not a model.

The development of IMGRAPH involved many false starts, some of which are instructive to retell. Hence I will try to interleave the presentation with explicit mention of a few of the design decisions taken in our current version of IMGRAPH and to provide motivation by making visible some of the misconceptions we overcame.

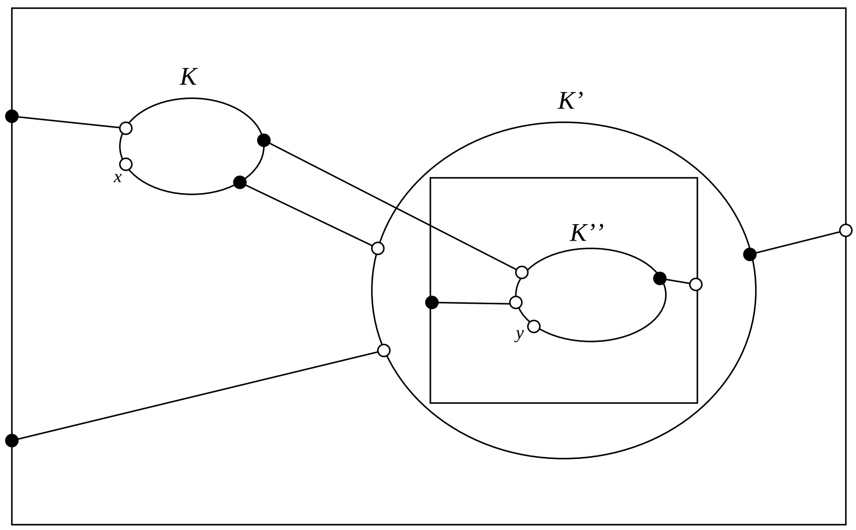

Figure 1: The filled ports are sources, the empty ones are sinks. If we take the arity monoid to be the natural numbers then this action is an arrow $2 \to 1$. Note that the action contained in K' is of arity $1 \to 1$ — the edge from the first source of K connects *directly* to the first sink of K''.

2 LAMC and PIC

In this section I will introduce informally the AC itself and, formally, the AC representations of the π-calculus and λ-calculus. I will then describe how mobility manifests itself graphically in the dynamics of PIC (the AC encoding for the π-calculus) but not in LAMC (the encoding for the λ-calculus) — an observation that motivates the construction of IMGRAPH.

The appendix to this report contains a summary of the algebraic axioms and notational conventions for the AC. Informally, though, an AC consists of a set of terms (also known as *actions*) and a reaction preorder on the terms. The actions, though representable algebraically, are essentially graphical. Actions consist of an input and output interface, whose structure is taken from an arity monoid, with source and sink ports, respectively. The body of the actions is a graph whose nodes, called *controls*, are labelled by a symbol from the signature of the AC. Each control is equipped with sink and source ports (corresponding to its arity) and may contain a list of actions nested inside — the number determined by the *rank* of the control. Every sink port is either labelled with a *name* (from the name-set of the AC) or is the target of exactly one edge originating from a source port. Each source port has zero or more edges leading from it. (See Figure 1.) Pictures bristling with ports are difficult to manipulate easily, so we will often dispense with the ports and simply put an arrow-head at the end of each edge to indicate unambiguously its target.

2.1 Definition (PIC) We consider PIC, a fragment of the unsorted π-calculus. It has two controls, **box** and **out**, which have the following arity rules:

$$\frac{a : 1 \to 0}{\mathbf{box}(a) : 1 \to 0} \qquad \mathbf{out} : 2 \to 0 \quad .$$

The reaction relation $\xrightarrow{\mathsf{Pi}}$ is generated by the single rule:

$$\mathbf{out}_x \otimes \mathbf{box}_x(a) \xrightarrow{\mathsf{Pi}} a$$

where

$$\begin{aligned} \mathbf{out}_x &\stackrel{\text{def}}{=} (\mathbf{id}_1 \otimes \langle x \rangle) \cdot \mathbf{out} \\ \mathbf{box}_x(a) &\stackrel{\text{def}}{=} \langle x \rangle \cdot \mathbf{box}(a) \end{aligned}$$

■

2.2 Definition (LAMC) We consider an untyped version of the λ-calculus with signature $\{\boldsymbol{\lambda}, \mathbf{ap}\}$ and arity inference rules:

$$\frac{a : m \to n}{\boldsymbol{\lambda}(a) : 0 \to 1} \qquad \mathbf{ap} : 1 + m \to n \quad .$$

The reaction relation is generated by:

$$\begin{aligned} \boldsymbol{\lambda}(a) \cdot \mathbf{copy} &\longrightarrow \boldsymbol{\lambda}(a) \otimes \boldsymbol{\lambda}(a) \\ \boldsymbol{\lambda}(a) \cdot \boldsymbol{\omega} &\longrightarrow \mathbf{id}_0 \\ [{}^a/_x]K(b_1, \ldots, b_n) &\longrightarrow \{{}^{\boldsymbol{\lambda}(a)}/_x\}K(b_1, \ldots, b_n) \end{aligned}$$

Note: In the last rule K ranges over $\{\boldsymbol{\lambda}, \mathbf{ap}\}$. We state the rule this way because when LAMC is added as modular extension to some other AC, we want K to range over the signature of the combined controls from both systems. The definitions of semantic substitution $[{}^-/_\_]$ and syntactic substitution $\{{}^-/_\_\}$ are in the appendix to this report. ■

2.3 Lemma (small-step substitution dynamics) The reaction relation of LAMC is equivalent to the one generated by the following "small-step" substitution rules:

$$\begin{aligned} (\boldsymbol{\lambda}(a) \otimes \mathbf{id}) \cdot \mathbf{ap} &\xrightarrow{\mathsf{Lam}} a \\ \boldsymbol{\lambda}(a) \cdot \mathbf{copy} &\xrightarrow{\mathsf{Lam}} \boldsymbol{\lambda}(a) \otimes \boldsymbol{\lambda}(a) \\ \boldsymbol{\lambda}(a) \cdot \boldsymbol{\omega} &\xrightarrow{\mathsf{Lam}} \mathbf{id}_0 \\ \boldsymbol{\lambda}(a) \cdot (x)\boldsymbol{\lambda}(b) &\xrightarrow{\mathsf{Lam}} \boldsymbol{\lambda}([{}^a/_x]b) \end{aligned}$$

These are depicted in Figures 5, 6, 7, and 8.

Proof Several term inductions. ■

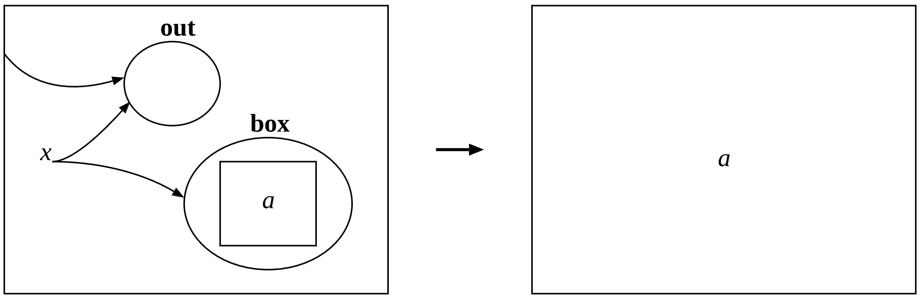

Figure 2: $\mathbf{out}_x \otimes \mathbf{box}_x(a) \xrightarrow{\mathsf{Pi}} a$

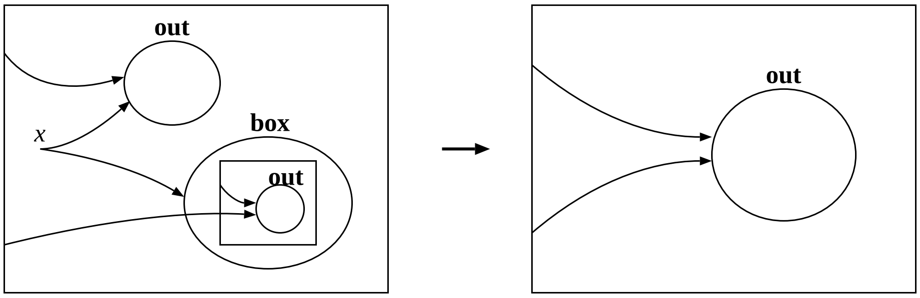

Figure 3: $(wy)(\langle w\rangle \cdot \mathbf{out}_x \otimes \mathbf{box}_x(\mathbf{out}_y)) \xrightarrow{\mathsf{Pi}} \mathbf{out}$

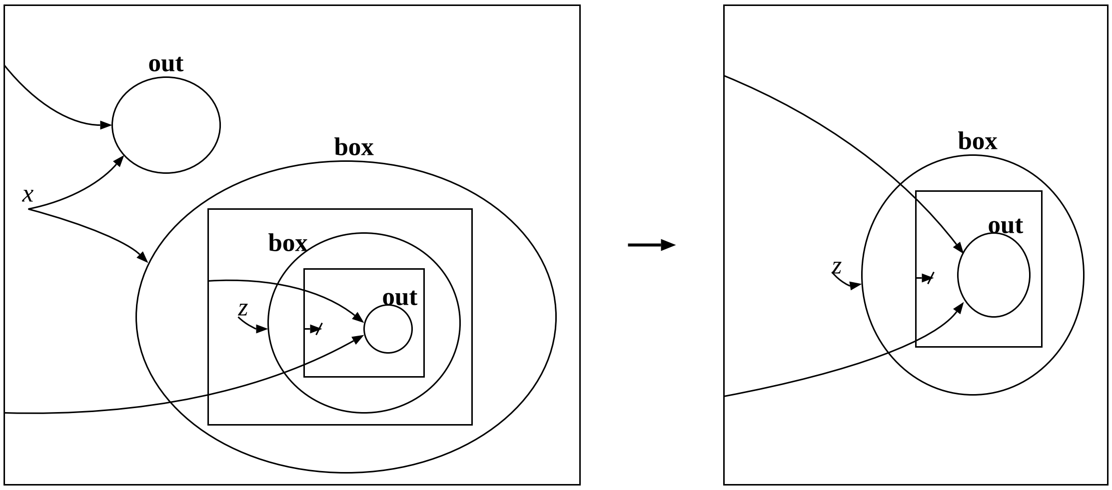

Figure 4: A dynamic reaction that cannot be embedded in IMGRAPH

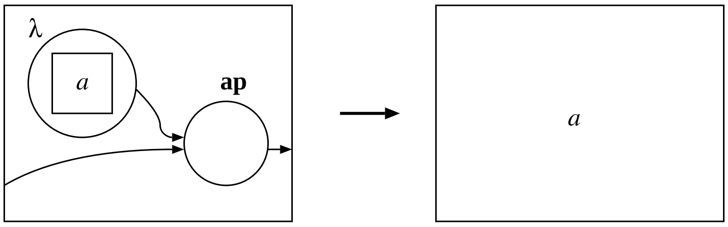

Figure 5: $(\boldsymbol{\lambda}(a) \otimes \mathbf{id}) \cdot \mathbf{ap} \xrightarrow{\mathsf{Lam}} a$

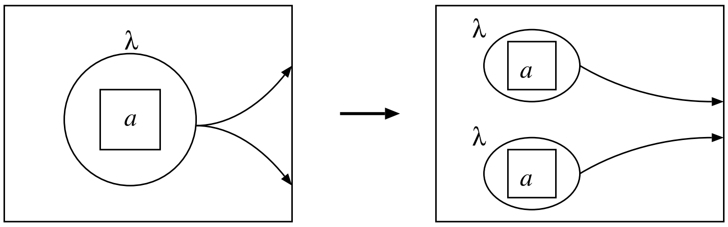

Figure 6: $\boldsymbol{\lambda}(a) \cdot \mathbf{copy} \xrightarrow{\mathsf{Lam}} \boldsymbol{\lambda}(a) \otimes \boldsymbol{\lambda}(a)$

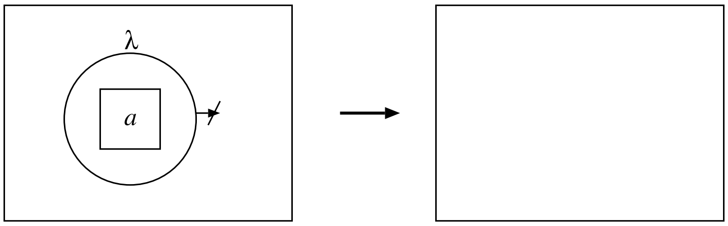

Figure 7: $\boldsymbol{\lambda}(a) \cdot \boldsymbol{\omega} \xrightarrow{\mathsf{Lam}} \mathbf{id}_0$

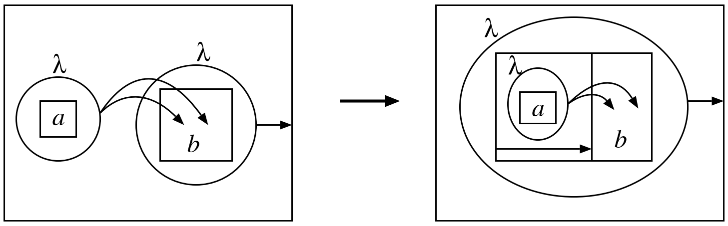

Figure 8: $\boldsymbol{\lambda}(a) \cdot (x)\boldsymbol{\lambda}(b) \xrightarrow{\mathsf{Lam}} \boldsymbol{\lambda}([{}^{a}\!/_{x}]b)$

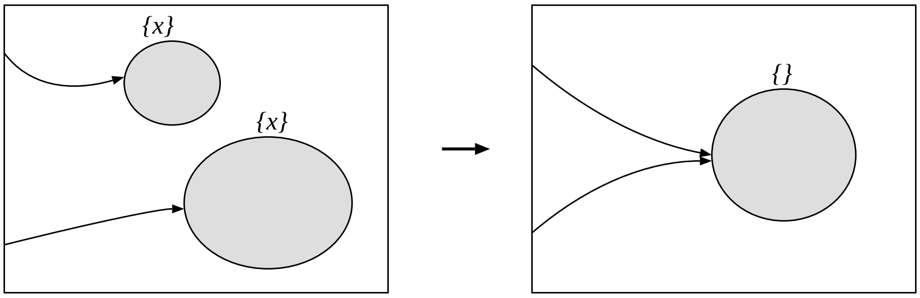

Figure 9: A view of Figure 3 through "rose-coloured glasses"

We now argue that the reactions of LAMC satisfy a dynamic discipline not satisfied by the reactions of PIC. Imagine donning a special pair of "rose-coloured" glasses through which we observe the reactions of all ACs, in particular LAMC and PIC. Suppose that the glasses filter out much of the detail: (i) we see every control construction, say, $K(a)$, as an empty, anonymous node, decorated by a set consisting of the union of the surface of a and the names that bind the control; (ii) we ignore the order and multiplicity of the edges coming from or going to a node; (iii) we view every edge that connects from outside a control construction to inside an action contained within the control as an edge to the control directly. In other words, our view through rose-coloured glasses of any action is of a mathematician's typical flat, directed graph with a few minor additions: the graph has a sequence of source ports and a sequence of sink ports aside from the nodes and each node is decorated with a set of names.

Figure 9 provides the key inspiration for how mobility manifests itself graphically: in the left-hand picture the two input resources (sources of the interface) have no path between them, but, after a PIC reaction, they become connected (the right-hand picture). Let us say that an action is *source-connected* iff there is a pair of source ports r and s connected by a zig-zag path: i.e., a path through a sequence of nodes that leads from r to s, where the edges of the graph are treated as undirected. An action is *source-separated* iff it is not source-connected. Then, we claim: *Every reaction in* LAMC *preserves source-separation, and this is exactly the property that the reactions of* PIC *need not preserve because of their mobility.*

How do we know that LAMC's reactions preserve source-separation? Let us invent two dynamic reactions that our rose-coloured graphs may undergo. We will show that these dynamics preserve source-separation and that LAMC's dynamics are special cases. The first is the *exploding* dynamic: a node "explodes" into a subgraph and the edges that enter and leave the node before the explosion, connect, after the explosion, to arbitrary parts of the subgraph (see Figure 11). The second is the *coalescing* dynamic: two nodes connected by an edge merge together to form one node whose edges are the union of all the edges of the original two nodes (see Figure 10). It is clear that both dynamics preserve source-separation (formal proofs to follow in later sections) and that LAMC's dynamics are special cases: viewed through rose-coloured glasses the middle two rules for $\xrightarrow{\mathsf{Lam}}$ (Figures 6 and 7) are instances of the exploding reaction; the fourth rule (Figure 8) is a coalescing reaction; and the first (Figure 5) is a coalescing of the $\boldsymbol{\lambda}$ and **ap** nodes followed by an explosion.

In the next sections we define IMGRAPH—"immobility graph"— a control structure (an AC model, see the appendix) over a signature $\{\mathbf{N}_{m,n}\}$ of node constructions. IMGRAPH's statics encode the flat graphs described above and its dynamics are exactly the exploding and coalescing reactions. We prove that IMGRAPH's dynamics preserve source-separation and there is a pair of rose-coloured glasses through which to observe the reactions of LAMC in IMGRAPH — that is, there is a morphism $\psi : \mathsf{LAMC} \to \mathsf{IMGRAPH}$ for which $t \longrightarrow t'$ in LAMC implies $\psi(t) \longrightarrow \psi(t')$ in IMGRAPH.

The interesting claim is that for *any* candidate morphism $\varphi : \mathsf{PIC} \to \mathsf{LAMC}$ there are terms t, t' in PIC for which $t \longrightarrow t'$ in PIC but $\varphi(t) \not\longrightarrow \varphi(t')$ in IMGRAPH. The conclusion follows if we show that $\varphi(t)$ is source-connected and $\varphi(t')$ is source-separated, for then $\varphi(t) \longrightarrow \varphi(t')$ contradicts the preservation of source-separateness by IMGRAPH's dynamics. Finding such a pair t, t' is trivial (compare Figures 3 and 9) if the only morphisms we consider send each control construction of PIC, **out** and **box**, to a single control construction, $\mathbf{N}$, in IMGRAPH. But we want a richer class of morphisms —the so-called structural morphisms— namely ones that send a control construction to an (almost) arbitrary *subgraph* of nodes. In that case we need to work harder, the details of which are contained in the next few sections. (I discuss the subtleties of exactly which class of morphisms we conjecture it is tractable to handle in section 11 of the report on my future research plans.) The last section on classification puts IMGRAPH on familiar algebraic footing by sketching an argument that IMGRAPH is the quotient model of an AC.

3 IMGRAPH notational conventions

Before defining IMGRAPH formally, we introduce some notational infrastructure.

The extension (relative to some universe) of any function f is $\overline{f}$:

$$\overline{f}(\alpha) \stackrel{\text{def}}{=} \begin{cases} f(\alpha) & \text{if } \alpha \in \operatorname{dom} f \\ \alpha & \text{otherwise} \end{cases}$$

For any relation E and function f we define the mapping of f over E as follows:

$$\begin{aligned} E[f] &\stackrel{\text{def}}{=} \{(\overline{f}(\alpha), \overline{f}(\beta)) \mid (\alpha, \beta) \in E\} \\ E[f]^+ &\stackrel{\text{def}}{=} \{(\overline{f}(\alpha), \beta) \mid (\alpha, \beta) \in E\} \\ E[f]^- &\stackrel{\text{def}}{=} \{(\alpha, \overline{f}(\beta)) \mid (\alpha, \beta) \in E\} \end{aligned}$$

For any set S, we define the "filtering" of E by S as follows:

$$\begin{aligned} E \backslash S &\stackrel{\text{def}}{=} \{(\alpha, \beta) \in E \mid \alpha, \beta \notin S\} \\ E \backslash^+ S &\stackrel{\text{def}}{=} \{(\alpha, \beta) \in E \mid \alpha \notin S\} \\ E \backslash^- S &\stackrel{\text{def}}{=} \{(\alpha, \beta) \in E \mid \beta \notin S\} \end{aligned}$$

Both $[]$ and $\backslash$ associate to the left.

4 IMGRAPH statics

We now define the static structure of IMGRAPH.

4.1 Definition (externals) Let X be the set of names and let the natural numbers be the arity monoid for all control structures we will consider. IMGRAPH has a family of controls $\mathbf{N}_{m,n}$ of rank 1 with the arity inference rule:

$$\frac{a : k \to l}{\mathbf{N}_{m,n}(a) : m \to n} \quad .$$

■

4.2 Definition (internals) Points of IMGRAPH are of the form (V, E, m, n), where V is a set of vertices, E an edge relation:

$$E : V \cup X \cup m \to V \cup n$$

and m and n the number of source and sink ports. We quotient IMGRAPH by the obvious graph isomorphisms $\cong$, where:

$$(V, E, m, n) \cong (V', E', m, n)$$

iff there exists a bijection $f : V \to V'$ such that $E' = E[f]$. The operations on IMGRAPH are as follows:

$$
\begin{aligned}
(V,E,m,n)\otimes(U,F,k,l) &\stackrel{\text{def}}{=} (V+U,\;\; E' \,\cup\, F'[add_m]^{+}[add_n]^{-},\;\; m+k,\;\; n+l)\\
(V,E,m,n)\cdot(U,F,n,k) &\stackrel{\text{def}}{=} (V+U,\;\; E'\setminus^{-} n \,\cup\, F'\setminus^{+} n \,\cup\, E';F',\;\; m,\;\; k)\\
\mathbf{ab}_x(V,E,m,n) &\stackrel{\text{def}}{=} (V,\;\; E[add_1][x\mapsto 0] \,\cup\, \{(0,0)\},\;\; m+1,\;\; n+1)\\
\mathbf{N}_{m,n}(V,E,k,l) &\stackrel{\text{def}}{=} (\{v\},\;\; \{(x,v)\mid x\in \mathit{names}\,E\}\\
&\qquad\qquad \cup\, \{(i,v)\mid i\in m\} \,\cup\, \{(v,j)\mid j\in n\},\;\; m,\;\; n)\\
\langle x\rangle &\stackrel{\text{def}}{=} (\varnothing,\;\; \{(x,0)\},\;\; 0,\;\; 1)\\
\boldsymbol{\omega} &\stackrel{\text{def}}{=} (\varnothing,\varnothing,1,0)\\
\mathbf{id}_m &\stackrel{\text{def}}{=} (\varnothing,\;\; \{(i,i)\mid i\in m\},\;\; m,\;\; m)\\
\mathbf{p}_{m,n} &\stackrel{\text{def}}{=} (\varnothing,\;\; \{(i,i+n)\mid i\in m\}\\
&\qquad\qquad \cup\, \{(j+m,j)\mid j\in n\},\;\; m+n,\;\; m+n)
\end{aligned}
$$

Note: In the first two equations, $+$ denotes disjoint union. Also

$$E' \stackrel{\text{def}}{=} E[inl] \qquad F' \stackrel{\text{def}}{=} F[inr]$$

where *inl* and *inr* are the injections:

$$inl : V \;\to\; V+U \qquad inr : U \;\to\; V+U \quad .$$

The choice of vertex v in the definition of $\mathbf{N}$ is arbitrary. The definition of *names* is:

$$\mathit{names}\,E \stackrel{\text{def}}{=} \{x\in X \mid xE\} \quad .$$

Often, we shall write $\mathit{names}\,(V,E,m,n)$ for $\mathit{names}\,E$. ■

We now state three sanity results.

4.3 Lemma The above equations are well-defined with respect to the graph isomorphism $\cong$.

Proof In the last four equations, the set of vertices is empty, so the result holds trivially. The first four follow from simple properties of $[f]$. ■

Henceforth, we freely make use of graph isomorphisms and write $=$ for $\cong$. We also will neglect to worry about taking disjoint unions in the definitions of the CS operations as we can always assume apriori that the operand graphs have disjoint vertex sets.

4.4 Lemma IMGRAPH is a control structure.

Proof Trivial but tedious. ■

4.5 Theorem (names-surf equivalence) For any graph G,

$$\mathit{names}\ G = \mathit{surf}\ G \quad .$$

Proof Via two inclusions. Expand the definitions of **ab** and $\cdot$ in IMGRAPH. ■

5 IMGRAPH dynamics

In this section we define the "exploding" and "coalescing" dynamics of IMGRAPH and show that they preserve the dynamic discipline of source-separation.

5.1 Definition (exploding) Let (V, E, m, n) be in IMGRAPH and $v \in V$. Define k, l, $\{\alpha_i\}$, and $\{\beta_j\}$ such that:

$$\begin{aligned} \{\alpha_0, \dots, \alpha_{k-1}\} &= \{\alpha \mid (\alpha, v) \in E\} \qquad (\alpha_i \text{ distinct for } i \in k) \\ \{\beta_0, \dots, \beta_{l-1}\} &= \{\beta \mid (v, \beta) \in E\} \qquad (\beta_j \text{ distinct for } j \in l) \quad . \end{aligned}$$

Let (U, F, k, l) be in IMGRAPH with U and V disjoint. Then $\xrightarrow{\mathsf{Exp}}$ is the smallest preorder satisfying:

$$(V, E, m, n) \xrightarrow{\mathsf{Exp}} (V \cup U,\ \ E \setminus \{v\} \cup F[i \mapsto \alpha_i]^+[j \mapsto \beta_j]^-,\ \ m,\ \ n) \quad .$$

See Figure 11 for an example of the exploding reaction. ■

5.2 Definition (coalescing) Let (V, E, m, n) be in IMGRAPH, $u, v \in V$, $u \neq v$ be such that $(u, v) \in E$. Let vertex $w \notin V$. Then $\xrightarrow{\mathsf{Coa}}$ is the smallest preorder satisfying:

$$(V, E, m, n) \xrightarrow{\mathsf{Coa}} (V \setminus \{u, v\} \cup \{w\},\ \ E[u, v \mapsto w],\ \ m,\ \ n) \quad .$$

See Figure 10 for an example of the coalescing reaction. ■

5.3 Theorem The dynamic relations $\xrightarrow{\mathsf{Exp}}$ and $\xrightarrow{\mathsf{Coa}}$ are context-closed.
Proof Non-trivial and very detailed. ■

5.4 Lemma (node explosion) Let G and G' be in IMGRAPH, with $G' : m \to n$ Then

$$\mathbf{N}_{m,n}(G) \xrightarrow{\mathsf{Exp}} G' \quad .$$

Proof Trivial. ■

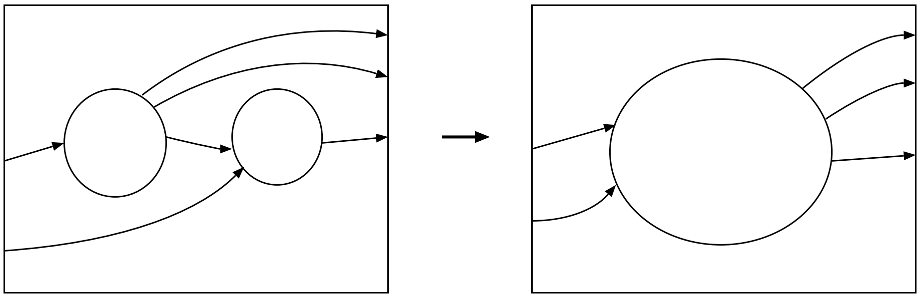

Figure 10: Coalescing reaction

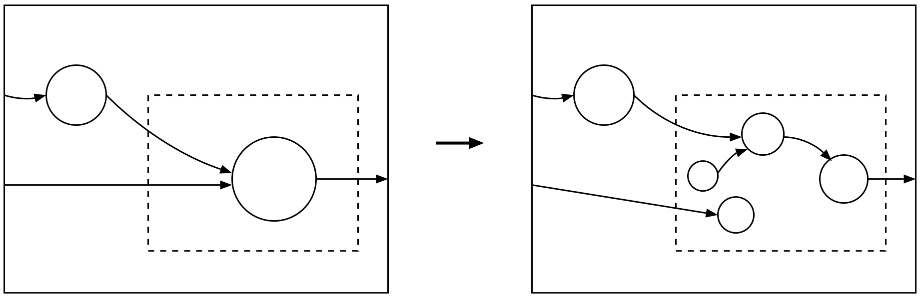

Figure 11: Exploding reaction

5.5 Definition (zig-zag path) Let $G = (V, E, m, n)$ be in $\mathsf{IMGRAPH}$ and $r, s \in m$. Then there is a *zig-zag path* from r to s iff there exist nodes $v_1, \ldots, v_k$ ($k \geq 1$) such that $(r, v_1) \in E$, $(v_k, s) \in E^{-1}$, and for $1 \leq i < k$,

$$(v_i, v_{i+1}) \in E \quad \text{or} \quad (v_i, v_{i+1}) \in E^{-1} \quad .$$

■

5.6 Definition (source-separateness, source-connectedness) Let $G = (V, E, m, n)$ be in $\mathsf{IMGRAPH}$. Then G is *source-connected* iff there exists a zig-zag path from r to s for some $r, s \in m$ ($r \neq s$). G is *source-separated* iff it is not source-connected. ■

5.7 Theorem (preservation of source-separateness) For arbitrary G, G' in $\mathsf{IMGRAPH}$, with G source-separated, if $G \xrightarrow{\mathsf{Exp}} G'$ or $G \xrightarrow{\mathsf{Coa}} G'$ then G' is source-separated.

Proof We sketch the $\xrightarrow{\mathsf{Exp}}$ case. Consider the contrapositive. Suppose $G = (V, E)$ and (U, F) are in $\mathsf{IMGRAPH}$ with $V \cap U = \varnothing$. Let $v \in V$ and let G' be the result of replacing v with (U, F) in G (see the definition of $\xrightarrow{\mathsf{Exp}}$). Suppose G' is source-connected, as witnessed by a zig-zag path from r to s in G':

$$\Gamma : r, v_1, \ldots, v_k, s$$

where $v_i \in V \setminus \{v\} \cup U$ for $1 \leq i \leq k$.

case $v_i \notin U$ for all i, $1 \leq i \leq k$: Then Γ is a also a path in G, so G is source-connected, as desired.

case $v_i \in U$ for some i, $1 \leq i \leq k$: Define:

$$\begin{array}{lcl} i_0 & = & \text{minimum } i,\ 1 \leq i \leq k, \text{ such that } v_{i_0} \in U \\ i_1 & = & \text{maximum } i,\ 1 \leq i \leq k, \text{ such that } v_{i_1} \in U \end{array}$$

Then

$$r, \underbrace{v_1, \ldots, v_{i_0-1}}_{\text{possibly empty}}, v, \underbrace{v_{i_1+1}, \ldots, v_k}_{\text{possibly empty}}, s$$

is a zig-zag path in G, as desired.

■

6 Structural morphisms

We now define formally the class of structural morphisms between control structures — the definition upon which we base the claim that LAMC "embeds" into IMGRAPH, but PIC does not. A *structural morphism* φ from control structure $\mathbb{D}$ over signature Σ to control structure $\mathbb{C}$ over signature Δ, where $\mathbb{D}$ and $\mathbb{C}$ have an identical arity monoid and name set, is a map that sends each $a : m \to n$ to $\varphi(a) : m \to n$ such that φ preserves the basic operations (tensor, composition, datum, etc.) and sends each control construction $K(\vec{a})$ $(K \in \Sigma)$ to $C[\,\varphi(\vec{a})\,]$. $C[\,\_\,]$ is a *structural context* over Δ, meaning $C[\,\_\,]$ has the form:

$$\begin{array}{rcll} C[\,\vec{t}\,] & ::= & C[\,\vec{b}\,] \otimes C[\,\vec{c}\,] & \text{for } \vec{t} \subseteq \vec{b} \cup \vec{c} \\ & | & C[\,\vec{b}\,] \cdot C[\,\vec{c}\,] & \text{as above} \\ & | & \mathbf{perm} \cdot C[\,\vec{d}\,] \cdot \mathbf{perm} & \text{for } \vec{t} \subseteq \vec{d} \\ & | & J(\vec{d}) & \text{as above and } J \in \Delta \end{array}$$

(Note that $\vec{b}, \vec{c}, \vec{d}$ are permuted arbitrarily. All we require is that $\vec{b} \cup \vec{c}$ exhausts $\vec{t}$, as does $\vec{d}$.) Structural morphisms satisfy some important properties — compositionality, for example — which are discussed in section 11 in the report on my future research plans. Structural contexts in IMGRAPH (where $\Delta = \{\mathbf{N}_{m,n}\}$) are particularly well-behaved in two important ways that we now address: (i) for every $t \in \vec{t}$, $C[\,\vec{t}\,]$ contains at least one node decorated by the surface of t; and (ii) every source port of $C[\,\vec{t}\,]$ has at least one edge coming out, hence the surface of $\langle\vec{x}\rangle \cdot C[\,\vec{t}\,]$ contains $\vec{x}$. The following theorems make these ideas precise.

6.1 Definition (source preservation) Let $G = (V, E, m, n)$ be in IMGRAPH. Then G is *source-preserving* iff for all $i \in m$, there exists β so that $(i, \beta) \in E$ — i.e., every source has at least one edge emanating from it. ■

6.2 Lemma (structural contexts are source-preserving) : Let $C[\vec{t}]$ be a structural context in IMGRAPH. Then $C[\vec{t}]$ is source-preserving.

Proof By induction on the structure of $C[\_]$. ■

6.3 Theorem (surface preservation) Let $C[\vec{t}]$ be a structural context in IMGRAPH. Then

$$\vec{x} \subseteq \mathit{names}\,(\langle\vec{x}\rangle \cdot C[\vec{t}]) \quad .$$

Proof By the previous result, $C[\vec{t}]$ is source-preserving, whence the result follows by expanding the IMGRAPH definition of $\langle x \rangle$ and $\cdot$. ■

6.4 Theorem (structural contexts box) Let $C[\_]$ be a structural context (assume of rank one, without loss of generality) in IMGRAPH and let G be in IMGRAPH. If $(V, E, m, n) = C[G]$ then there exists a $v \in V$ such that

$$\mathit{names}\,G \subseteq \{x \in X \mid (x, v) \in E\} \quad .$$

Proof By a long, but not difficult, induction on the structure of $C[\_]$. ■

7 Embedding LAMC and excluding PIC

We now present the two main results of this report. Their proofs, resting on all the preceeding infrastructure, go through very simply.

7.1 Theorem (LAMC embeds in IMGRAPH) There is a structural morphism $\varphi : \mathsf{LAMC} \to \mathsf{IMGRAPH}$.

Proof Because φ preserves the basic actions and because LAMC is an AC, so contains no junk, φ is completely determined by its action on $\boldsymbol{\lambda}$ and **ap**. Define φ recursively as follows:

$$\begin{array}{rcl} \varphi(\mathbf{ap} : m \to n) & = & \mathbf{N}_{m,n}(\mathbf{id}_0) \\ \varphi(\boldsymbol{\lambda}(a) : 0 \to 1) & = & \mathbf{N}_{0,1}(\varphi(a)) \end{array}$$

φ is visibly a static structural morphism. Each small-step reaction of LAMC is a combination of the exploding and coalescing dynamics, hence φ is also a dynamic structural morphism, as desired. ■

7.2 Theorem (excluding PIC) There is no dynamic structural morphism ψ : PIC $\to$ IMGRAPH.

Proof Deny for contradiction. Define

$$\begin{array}{rcl} t & \stackrel{\text{def}}{=} & \mathbf{out}_x \otimes (w)(\mathbf{box}_x((y)\mathbf{box}_z(\langle y,w\rangle \cdot \mathbf{out}))) \\ t' & \stackrel{\text{def}}{=} & (y,w)\mathbf{box}_z(\langle y,w\rangle \cdot \mathbf{out}) \quad . \end{array}$$

Then $t \xrightarrow{\mathsf{Pi}} t'$ — see Figure 4.

$\psi(t)$ **is source-separated:** Suppose ψ sends the top level **out** and **box** expressions in t to s and s' respectively. Then

$$\psi(t) = ((\mathbf{id} \otimes \langle x\rangle) \cdot s) \ \otimes \ (w)s'$$

Define

$$h \stackrel{\text{def}}{=} ((\mathbf{id} \otimes \langle x\rangle) \cdot \mathbf{N}_{2,0}(\mathbf{id}_0)) \ \otimes \ (w)\mathbf{N}_{1,0}(\mathbf{id}_0) \quad .$$

Then h is clearly source-separated in IMGRAPH. By lemma 5.4 ("node explosion"), we have the following reactions in IMGRAPH:

$$\begin{array}{rcl} \mathbf{N}_{2,0}(\mathbf{id}_0) & \longrightarrow & s \\ \mathbf{N}_{1,0}(\mathbf{id}_0) & \longrightarrow & s' \end{array}$$

hence $h \longrightarrow \psi(t)$ in IMGRAPH. Because IMGRAPH's dynamics preserve source-separation, $\psi(t)$ is source-separated, as desired.

$\psi(t')$ **is source-connected:** ψ is a structural morphism, so by theorem 6.3 ("surface preservation") the surface of $\psi(\langle y,w\rangle \cdot \mathbf{out})$ includes $\{y,w\}$. By theorem 6.4 ("structural contexts box"), $\psi(\mathbf{box}_z(\langle y,w\rangle \cdot \mathbf{out}))$ has a node decorated with $\{y,w\}$. Hence $\psi(t')$ is source-connected, as desired.

Therefore we cannot have $\psi(t) \longrightarrow \psi(t')$ in IMGRAPH, as desired. ∎

8 Addendum: IMGRAPH is the quotient of an AC

This section we sketch proofs that show IMGRAPH is a familiar algebraic structure, namely the quotient of an AC. As a byproduct, we get an inductively defined "reverse translation" that maps any standard IMGRAPH to an equivalent term expression.

8.1 Definition (standard IMGRAPHs) A graph (V,E,m,n) is *standard* iff (i) every sink port has exactly one predecessor, i.e., for all $j \in n$ there exists a unique α so that $(\alpha,j) \in E$; and (ii) E is acyclic in the vertices, i.e., there is no vertex sequence $v_0, v_1, \ldots, v_k = v_0$ (for $k \geq 1$) so that $(v_i, v_{i+1}) \in E$ for all $i \in k$. ∎

8.2 Lemma (term-expressible actions are standard) In IMGRAPH, any action expressible via the fundamental operations and the controls is standard.

Proof Induction on the structure of expression. ∎

Our next goal is to establish the converse of the above lemma.

8.3 Definition (node-molecules) A *node-molecule* μ is of the form

$$\mu \ ::= \ \langle \vec{x} \rangle \cdot \mathbf{N}_{k,l}(\mathbf{id}_0) \cdot (\vec{w}) \quad ,$$

where every element of $\vec{x}$ is distinct and every element of $\vec{w}$ is distinct. (Both are potentially empty.) A node-molecule is not a term, but a term "operator" because of the abstraction at the end. ∎

8.4 Definition (normal forms) A term is in *normal* form iff it is expressible as:

$$(\vec{y})\vec{\mu}\langle \vec{z} \rangle \quad .$$

∎

8.5 Theorem (reverse translation) Let $G = (V, E, 0, n)$ be a standard IMGRAPH. Then G has a normal form: there exist node-molecules $\mu_1, \ldots, \mu_{|V|}$ and a name vector $\vec{y} : m$ so that:

$$G = \mu_1 \ldots \mu_{|V|} \langle \vec{y} \rangle \quad .$$

(The abstraction at the front is empty because the source-arity of G is 0.)

Proof By induction on the size of V.

case $V = \varnothing$**:** The source arity of G is 0 so $E : X \to m$. G is standard, therefore every sink port has exactly one entering edge. Hence define names $x_0, \ldots, x_{m-1}$ so that:

$$E = \{(x_j, j) \mid j \in m\} \quad .$$

Then

$$G = \langle x_0, \ldots, x_{m-1} \rangle \quad ,$$

as desired.

case $V \neq \varnothing$**:** G is standard, hence acyclic, so there is a vertex $v \in V$ with no vertex predecessors (i.e., $(v', v) \notin E$ for all $v' \in V$). Since the source arity of G is 0, the only predecessors of v are names. Define k and (distinct) names $x_0, \ldots, x_{k-1}$ so that

$$\{x_0, \ldots, x_{k-1}\} = \{x \in X \mid (x, v) \in E\} \quad .$$

Define l and (distinct) $\beta_0, \ldots, \beta_{l-1}$ so that

$$\{\beta_0, \ldots, \beta_{l-1}\} = \{\beta \mid (v, \beta) \in E\} \quad .$$

Let $w_0, \ldots, w_{l-1}$ be distinct names not appearing in the names of G. Then

$$G = \langle x_0, \ldots, x_{k-1}\rangle \cdot \mathbf{N}_{k,l}(\mathbf{id}_0) \cdot (w_0, \ldots, w_{l-1})G'$$

where

$$G' = (V \setminus \{v\}, \ \ E \backslash \{v\} \cup \{(w_j, \beta_j) \mid j \in l\}, \ \ 0, \ \ n) \quad .$$

By the inductive hypothesis, $G' : 0 \to n$ has a normal form with $|V| - 1$ node-molecules, hence G has a normal form with $|V|$ node-molecules, as desired.

■

8.6 Theorem (standard graphs have normal forms) Let $G = (V, E, m, n)$ be an arbitrary standard graph. Then G has a normal form.

Proof Let names $x_0, \ldots, x_{m-1}$ be distinct and not appear in the names of G. Then

$$G = (\vec{x})(\langle\vec{x}\rangle \cdot G)$$

with $\langle\vec{x}\rangle \cdot G : 0 \to n$, which, by the previous theorem has a normal form, hence so does G. ■

We now define the AC ACIMGRAPH and show that the standard part of IMGRAPH is equivalent to ACIMGRAPH quotiented by a congruence.

8.7 Definition (ACIMGRAPH) ACIMGRAPH is the AC over the signature $\{\mathbf{N}_{m,n}\}$; $\sim$ is the smallest congruence over ACIMGRAPH satisfying:

$$\begin{array}{rcl}
\mathbf{N}_{m,n}(a) & \sim & \mathbf{M}_m(a) \cdot \underline{n} \\
\mathbf{perm} \cdot \mathbf{M}(a) & \sim & \mathbf{M}(a) \\
(\mathbf{copy} \otimes \mathbf{id}) \cdot \mathbf{M}_{m+1}(a) & \sim & \mathbf{M}_m(a) \\
\mathbf{M}(\mathbf{perm} \cdot a \cdot \mathbf{perm}) & \sim & \mathbf{M}(a) \\
\mathbf{M}(\boldsymbol{\omega} \otimes a) & \sim & \mathbf{M}(a) \\
\mathbf{M}(\mathbf{id} \otimes a) & \sim & \mathbf{M}(a) \\
\mathbf{M}(\mathbf{copy} \otimes a) & \sim & \mathbf{M}(a) \\
\mathbf{M}(a \cdot (\mathbf{N}(\mathbf{id}_0) \otimes \mathbf{id})) & \sim & \mathbf{M}(a) \\
\mathbf{M}_m(\langle x\rangle \otimes a) & \sim & (\langle x\rangle \otimes \mathbf{id}) \cdot \mathbf{M}_{m+1}(a) \quad .
\end{array}$$

Notation: we define the forms $\mathbf{M}_m$ and $\underline{n}$ as follows:

$$\begin{array}{rcl} \mathbf{M}_m(a) & \stackrel{\text{def}}{=} & \mathbf{N}_{m,1}(a) \\ \underline{n} & \stackrel{\text{def}}{=} & (x)\langle\underbrace{x,\ldots,x}_{n \text{ times}}\rangle \end{array}$$

The first three rules for $\sim$ state that the order and multiplicity of the incoming and outgoing edges of a node are irrelevant; the last six state that the internal structure of a boxed action is irrelevant, save for its surface. The reaction relation for ACIMGRAPH is $\longrightarrow$, the smallest preorder satisfying the following two rules (corresponding to the "exploding" and "coalescing" dynamics of IMGRAPH)

$$\begin{array}{rcl} \mathbf{N}_{m,n}(a) & \longrightarrow & t \\ (\mathbf{id}_k \otimes \mathbf{N}_{m,2}(a)) \cdot (\mathbf{M}_{k+1}(b)) & \longrightarrow & \mathbf{N}_{k+m,2}(a \otimes b) \end{array}$$

as well as LHS $\longrightarrow$ RHS and RHS $\longrightarrow$ LHS for each of the rules LHS $\sim$ RHS generating the congruence. ■

8.8 Theorem (soundness) For term a, b, if $a \sim b$ in ACIMGRAPH then $a = b$ in IMGRAPH.

Proof Expand the left-hand and right-hand sides of each congruence rule in IMGRAPH. ■

8.9 Theorem (completeness of normal forms) If normal forms $(\vec{x})\vec{\mu}\langle\vec{y}\rangle$ and $(\vec{w})\vec{\rho}\langle\vec{z}\rangle$ are equal in IMGRAPH then they are congruent in ACIMGRAPH.

Proof Suppose the hypothesis. Then clearly $\vec{\mu}$ and $\vec{\rho}$ are the same length. Induct on their length. ■

8.10 Theorem (reverse translation preserves $\sim$) If $(\vec{x})\vec{\mu}\langle\vec{y}\rangle$ is the reverse translation of a then $(\vec{x})\vec{\mu}\langle\vec{y}\rangle \sim a$.

Proof Induct on the term structure of a. (I have not proved this in detail yet, so this result is more a conjecture than a theorem.) ■

8.11 Theorem (completeness) For terms a, b, if $a = b$ in IMGRAPH then $a \sim b$ in ACIMGRAPH.

Proof Suppose the hypothesis. Let $(\vec{x})\vec{\mu}\langle\vec{y}\rangle$ be the reverse translation of a, and $(\vec{w})\vec{\rho}\langle\vec{z}\rangle$ the reverse translation of b. Then

$$a \sim (\vec{x})\vec{\mu}\langle\vec{y}\rangle \sim (\vec{w})\vec{\rho}\langle\vec{z}\rangle \sim b$$

as desired. ■

8.12 Theorem For terms a, b, $a \longrightarrow b$ in IMGRAPH iff $a \longrightarrow b$ in ACIMGRAPH.

Proof To be done. ■

9 Concluding thoughts and a summary of other research

Mobility is the first substantial classification problem we have attempted and it involved many false starts and revisions, mostly in our notion of structural morphism where we often considered definitions that were too week to allow the exclusion of PIC or too strong and restrictive to be satisfying. It is not yet clear how universal a hold we have on "immobility": we have not even begun to show that an AC encoding of CCS (which we believe is not mobile) embeds into IMGRAPH nor have we considered the status of the join-calculus (believed to be mobile?). On the other hand, IMGRAPH seems to be a robust object: it has a graph-theoretic edge-relation representation, as well as a presentation as the quotient of an AC. Furthermore, because it describes unnested graphs and the *connectedness* of nodes, IMGRAPH's static structure, equipped with varying dynamics, has the potential to describe a great variety of dynamic disciplines involving the flow and reachability of resources, as was done in the case of classifying mobility. Furthermore, as discussed in the summary of my future research plans, IMGRAPH may inspire us to explore important variations to the traditional AC.

In the next two paragraphs, I give a brief gloss on the other research I did this year.

During the winter term, I worked on designing a program to identify a redex in an AC term and then perform the rewrite specified by a reaction rule. Ole Jensen and I showed how his hypergraph representation of AC terms and his associated theory of graph-rewriting (Jensen, 1996) led directly to a clean and efficient mechanical implementation. I believe that the same technique will work well for many control structures, such as IMGRAPH (whose terms are after all *closer* to traditional flat graphs than an AC's are), and I wish to investigate this further.

For the past two terms, I collaborated with Adriana Compagnoni, Ole Jensen, and Peter Sewell on a project to encode a fragment of ML —with functions, references and updates, and exception handling— as a higher-order reflexive AC named FUNCREF (Compagnoni, 1992). Peter and I have now begun to look at the encoding of records and pattern matching, though this is still very preliminary work. The investigation has revealed several points: (i) The reflexive AC is perfect for encoding the store, a mapping from locations to values, where it is essential that we be able to represent arbitrarily cyclic structures. (ii) The automatic context-closure of every reaction rule allows for a very smooth presentation of the dynamics: for example the reaction of dereferencing a stored value makes no mention of the store as a whole, just the relevant redex (see Figure 12):

$$\mathbf{loc}(a) \cdot \mathbf{copy} \cdot (\mathbf{id} \otimes \mathbf{ref} \cdot \mathbf{deref}) \longrightarrow \mathbf{loc}(a) \otimes a$$

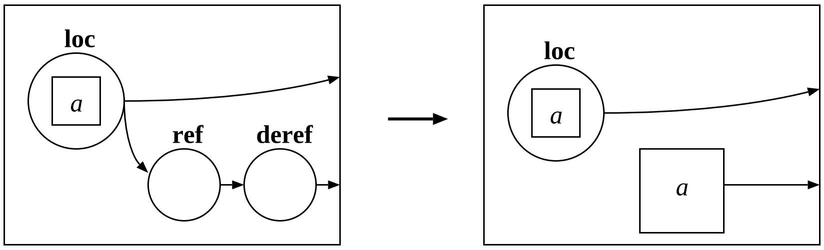

Figure 12: Dereferencing a location

(iii) The proofs of soundness and completeness with respect to a standard small-step operational semantics of ML are difficult, especially the justifications involving long manipulations of reflexive terms (the cost of having such simply stated reaction rules). We hope that the AC will be well-suited to encode a concurrent version of ML, for which the AC tensor is the perfect construction for combining processes that can intercommunicate. This is ongoing research.

Part II
Research Plans

10 Introduction to classification

As explained in the report on my first-year activities, the Action Calculus (AC) provides a uniform means of encoding the syntax and semantics of diverse operational systems, including the π-calculus, λ-calculus, CCS, fragments of ML, Petri-nets, and others (Milner, 1996). The term language for the AC is quotiented by a congruence relation which allows the terms to express a large class of nested graphs — and because of this we often reason informally about the AC by manipulating the graphs —canonical representations of the equivalence class of corresponding terms— themselves.[1] The semantics of an AC is encoded via a reduction relation, which is closed in all contexts except for in a control (which freezes its contents until they are freed). This convention is similar in concept, though more general, to the evaluation context style of presenting operational semantics and caters for a streamlined presentation of the semantics of a language: each dynamic rule need only describe the redex that "changes".

The Control Structures (CS) —models of AC— form the objects of a category with the maps that preserve the CS operations (tensor, composition, abstraction, etc.) as the *static* morphisms and the maps that also preserve the dynamic relation as the *dynamic* morphisms (*the* morphisms of the category). Each CS $\mathbb{C}$ *classifies* the rest: the universe of CSs is divided into those for which there exists a morphism to $\mathbb{C}$ and those for which there is not. If there is a morphism from $\mathbb{D}$ to $\mathbb{C}$ then, intuitively, if we observe the possible dynamic rewrites of $\mathbb{D}$ though "rose-coloured glasses" (i.e., we ignore differences collapsed by the morphism) the rewrites satisfy the dynamics of $\mathbb{C}$. And that is to say, $\mathbb{C}$ is a *type* for $\mathbb{D}$, i.e., $\mathbb{C}$ encodes a dynamic discipline satisfied by $\mathbb{D}$. My primary research goal is to investigate the theoretical foundations and practical applications of classification.

[1] Jensen (1996) formalises graphical reasoning about terms by encoding them as hypergraphs generated as solutions to categorical push-out diagrams.

11 Question inspired by classification

Any theory of classification is intimately dependent on our notion of morphism — and in defining that notion lies many open questions which I wish to investigate.

The first question to ask is: What restrictions do we need to impose on the image of a control construction under a morphism to allow for a good theory of classification? As I discussed in my first-year report, morphisms must be allowed to map between CSs over *different* signatures (for otherwise there always exists a classifier distinguishing between an CS and its "twin", identical except for the renaming of the signature — not a desirable property of classification). Consider then a map φ from a CS $\mathbb{D}$ over signature Σ to a CS $\mathbb{C}$ over Δ. It is meaningless to impose a condition that $\varphi(K(a)) = K(\varphi(a))$ for each $K \in \Sigma$ because K may not be in Δ. In general, φ sends $K(a)$ —a "node" containing a, at least if $\mathbb{D}$ is graph-like— to $C[\,\varphi(a)\,]$, where $C[\,\_\,]$ is an arbitrary context constructed from the basic operations and the controls in Δ: in other words $C[\,\varphi(a)\,]$ is a "subgraph" containing $\varphi(a)$, as least if $\mathbb{C}$ is graph-like. The danger is that $C[\,\varphi(a)\,]$ is so structureless —a mess of wires with no controls and no use of $\varphi(a)$, for example— that we cannot hope that $\mathbb{C}$ excludes $\mathbb{D}$, i.e., that there is no dynamic morphism from $\mathbb{D}$ to $\mathbb{C}$: in the extreme case, we can pick φ such that $\varphi(a) = \varphi(b)$ in $\mathbb{C}$ for all $a, b \in \mathbb{D}$ and then $\varphi(a) \longrightarrow \varphi(b)$ (in $\mathbb{C}$) by reflexivity of the dynamic relation $\longrightarrow$ (The Moral of the story is: Everything appears the same —and hence is indistinguishable— when looking through rose-coloured *opaque* glasses.)

So we ask: What are "reasonable","natural", "universal", "beautiful" conditions we can place on the behaviour (with respect to controls) of morphisms so that classifiers can successfully *exclude* certain CSs? For the $\mathsf{IMGRAPH}$ proofs, the following simple-minded syntactic restriction sufficed. If $\varphi : \mathbb{D} \to \mathbb{C}$ ($\mathbb{D}, \mathbb{C}$ with signatures Σ, Δ, respectively, as above) then for each $K \in \Sigma$, $\varphi(K(\vec{a})) = C[\,\varphi(\vec{a})\,]$ where $C[\,\vec{t}\,]$ is a context of the syntactic form:

$$
\begin{array}{rcll}
C[\,\vec{t}\,] & ::= & C[\,\vec{b}\,] \otimes C[\,\vec{c}\,] & \text{for } \vec{t} \subseteq \vec{b} \cup \vec{c} \\
& | & C[\,\vec{b}\,] \cdot C[\,\vec{c}\,] & \text{as above} \\
& | & \mathbf{perm} \cdot C[\,\vec{d}\,] \cdot \mathbf{perm} & \text{for } \vec{t} \subseteq \vec{d} \\
& | & J(\vec{d}) & \text{as above and } J \in \Delta
\end{array}
$$

(Note that $\vec{b}, \vec{c}, \vec{d}$ are permuted arbitrarily. All we require is that $\vec{b} \cup \vec{c}$ exhausts $\vec{t}$, as does $\vec{d}$.)

When we pick $\mathsf{IMGRAPH}$ for $\mathbb{C}$ and $\{\mathbf{N}_{m,n}\}$ for Δ the structural condition guarantees (i) $C[\,\varphi(a)\,]$ contains at least one node with surface containing *surf* a; and (ii) every source of the interface of $C[\,\varphi(a)\,]$ has at least one edge emanating from it, i.e., $\mathit{surf}\,(\langle\vec{x}\rangle \cdot C[\,\varphi(a)\,]) \supseteq \vec{x}$. What's more, $\mathit{surf}\ a = \mathit{surf}\ (\varphi(a))$ — which is so beautiful that we are bound to conclude that something like it, some semantic condition stated in terms of *surf* , will be a sufficient restriction on morphisms to allow IMGRAPH to exclude PIC, and, perhaps, be a suitable candidate for a "universal" definition of "classifying" morphisms — an issue I will pursue.

Structural morphisms (those satisfying the above syntactic condition) are compositional. Intuitively this is obvious: a morphism φ explodes each node construction $K(\vec{t})$ into a subgraph, boxing every $\varphi(t)$ for t in $\vec{t}$: say $\varphi(\vec{d})$ is the collection of boxed actions, with $\vec{t} \subseteq \vec{d}$. Applying a new morphism ψ to that subgraph does the same for each node ... resulting in a graph boxing every action in $\psi(\varphi(\vec{d}))$, hence every action in $\psi(\varphi(\vec{t}))$, as required. Because structural morphisms are compositional any classifier that excludes $\mathbb{D}$ excludes all CSs into which $\mathbb{D}$ embeds. Concretely, IMGRAPH excludes all systems PIC' richer than PIC, i.e., for which there exists a structural morphism $\mathsf{PIC} \to \mathsf{PIC}'$. This is a powerful aspect of classification and it gives us many results for free.

Compositionality of structural morphisms leads to some open questions about the algebraic properties of, for example, the subcategory of classifiers that all include and exclude a pair of CSs, say all include $\mathbb{D}$ and all exclude $\mathbb{D}'$. How do pushouts and pullbacks behave in this subcategory? The subcategory divides naturally into equivalence classes as follows: If there exists structural morphisms (not necessarily mutually inverse) both ways between classifiers $\mathbb{C}$ and $\mathbb{C}'$, $\mathbb{C} \to \mathbb{C}'$ and $\mathbb{C}' \to \mathbb{C}$, then, by compositionality, $\mathbb{C}$ and $\mathbb{C}'$ include and exclude exactly the same sets of CSs, so are "equivalent" in their power.

Classification forces us to reconsider issues involving the polymorphism of our arity system. We have as yet taken a simple minded view and thought of a control that can have many different arities as a set of monomorphic controls, one control for each separate arity. However, this choice leads to some difficult problems when trying to define the well-behaved morphisms for classification: a morphism could send different arity instances of a control to completely different terms, for example sending $K_{m,n} : m \to n$ to J if $m + n$ is even and to J' if it is odd. We want to be able to get a hold of the notion of a morphism behaving *uniformly* on the different arity instances of a control and to do this we need a truly polymorphic type theory for which we can then precisely say what it means for a morphism to "uniformly preserve the polymorphism". This is another issue that may I consider.

12 A brief diversion into the mirrored AC

Inspired by the ease with which IMGRAPH deals with much of the AC in strictly graph-theoretic terms, Robin's group has started to think about a variation on the AC —the "mirrored" AC— that is closer to pure graph theory and does not have the asymmetries of datum and abstraction. The idea is to consider a symmetric action structure and equip it with a symmetric datum operation $x : p \to p$ for each name x, along with operations "new" $\boldsymbol{\nu} : \epsilon \to p$ and "old"

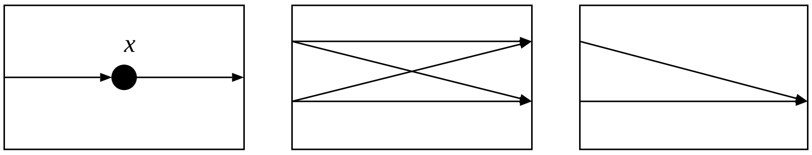

Figure 13: Wire merging arises naturally: x, $\mathbf{ab}_x x$, $(\mathbf{ab}_x x) \cdot (\boldsymbol{\omega} \otimes \mathbf{id})$

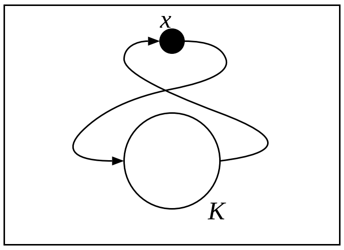

Figure 14: Reflexion arises naturally: $x \cdot K \cdot x$

$\boldsymbol{\omega} : p \to \epsilon$. We think of x as a named wire and, as such, we postulate:

$$\begin{aligned} x \cdot y &= y \cdot x \\ x \cdot x &= x \end{aligned}$$

along with analogies to many of the CS axioms, for example,

$$\begin{aligned} \mathbf{ab}_x y &= \mathbf{id} \otimes y, \qquad x \neq y \\ \mathbf{ab}_x \boldsymbol{\nu} &= \mathbf{id} \otimes \boldsymbol{\nu} \\ \mathbf{ab}_x \boldsymbol{\omega} &= \mathbf{id} \otimes \boldsymbol{\omega} \\ \boldsymbol{\nu} \cdot \boldsymbol{\omega} &= \mathbf{id}_\epsilon \\ \mathbf{ab}_x(\boldsymbol{\nu} \cdot x \cdot \boldsymbol{\omega}) &= \mathbf{id}_p \\ &\vdots \end{aligned}$$

The two most exciting benefits of a mirrored AC are that wire-merging arises naturally (see Figure 13) —which may turn out to provide a much cleaner and more powerful way of looking at nondeterminism that we currently have— and that a form of reflexion does as well (Figure 14). A great deal of work must be done to set up an appropriate axiomatisation, but this calculus looks potentially very attractive. I believe that it also comes with very natural molecular forms (Milner, 1996): each molecule is of the form $\langle Y_1, \ldots, Y_m \rangle K(Z_1, \ldots, Z_n)$ where the Y_i and Z_i are possibly empty *sets* of names, instead of individual ones.

13 Applications of classification?

The theoretical explorations of classification are, to some extent, intellectually unsatisfying *unless* we are able to use classification to gain insights into problems *outside* of the AC (and CSs). Arguing about which morphisms are most appropriate for classification without considering "real" systems to *classify* and "real" properties to *specify* —such as failure properties of distributed systems, security properties of protocols, etc.— leads to frustration because we lack concrete criterion for comparison. The AC is very young and as such we have few applications for it and therefore many of the problems we consider are, naturally, those *created* by our particular design choices in setting up the AC.

As I outlined in my first-year report, the classifier IMGRAPH excludes a subtle *intentional* property of process calculi, namely mobility, and thus separates the λ-calculus from the π-calculus. It is exactly this ability for classifiers to specify intentional properties —dynamic disciplines— that I wish pursue: my goal for the next year is to learn about "real" distributed systems, to try to encode their operational behaviour in the AC, and then to try to use classifiers and morphisms to specify "desirable" properties and to justify refinement and transformation steps.

I believe that classifiers *will* contribute to our understanding of other branches of computing. The success of type theory in presenting a mathematically tractable and eminently useful programming discipline is an inspiration: What types do for functional and imperative programming paradigms, I hope classifiers may eventually do for non-confluent, non-terminating, "interactive" systems. Of course, I may only have time to address a small piece of this problem in my thesis but, nonetheless, I think this is a topic worth pursuing and that there is much difficult and interesting work to be done.

Acknowledgements

I wish to thank my supervisor, Robin Milner, for the time and energy he has given me and for his inspirational criticisms and counterproposals he offered when I needed them most. I also wish to thank Adriana Compagnoni, Philippa Gardner, Andy Gordon, Martin Hyland, Ole Jensen, Benjamin Pierce, Andy Pitts, Peter Sewell — with whom I have had countless conversations and debates and who have offered me copious good advice.

Bibliography

(Barendregt, 1992) Barendregt, H. 'Lambda Calculi with Types' from Abramsky, S., Gabbay, D., and Maibaum T., *Handbook of Logic in Computer Science*, vol. 2. Oxford University Press.

(Compagnoni, 1996) Compagnoni, A., Jensen, O.H., Leifer, J., Sewell, P. 'An Action Calculus for Functional Programming'. In preparation.

(Jensen, 1996) Jensen, O.H. Forthcoming Ph.D. thesis. Cambridge Computing Laboratory.

(Kobayashi, 1995) Kobayashi, N., Pierce, B.C., Turner, D.N. 'Linearity and the Pi-Calculus'. Technical report, Department of Information Science, University of Tokyo and Computer Laboratory, University of Cambridge.

(Mifsud, 1994) Mifsud, A., Milner, R., Power, J. 'Control Structures'. LICS94.

(Milner, 1989) Milner, R. *Communication and Concurrency.* Prentice-Hall.

(Milner, 1992) Milner, R. and Parrow, J. and Walker, D. 'A Calculus of Mobile Processes'. *Information and Computation*, vol. 100, pp. 1–77.

(Milner, 1996) Milner, R. 'Calculi for Interaction'. *Acta Informatica.* (To appear.)

Appendix: The Axioms and Notational Conventions of the AC

This appendix summarises the fundamental definitions of the AC. Rather then give a syntactic presentation (Milner, 1996) I will show the semantic axiomatisation (by lifting out much of the salient portions of the discussion in Mifsud, 1994). An advantage of this axiomatisation is that it introduces uniformly both Control Structures and Action Calculi.

13.1 Definition (action structure statics) An *action structure* $\mathbb{A}$ is a strict monoidal category whose objects are taken from an arity monoid $(M, \otimes, \epsilon)$ and whose arrows are the *actions* (terms) of the structure. Every arity factorises "uniquely" into a product of *prime arities* and for each prime arity there is an infinite family of *names*, drawn from the name set X, of that arity. (Notation: "name x has prime arity p" is written $x : p$.)

The abstraction operator $\mathbf{ab}_x$ is an endofunctor satisfying the following arity condition:

$$\frac{a : m \rightarrow n}{\mathbf{ab}_x a : p \otimes m \rightarrow p \otimes n} (x : p)$$

We now summarise the axioms:

$$\begin{gathered}
a \otimes (b \otimes c) = (a \otimes b) \otimes c \\
a \cdot (b \cdot c) = (a \cdot b) \cdot c \\
a \otimes \mathbf{id}_\epsilon = a = \mathbf{id}_\epsilon \otimes a \\
a \cdot \mathbf{id} = a = \mathbf{id} \cdot a \\
\mathbf{id} \otimes \mathbf{id} = \mathbf{id} \\
\mathbf{ab}_x(a \cdot b) = \mathbf{ab}_x a \cdot \mathbf{ab}_x b \\
\mathbf{ab}_x \mathbf{id} = \mathbf{id}
\end{gathered}$$

Typically we shall leave out the subscripting of operations with their arities (such as with **id**) when the context removes the ambiguity. ■

13.2 Definition (action structure dynamics) A *dynamic* action structure is equipped with a *dynamic reaction relation* $\longrightarrow$, a preorder on the hom-set $\mathbb{A}(m,n)$ (the actions from m to n) for each pair of arities m and n. The reaction relation is context-closed: if $t \longrightarrow t'$ then $C[\,t\,] \longrightarrow C[\,t'\,]$ for any context $C[\,\_\,]$ in $\mathbb{A}$. The identity $\mathbf{id}_m$ is an extremal element for each $\mathbb{A}(m,m)$: if $\mathbf{id} \longrightarrow t$ then $t = \mathbf{id}$. ■

13.3 Definition (symmetric action structure) A *symmetric* action structure has a symmetry operation $\mathbf{p}_{m,n} : m \otimes n \;\rightarrow\; n \otimes m$ satisfying the following symmetry axioms:

$$\begin{gathered}
\mathbf{p}_{m,n} \cdot (b \otimes a) = (a \otimes b) \cdot \mathbf{p}_{k,l} \\
\mathbf{p}_{m,n} \cdot \mathbf{p}_{n,m} = \mathbf{id}_{m \otimes n} \\
(\mathbf{p}_{m,n} \otimes \mathbf{id}_k) \cdot (\mathbf{id}_n \otimes \mathbf{p}_{m,k}) = \mathbf{p}_{m,n \otimes k}
\end{gathered}$$

where $a : m \;\rightarrow\; k$ and $b : n \;\rightarrow\; l$. We postulate the relationship between symmetries and abstraction as follows:

$$\begin{gathered}
\mathbf{ab}_x\mathbf{p} = \mathbf{id} \otimes \mathbf{p} \\
\mathbf{ab}_x(\mathbf{ab}_x a) = \mathbf{id} \otimes \mathbf{ab}_x a \\
\mathbf{ab}_x(a \otimes \mathbf{id}) = \mathbf{ab}_x a \otimes \mathbf{id} \\
(\mathbf{p}_{p,q} \otimes \mathbf{id}) \cdot \mathbf{ab}_y\mathbf{ab}_x a = \mathbf{ab}_x\mathbf{ab}_y a \cdot (\mathbf{p}_{p,q} \otimes \mathbf{id}) \qquad (\text{for } x : p, y : q, x \neq y)
\end{gathered}$$

■

13.4 Definition (surface) The *surface* of an action a is written *surf* a and satisfies:

$$\mathit{surf}\ a \stackrel{\text{def}}{=} \{x \in X \mid \mathbf{ab}_x a \neq \mathbf{id} \otimes a\} \quad .$$

The surface is a semantic means of getting at the "free names" of an action — the names that abstraction "cares" about. ■

13.5 Lemma In any symmetric action structure:

$$\begin{gathered}
\mathit{surf}\ \mathbf{id} = \varnothing \\
\mathit{surf}\ \mathbf{p} = \varnothing \\
\mathit{surf}\ (a \cdot b) \subseteq \mathit{surf}\ a \cup \mathit{surf}\ b \\
\mathit{surf}\ (a \otimes b) \subseteq \mathit{surf}\ a \cup \mathit{surf}\ b \\
\mathit{surf}\ (\mathbf{ab}_x a) \subseteq \mathit{surf}\ a \setminus \{x\}
\end{gathered}$$

Equality holds in the last line within any CS (a stronger system than a symmetric action structure), which we are about to define. ■

13.6 Definition (control structure) A CS $\mathbb{C}$ over signature Σ is a symmetric action structure together with a datum operation $\langle x\rangle : \epsilon \rightarrow p$ for each name $x : p$ and each prime p; a discard operation $\boldsymbol{\omega} : p \rightarrow \epsilon$ for each prime p; and a control operation K for each $K \in \Sigma$. A control operation K of rank r takes r actions as arguments and satisfies an arity inference rule of the form:

$$\frac{\begin{array}{c} a_1 : m_1 \rightarrow n_1 \\ \vdots \\ a_r : m_r \rightarrow n_r \end{array}}{K(a_1, \ldots, a_r) : m \rightarrow n}\ \chi$$

with side condition χ. We then postulate the following axioms for $\mathbb{C}$:

$$\begin{array}{rcll}
\mathbf{ab}_x\langle y\rangle & = & \mathbf{id} \otimes \langle y\rangle & (x \neq y) \\
\mathbf{ab}_x\boldsymbol{\omega} & = & \mathbf{id} \otimes \boldsymbol{\omega} & \\
(x)\langle x\rangle & = & \mathbf{id} & \\
[{}^{x}\!/_{x}]a & = & a & \\
[{}^{y}\!/_{x}](\langle x\rangle \otimes \langle x\rangle) & = & \langle y\rangle \otimes \langle y\rangle & \\
[{}^{y}\!/_{x}]K(a_1, \ldots, a_r) & = & K([{}^{y}\!/_{x}]a_1, \ldots, [{}^{y}\!/_{x}]a_r) &
\end{array}$$

where we define:

$$\begin{array}{rcl}
(x)a & \stackrel{\text{def}}{=} & \mathbf{ab}_x a \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) \\
[{}^{y}\!/_{x}]a & \stackrel{\text{def}}{=} & (\langle y\rangle \otimes \mathbf{id}) \cdot (x)a \quad .
\end{array}$$

■

13.7 Definition (notational conventions) We adopt the following notational conventions:

$$\begin{array}{rcll}
\langle x_1, \ldots, x_n\rangle & \stackrel{\text{def}}{=} & \langle x_1\rangle \otimes \cdots \otimes \langle x_n\rangle & \\
\langle\rangle & \stackrel{\text{def}}{=} & \mathbf{id}_\epsilon & \\
(x_1, \ldots, x_n)a & \stackrel{\text{def}}{=} & (x_1) \cdots (x_n)a & \\
()a & \stackrel{\text{def}}{=} & a & \\
\mathbf{copy} & \stackrel{\text{def}}{=} & (x)(\langle x\rangle \otimes \langle x\rangle) & \\
\mathbf{perm} & \stackrel{\text{def}}{=} & \mathbf{id} \otimes \mathbf{p} \otimes \mathbf{id} & \\
\boldsymbol{\omega}_m & \stackrel{\text{def}}{=} & \boldsymbol{\omega}_{p_1} \otimes \cdots \otimes \boldsymbol{\omega}_{p_n} & (m = p_1 \otimes \cdots \otimes p_n)
\end{array}$$

And we conclude with some miscellaneous definitions about substitution. Above we defined the "semantic substitution of y for x (both names) in b", written $[{}^{y}\!/\!_{x}]b$. We extend the idea to include the "semantic substitution of an action a for a name x" in any CS that includes the $\boldsymbol{\lambda}$ control (see the definition of LAMC in section 2):

$$[{}^{a}\!/\!_{x}]b \stackrel{\text{def}}{=} (\boldsymbol{\lambda}(a) \otimes \mathbf{id}) \cdot (x)b$$

We denote the syntactic substitution of $a : \epsilon \to p$ for $x : p$ as $\{{}^{a}\!/\!_{x}\}$:

$$\begin{array}{rcl}
\{{}^{a}\!/\!_{x}\}\langle x\rangle & = & a \\
\{{}^{a}\!/\!_{x}\}\langle y\rangle & = & \langle y\rangle \qquad (y \neq x) \\
\{{}^{a}\!/\!_{x}\}(b \otimes c) & = & \{{}^{a}\!/\!_{x}\}b \otimes \{{}^{a}\!/\!_{x}\}c \\
 & \vdots & \\
\{{}^{a}\!/\!_{x}\}K(b_1, \ldots, b_n) & = & K(\{{}^{a}\!/\!_{x}\}b_1, \ldots, \{{}^{a}\!/\!_{x}\}b_n)
\end{array}$$

The above is only relevant in an AC — where there is no junk so we have recursion and induction principles. ■

We now come to the punch-line:

13.8 Definition (action calculus) $\mathbb{A}$ is an AC over signature Σ with reaction relation $\longrightarrow$ iff $\mathbb{A}$ is the smallest CS generated by the basic operations and the controls from Σ satisfying the axioms. In other words, $\mathbb{A}$ is the initial object in the category of all CSs over Σ and $\longrightarrow$. (See Mifsud (1994) for detailed justification.) ■