Action Calculi VI : Strong Normalisation at Higher-order

Robin Milner
Edinburgh, December 1993

DRAFT. PLEASE DO NOT REFERENCE.
COMMENTS ARE WELCOME.

1 Introduction

This paper has a single purpose: to prove that β-reduction is strongly normalising and confluent in higher-order action calculi [5]. The proof adapts the standard method for the simply-typed λ-calculus summarised by Barendregt [2] (Theorem A.2.3, p569).

The paper is self-contained, but does not provide motivation. In Section 2 the molecular forms of higher-order action calculi are introduced together with their algebraic operations: identity, composition, product, abstraction, data, discard and controls. The algebraic theory is not given, since only simple algebraic manipulations are needed in the paper; if necessary it can be found in [4, 5]. Section 3 deals with simple properties of reduction and proves weak confluence.

Section 4 establishes so-called *convergence sets* $C_{m\to n}$ of actions, for all arities m, n. The main result of the section is that all members of $C \stackrel{\text{def}}{=} \bigcup_{m,n} C_{m\to n}$ are strongly normalising. Finally, Section 5 defines the *molecular structure* of actions; then by induction on molecular structure it shows that $\sigma a \in C$ for every action a and every appropriate substitution σ. Since the identity substitution is appropriate, it follows that every action is in C and hence strongly normalising.

Another reduction relation which arises naturally in higher-order action calculi is the combination of β-reduction with explicit substitution, analogous to $\lambda\sigma$-calculus [1]. In the simply-typed version of the latter strong normalisation is open problem, though a restricted form of it has been solved by Ritter [6]. The problem is also open in higher-order action calculi.

2 Preliminaries

2.1 Arities and names Let $(M, \otimes, \epsilon)$ be a monoid. We say that $p \in M$ is *prime* if $p \neq \epsilon$, and $p = m_1 \otimes m_2$ implies that $m_1 = \epsilon$ or $m_2 = \epsilon$. We say that M *has primes* P if (1) P is the set of primes in M; (2) every $m \in M$ is a product of primes, and (3) whenever $p_1 \otimes \cdots \otimes p_r = q_1 \otimes \cdots \otimes q_s$ where each p_i and q_j is prime, then $r = s$ and $p_i = q_i$ $(1 \leq i \leq r)$. In other words, M has primes if each of its members is uniquely expressible as a product of primes.

Let M have primes P. If we extend M to $M^{\Rightarrow}$ by freely adding the binary constructor $\Rightarrow$, *exponentiation*, then $M^{\Rightarrow}$ has primes

$$P^{\Rightarrow} \stackrel{\text{def}}{=} P \cup \{m \Rightarrow n \mid m, n \in M^{\Rightarrow}\}\ .$$

We henceforth define the *arities* to be $M^{\Rightarrow}$ for some M which has primes. Thus the arities are just the familiar types of a typed λ-calculus with tupling. To simplify the exposition (though all our results hold in general) we shall assume that M has only a single prime, written "1", so that each prime of $M^{\Rightarrow}$ is either 1 or of form $m \Rightarrow n$. We shall also write ϵ as "0".

We assume a set X of *names*. We shall use u, v, w, x, y, z to range over X. We assume that each name $x \in X$ has a prime arity $p \in M^{\Rightarrow}$; we write $x : p$. We further assume that each prime arity is associated with infinitely many names. We shall write $\vec{x}$ for a vector $x_1 \cdots x_r$ of names; if $x_i : p_i$ for $1 \leq i \leq r$ then we write $\vec{x} : p_1 \otimes \cdots \otimes p_r$.

2.2 Actions and controls A *higher-order action calculus* over M is a action structure [3] whose objects are $M^{\Rightarrow}$ and whose arrows $a : m \to n$ are, for the purpose of this paper, represented by so-called *molecular forms*. Each specific higher-order action calculus $\mathsf{HAC}^{\beta}(\mathcal{K})$ is determined by a set $\mathcal{K}$ of *controls*, which are action constructors. Each control $K \in \mathcal{K}$ has an arity rule of the form

$$\frac{a_1 : m_1 \to n_1 \quad \cdots \quad a_r : m_r \to n_r}{K(a_1, \ldots, a_r) : m \to n} \ (\chi)$$

whose side-condition χ may constrain the values of r, m_i, n_i, m, n; this determines the arity of the construction $K(a_1, \ldots, a_r)$ in terms of the arities of its arguments. We often write the construction as $K\vec{a}$. in addition, there is a special control **ap** called *application* which takes no arguments; its arity rule is

$$\frac{}{\mathbf{ap} \ : (m \Rightarrow n) \otimes m \to n} \ .$$

2.3 Molecules, molecular forms and data We define the following by mutual recursion:

1. A *datum* d is either a name $x : p$ (p prime) or a *code* $\ulcorner a \urcorner : m \Rightarrow n$, where $a : m \to n$ is any action in molecular form. Thus every datum has prime arity. We use d, e, f, g to range over data.

2. An *action* $a : m \to n$ (in molecular form) is

$$(\vec{x})\, \mu_1, \ \ldots, \ \mu_r \ \langle \vec{d} \rangle \qquad (\vec{x} : m, \ \vec{d} : n)$$

   where each μ_i is a molecule. In a we call the bound names $\vec{x}$ its *imports*, the data $\vec{d}$ its *exports*, and the molecule sequence $\vec{\mu}$ its *body*. We shall use a, b, c to range over actions.

3. A *molecule* μ takes the form

$$\langle \vec{e} \rangle K \vec{b} (\vec{y}) \qquad (\vec{e} : k, \vec{y} : \ell)$$

   where $K \in \mathcal{K} \cup \{\mathbf{ap}\}$ and $K\vec{b} : k \to \ell$. We use $\kappa, \lambda, \mu, \nu$ to range over molecules. Note in particular that an **ap** molecule is of the form $\langle f \vec{e} \rangle \mathbf{ap} (\vec{y})$, with $f : m \Rightarrow n$, $\vec{e} : m$ and $\vec{y} : n$.

The molecule occurrences which make up the body of an action a are called *top-level* occurrences. Each such molecule $\mu = \langle\vec{e}\rangle K\vec{b}\,(\vec{y})$ binds the variables $\vec{y}$; the scope of this binding extends to the right of μ over the whole of a, including its exports.

We do not distinguish actions which only differ by alpha-conversion of imported names or of names bound by molecules; also we ignore any permutation of neighbouring molecules in a body if neither binds a name free in the other. We shall always assume that each name bound within an action a is distinct from any name occurring outside its scope (either within a or in the context in which a occurs).

An action $a : 0 \to n$ without imports, and $b : m \to 0$ without exports, will be written $\vec{\lambda}\,\langle\vec{d}\rangle$ and $(\vec{x})\,\vec{\mu}$ respectively.

2.4 Substitutions A substitution takes the form $\{\vec{d}/\vec{x}\}$, where $\vec{d}$ is a vector of data and has the same length as the vector $\vec{x}$ of distinct names. We shall use σ, τ to stand for substitutions. If $\sigma = \{\vec{d}/\vec{x}\}$ then by σa we mean the simultaneous replacement of x_i by d_i in a, assuming always that the bound names of a occur nowhere in σ. If $\tau = \{\vec{e}/\vec{y}\}$ then by $\tau\sigma$ we mean the composition of the two substitutions, i.e. $\{\vec{e}/\vec{y}, {}^{\tau\vec{d}}/\vec{x}\}$, again assuming that the substituted names $\vec{x}$ of σ occur nowhere in τ.

2.5 Operations We shall now define the action calculus operations over $\mathsf{HAC}^{\beta}(\mathcal{K})$. Suppose that $a = (\vec{x})\vec{\lambda}\langle\vec{d}\rangle$ and $b = (\vec{y})\vec{\mu}\langle\vec{e}\rangle$, with suitable arities, and that no name bound in either occurs in the other. The first four operations are those of an action structure:

$$
\begin{array}{rrcll}
\text{identity :} & \mathbf{id}_m & \stackrel{\text{def}}{=} & (\vec{x})\,\langle\vec{x}\rangle & (\vec{x} : m) \\
\text{composition :} & a \cdot b & \stackrel{\text{def}}{=} & (\vec{x})\,\vec{\lambda}, \sigma\vec{\mu}\,\langle\sigma\vec{e}\rangle & (\sigma = \{\vec{d}/\vec{y}\}) \\
\text{product :} & a \otimes b & \stackrel{\text{def}}{=} & (\vec{x}\vec{y})\,\vec{\lambda}, \vec{\mu}\,\langle\vec{d}\vec{e}\rangle & \\
\text{abstraction :} & \mathbf{ab}_x a & \stackrel{\text{def}}{=} & (x\vec{x})\,\vec{\lambda}\,\langle x\vec{d}\rangle & (x \text{ not bound in } a)\ .
\end{array}
$$

It is routine to check that these definitions satisfy the action structure axioms. The product $a \otimes b$ can be considered as simply the juxtaposition of the two actions; no order is dictated by the concatenation of the bodies, since the convention of disjoint bound names ensures that $\vec{\lambda}, \vec{\mu} = \vec{\mu}, \vec{\lambda}$ in this case. But in $a \cdot b$ the substitution σ may make $\vec{\lambda}$ bind some names in $\sigma\vec{\mu}$, so that $\vec{\lambda}, \sigma\vec{\mu} \neq \sigma\vec{\mu}, \vec{\lambda}$.

The remaining three operations, for which the axioms of higher-order action calculi [4] are satisfied, are as follows:

$$
\begin{array}{rrcll}
\text{data :} & \langle\vec{d}\rangle & \stackrel{\text{def}}{=} & (\,)\langle\vec{d}\rangle & \\
\text{discard :} & \boldsymbol{\omega} & \stackrel{\text{def}}{=} & (\vec{x})\langle\,\rangle & \\
\text{control :} & K\vec{a} & \stackrel{\text{def}}{=} & (\vec{x})\,\langle\vec{x}\rangle K\vec{a}\,(\vec{y})\,\langle\vec{y}\rangle & (K \in \mathcal{K} \cup \{\mathbf{ap}\})\ .
\end{array}
$$

We shall often write just $\ulcorner a \urcorner$ instead of $\langle\ulcorner a \urcorner\rangle$ for a code datum occurring as an action; similarly, we write $\ulcorner a \urcorner\langle\vec{d}\rangle$ instead of $\langle\ulcorner a \urcorner\vec{d}\rangle$.

It must be stressed that in the reduction system we describe below the syntactic entities which are reduced are molecular forms, not expressions formed using the operation symbols we have just defined. An expression of the latter sort, such as $(a \otimes \mathbf{id}) \cdot \mathbf{ap}$, is to be understood as denoting a certain molecular form.

2.6 β-reduction In $\mathsf{HAC}^\beta(\mathcal{K})$, the *reduction relation* $\longrightarrow_\beta$ is the smallest relation over molecular forms which satisfies the following conditions:

1. Let
$$a \;=\; (\vec{z})\,\vec{\lambda},\ \rho,\ \vec{\mu}\,\langle\vec{f}\rangle$$
where ρ is the molecule $\ulcorner c \urcorner \langle\vec{d}\rangle \mathbf{ap}\,(\vec{y})$. Then ρ is a *top-level redex* of a. Let $c = (\vec{x})\,\vec{\kappa}\,\langle\vec{e}\rangle$, $\sigma = \{\vec{d}/\vec{x}\}$ and $\tau = \{\sigma\vec{e}/\vec{y}\}$. Then $a \longrightarrow_\beta a'$, where
$$a' \;=\; (\vec{z})\,\vec{\lambda},\ \sigma\vec{\kappa},\ \tau\vec{\mu}\,\langle\tau\vec{f}\rangle\ .$$

2. If b occurs within a then any top-level redex of b is a *redex* of a; if $b \longrightarrow_\beta b'$ and a' results from performing this reduction within a then $a \longrightarrow_\beta a'$.

Note that a redex is (an occurrence of) an **ap** molecule whose first datum is a code, not an name. We shall use ρ, π to range over redexes; we must remember that they are molecule *occurrences*, but we shall not bother to designate occurrences formally.

We extend reduction to data and substitutions; we write $\vec{d} \longrightarrow_\beta \vec{d'}$ and $\{\vec{d}/\vec{x}\} \longrightarrow_\beta \{\vec{d'}/\vec{x}\}$ if $\vec{d'}$ results from reducing an action within $\vec{d}$.

The action a is said to be a *normal form*, or *in normal form*, if it contains no redex. If every reduction sequence of a reaches a normal form then we say that a is *strongly normalising*, or *convergent*. Otherwise a has an infinite reduction sequence, which we call a *divergence* of a, and we say a is *divergent*.

We shall usually omit the subscript β from $\longrightarrow_\beta$. If ρ is a redex of a then we write $a \longrightarrow^\rho a'$ for the reduction by ρ; if π is another redex of a whose residue by ρ (see 2.7 below) is a single molecule, then we may write $a \longrightarrow^\rho a' \longrightarrow^\pi a''$ for the successive reduction by ρ and then by (the residue of) π.

We shall say that two redexes are *independent* if they are syntactically disjoint, and neither lies within the scope of (the names bound by) the other.

2.7 Residues Let ρ be a redex in a, and let the molecule ν occur at top-level in a. We define the *residue of ν by (the reduction of) ρ*, distinguishing two cases:

1. ρ is top-level. Then, using the notation of the previous section, if $\nu \in \vec{\lambda}$ its residue is again ν; if $\nu = \rho$ its residue is the vector $\sigma\vec{\kappa}$; if $\nu \in \vec{\mu}$ its residue is $\tau\nu$.

2. ρ is not top-level. Then if ν contains ρ its residue is the molecule ν' which results from performing the reduction of ρ within ν; if ν does not contain ρ then its residue is ν itself.

Thus the residue of ν by ρ is a single molecule occurrence except in the case $\nu = \rho$, when it is a vector of molecules. By a natural extension, the residue of a vector $\vec{\nu}$ of top-level molecules by a reduction sequence is another vector $\vec{\nu'}$ of top-level molecules.

A more complex notion of residue can be defined for molecules not at top-level, but we shall not need it.

2.8 Symbolic convention Throughout the paper we shall adhere strictly to the following usage of symbols:

Italic			Greek
actions a, b, c		arities k, ℓ, m, n	molecules $\kappa, \lambda, \mu, \nu$
data d, e, f, g		prime arities p, q	redexes π, ρ
natural numbers h, i, j, r, s		names u, v, w, x, y, z	substitutions σ, τ .

3 Properties of reduction

This section is concerned with the more elementary properties of reduction. It includes lemmas which are needed for later proofs, and also a simple proof of weak confluence.

3.1 Lemma (Preservation of reduction) Let $a \longrightarrow a'$ and $\sigma \longrightarrow \sigma'$. Then for any action b and substitution τ

$$\begin{array}{llll}
\text{(1a)} & \tau a \longrightarrow \tau a' & \text{(1b)} & \sigma b \overset{*}{\longrightarrow} \sigma' b \\
\text{(2a)} & b \otimes a \longrightarrow b \otimes a' & \text{(2b)} & a \otimes b \longrightarrow a' \otimes b \\
\text{(3a)} & b \cdot a \longrightarrow b \cdot a' & \text{(3b)} & a \cdot b \overset{*}{\longrightarrow} a' \cdot b \ .
\end{array}$$

Proof Let ρ be the redex of a or of σ. In (1a) clearly $\tau\rho$ is a redex of τa; the result follows by simple properties of substitution. In (1b) the substitution makes a set of copies of ρ throughout b; these are independent redexes of σb, and $\sigma' b$ is reached by their reduction (in any order).

In (2) ρ is also a redex of $b \otimes a$ and of $a \otimes b$, and the result is obvious.

In (3a) the composition $b \cdot a$ exerts a substitution –say τ– upon a and the result follows as for (1a). In (3b) it is possible that ρ lies in the export vector of a; if so, then the composition $a \cdot b$ exerts a substitution –say σ– upon b, and $a' \cdot b$ exerts σ' upon b where $\sigma \longrightarrow \sigma'$. We then argue as for (1b). ■

3.2 Lemma (Redex promotion) Let ρ be a top-level redex of a. Assume that there is a reduction

$$a \overset{*}{\longrightarrow} b \overset{\rho}{\longrightarrow} b'$$

in which (the residue of) ρ is not reduced until the last step. Then for some a' there is a reduction

$$a \overset{\rho}{\longrightarrow} a' \overset{*}{\longrightarrow} b' \ .$$

Proof (Outline) It is clearly enough to consider the case where the initial reduction is one step, by another redex π, i.e.

$$a \overset{\pi}{\longrightarrow} b \overset{\rho}{\longrightarrow} b' \ .$$

Suppose that $a = (\vec{z})\,\vec{\lambda}, \rho, \vec{\mu}\,\langle\vec{f}\rangle$, and define a' by $a \overset{\rho}{\longrightarrow} a'$. There are several cases, depending on the relative positions of ρ and π. We present the important features of each case, leaving some details to be supplied by the reader.

1. ρ and π are independent. Then the reductions can be performed in either order without interaction, and we obtain $a \xrightarrow{\rho} a' \xrightarrow{\pi} b'$.

2. ρ is in the scope of π, i.e. $\pi \in \vec{\lambda}$. Then the residue of π by ρ is π itself; by analysis of the substitutions invoked by each order of reduction we again obtain $a \xrightarrow{\rho} a' \xrightarrow{\pi} b'$.

In all the remaining cases, we assume $\rho = \ulcorner c \urcorner \langle \vec{d} \rangle \mathbf{ap}\,(\vec{y})$ with $c = (\vec{x})\vec{\kappa}\langle \vec{e} \rangle$. We let $\sigma = \{\vec{d}/\vec{x}\}$ and $\tau = \{\sigma\vec{e}/\vec{y}\}$.

3. π lies in a code in $\vec{d}$. Let $\vec{d} \xrightarrow{\pi} \vec{d}'$ and $\sigma' = \{\vec{d}'/\vec{x}\}$; then $\sigma \xrightarrow{\pi} \sigma'$. But a' and b' differ only in that a' exerts the substitution σ while b' exerts σ', so by 3.1(1b) we obtain $a \xrightarrow{\rho} a' \xrightarrow{*} b'$.

4. π is, or lies within, a member of $\vec{\kappa}$. Then a' contains a corresponding redex $\sigma\pi$ within $\sigma\vec{\kappa}$, and we obtain $a \xrightarrow{\rho} a' \xrightarrow{\pi} b'$.

5. π lies in a code in $\vec{e}$. Let $\vec{e} \xrightarrow{\pi} \vec{e}'$ and $\tau' = \{\sigma\vec{e}'/\vec{y}\}$; then $\tau \xrightarrow{\pi} \tau'$ by 3.1(1a). But a' and b' differ only in that a' exerts τ while b' exerts τ', so by 3.1(1b) we obtain $a \xrightarrow{\rho} a' \xrightarrow{*} b'$.

6. π is in the scope of ρ, i.e. π lies within $\vec{\mu}$ or $\vec{f}$. Then a' correspondingly contains the redex $\tau\pi$, and by 3.1(1a) we again obtain $a \xrightarrow{\rho} a' \xrightarrow{\pi} b'$. ■

This lemma will be important in the proof of strong normalization later. But it also leads directly to the following

3.3 Proposition (Weak Confluence) Let $a \longrightarrow b$ and $a \longrightarrow c$. Then there exists a' which completes the following diagram:

$$\begin{array}{ccc} a & \longrightarrow & b \\ \downarrow & & \downarrow_{*} \\ c & \xrightarrow{*} & a' \end{array}$$

Proof Let $a \xrightarrow{\rho} b$ and $a \xrightarrow{\pi} c$ be the given reductions. If ρ and π lie within disjoint subactions of a then they are independent, so the result follows with $b \xrightarrow{\pi} a'$ and $c \xrightarrow{\rho} a'$. Otherwise there is some subaction of a, w.l.o.g. a itself, in which at least one of the redexes –say ρ– is top-level; now apply Lemma 3.2. ■

3.4 Lemma (Coding) Define the action context $\mathcal{D}[a] = b \cdot a \cdot c$ for any $a : m \to n$ and fixed $b : 0 \to m$, $c : n \to 0$. Then if a, b and $\mathcal{D}[a]$ converge, so does $\mathcal{D}[(\ulcorner a \urcorner \otimes \mathbf{id}_m) \cdot \mathbf{ap}]$.

Remark: One can check that $(\ulcorner a \urcorner \otimes \mathbf{id}_m) \cdot \mathbf{ap} \longrightarrow a$; intuitively, the lemma asserts that by postponing this reduction we cannot gain a divergence in the context $\mathcal{D}$.

Proof Let $b = \vec{\kappa}\langle \vec{d} \rangle$ and $c = (\vec{y})\vec{\mu}$, where $\vec{d} : m$, $\vec{y} : n$.

Now suppose $a = (\vec{x})\vec{\lambda}\langle\vec{e}\rangle$. Write $a^\dagger$ for $(\ulcorner a \urcorner \otimes \mathbf{id}_m) \cdot \mathbf{ap}$. Then we find that

$$\begin{array}{rcl}\mathcal{D}[a] & = & \vec{\kappa},\ \tau\vec{\lambda},\ \tau\vec{\mu} \qquad (\tau = \{\vec{d}/\vec{x}\}\{\vec{e}/\vec{y}\}) \\ \mathcal{D}[a^\dagger] & = & \vec{\kappa},\ \underbrace{\ulcorner a \urcorner\langle\vec{d}\rangle\mathbf{ap}\,(\vec{y})}_{\rho},\ \vec{\mu}\ .\end{array}$$

Note that $\tau\vec{\kappa} = \vec{\kappa}$ and $\tau\vec{\lambda} = \{\vec{d}/\vec{x}\}\vec{\lambda}$, since the names $\vec{x}, \vec{y}$ do not occur in $\vec{\kappa}$ and the names $\vec{x}$ do not occur in $\vec{\lambda}$.

Now suppose that $\mathcal{D}[a^\dagger]$ diverges. If (the residue of) ρ is ever reduced in this divergence then by Lemma 3.2 there is also a divergence which begins with ρ. But its first step is $\mathcal{D}[a^\dagger] \longrightarrow \mathcal{D}[a]$, contradicting the hypothesis that $\mathcal{D}[a]$ converges. Thus we may assume that the divergence of $\mathcal{D}[a^\dagger]$ never reduces ρ.

There must be a divergence either in the residue of $\vec{\kappa}, \rho$ or in the residue of $\vec{\kappa}, \vec{\mu}$. In the first case, the code $\ulcorner a \urcorner$ can contribute only finitely many steps since a converges; this implies that $b = \vec{\kappa}\langle\vec{d}\rangle$ diverges. In the second case, recalling that $\tau\vec{\kappa} = \vec{\kappa}$, we deduce by Lemma 3.1 that $\vec{\kappa}, \tau\vec{\mu}$ diverges, hence also $\mathcal{D}[a]$ diverges. Thus both cases contradict a hypothesis; hence $\mathcal{D}[a^\dagger]$ converges as required. ■

3.5 Lemma Let b converge and $b \cdot a$ diverge. Then for some b' with export vector $\vec{e}$, $b \longrightarrow^* b'$ and $\langle\vec{e}\rangle \cdot a$ diverges.

Proof It will be enough to deal with the case in which b has no imports. So let $b = \vec{\kappa}\langle\vec{d}\rangle$ and $a = (\vec{x})\vec{\mu}\langle\vec{f}\rangle$. Then

$$b \cdot a = \vec{\kappa},\ \sigma\vec{\mu}\,\langle\sigma\vec{f}\rangle \qquad (\sigma = \{\vec{d}/\vec{x}\})\ .$$

In the divergence of $b \cdot a$ there are only finitely many reductions in the residue of $\vec{\kappa}$, since b converges. By Lemma 3.2, $b \cdot a$ has a divergence in which all these reductions occur first. Let this reduction sequence leave $\vec{\lambda}$ as the residue of $\vec{\kappa}$ and induce a substitution τ, so that $b \longrightarrow^* \vec{\lambda}\,\langle\tau\vec{d}\rangle$ and

$$b \cdot a \ \longrightarrow^* \ \vec{\lambda},\ \tau\sigma\vec{\mu}\,\langle\tau\sigma\vec{f}\rangle$$

which then diverges with no further reductions within $\vec{\lambda}$. This means that $\tau\sigma\vec{\mu}\,\langle\tau\sigma\vec{f}\rangle$ diverges. But this is just $\langle\tau\vec{d}\rangle \cdot a$, so we are done if we take $\vec{e}$ to be $\tau\vec{d}$. ■

Finally, the following is an easy result:

3.6 Lemma Let a and b converge. Then also $a \otimes b$ and $a \cdot \boldsymbol{\omega}$ converge. ■

4 Convergence sets and groundings

In this section we first define certain sets $C_{m\to n}$ of actions by induction on the arity $m \to n$; then by means of a notion called *grounding context* we prove that every action in $C \stackrel{\text{def}}{=} \bigcup_{m,n} C_{m\to n}$ is convergent. The proof that $C_{m\to n}$ contains *all* actions $a : m \to n$, and hence that all actions converge, is completed in Section 5.

We shall extensively use induction on arities $m \to n$. This is based upon the well-founded partial order $\prec$ generated by the axioms

$$\begin{array}{rcll} 0 \to p_i & \prec & 0 \to p_1 \otimes \cdots \otimes p_r & (1 \leq i \leq r,\ r > 1) \\ m \to n & \prec & 0 \to m \Rightarrow n & \\ 0 \to m,\ 0 \to n & \prec & m \to n & (m \neq 0)\ . \end{array}$$

We shall also use the "projections" $\downarrow_i \stackrel{\text{def}}{=} (x_1 \cdots x_r)\langle x_i \rangle$ where $1 \leq i \leq r$ and $r > 1$.

4.1 Convergence sets The sets $C_{m\to n}$ are inductively defined to be the smallest such that

1. $a : 0 \to 0$ is in $C_{0\to 0}$ if it converges.
2. $a : 0 \to 1$ is in $C_{0\to 1}$ if it converges.
3. $a : 0 \to m \Rightarrow n$ is in $C_{0\to m\Rightarrow n}$ if $(a \otimes \mathbf{id}_m) \cdot \mathbf{ap}$ is in $C_{m\to n}$.
4. $a : 0 \to p_1 \otimes \cdots \otimes p_r$ $(r > 1)$ is in $C_{0\to p_1\otimes\cdots\otimes p_r}$ if $a \cdot \downarrow_i$ is in $C_{0\to p_i}$ for all $i \leq r$.
5. $a : m \to n$ is in $C_{m\to n}$ if whenever b is in $C_{0\to m}$ then $b \cdot a$ is in $C_{0\to n}$.

Due to the implication in clause 5 we cannot immediately prove that every $C_{m\to n}$ is non-empty, nor that its members converge. Hardly any simple properties of C –e.g. closure under composition, product or abstraction– are easy to prove.

In each of the clauses 3–5, the criterion for $a \in C_{m\to n}$ is that $\mathcal{D}[a] \in C_{k\to\ell}$, where $\mathcal{D}$ is some context and $k \to \ell \prec m \to n$. By iteration, the criterion for $a \in C_{m\to n}$ can be expressed as follows: that for all contexts $\mathcal{G}$ of a certain kind (dependent on $C_{k\to\ell}$ for smaller $k \to \ell$), $\mathcal{G}[a]$ converges. We now make this precise.

4.2 Groundings A *grounding for* $m \to n$ is a certain kind of context $\mathcal{G}$ such that $\mathcal{G}[a]$ is well-formed for $a : m \to n$. These contexts are the smallest set satisfying the following conditions, in which we use [ ] for the empty context and we write $\mathcal{G}$ *grounds* $m \to n$ to mean that $\mathcal{G}$ is a grounding for $m \to n$:

1. $[\,]$ grounds $0 \to 0$.
2. $[\,] \cdot \boldsymbol{\omega}$ grounds $0 \to 1$.
3. $\mathcal{G}[([\,] \otimes \mathbf{id}_m) \cdot \mathbf{ap}]$ grounds $0 \to m \Rightarrow n$ if $\mathcal{G}$ grounds $m \to n$.
4. $\mathcal{G}[[\,] \cdot \downarrow_i]$ grounds $0 \to p_1 \otimes \cdots \otimes p_r$ $(r > 1)$ if $i \leq r$ and $\mathcal{G}$ grounds $0 \to p_i$.
5. $\mathcal{G}[b \cdot [\,]]$ grounds $m \to n$ $(m \neq 0)$ if $b \in C_{0\to m}$ and $\mathcal{G}$ grounds $0 \to n$.

The following can be proved by a simple induction on arities:

4.3 Proposition (Grounding and convergence) $a \in C_{m\to n}$ iff, for all groundings $\mathcal{G}$ for $m \to n$, $\mathcal{G}[a]$ converges. ■

We shall now establish a useful property of groundings.

4.4 Lemma (Form of grounding) Every grounding $\mathcal{G}$ for $m \to n$ takes the form $b \cdot [\,] \cdot c$ where $b \in C_{0\to m}$ and $c : n \to 0$.

Proof By induction on arity. For the base cases $0 \to 0$ and $0 \to 1$ the result is obvious.

If $\mathcal{G}$ grounds $0 \to m \Rightarrow n$ then $\mathcal{G}[\,] = \mathcal{G}'[([\,] \otimes \mathbf{id}) \cdot \mathbf{ap}]$ where $\mathcal{G}'$ grounds $m \to n$. By induction we have $\mathcal{G}'[\,] = b' \cdot [\,] \cdot c'$, so

$$\begin{array}{rcll} \mathcal{G}[\,] & = & b' \cdot ([\,] \otimes \mathbf{id}_m) \cdot \mathbf{ap} \cdot c' & \quad (b' \in C_{0\to m}) \\ & = & [\,] \cdot (\mathbf{id}_{m\Rightarrow n} \otimes b') \cdot \mathbf{ap} \cdot c' & \end{array}$$

which is of the required form.

If $\mathcal{G}$ grounds $0 \to p_1 \otimes \cdots \otimes p_r$ $(r > 1)$ then $\mathcal{G}[\,] = \mathcal{G}'[[\,] \cdot \downarrow_i]$ where $\mathcal{G}'$ grounds $0 \to p_i$. By induction we have $\mathcal{G}'[\,] = b' \cdot [\,] \cdot c'$, so

$$\mathcal{G}[\,] \;=\; b' \cdot [\,] \cdot \downarrow_i \cdot c' \qquad (b' \in C_{0\to 0})$$

which is of the required form.

If $\mathcal{G}$ grounds $m \to n$ $(m \neq 0)$ then $\mathcal{G}[\,] = \mathcal{G}'[b \cdot [\,]]$ where $b \in C_{0\to m}$ and $\mathcal{G}'$ grounds $0 \to n$. By induction we have $\mathcal{G}'[\,] = b' \cdot [\,] \cdot c'$, so

$$\begin{array}{rcll} \mathcal{G}[\,] & = & b' \cdot b \cdot [\,] \cdot c' & \quad (b' \in C_{0\to 0}) \\ & = & b \cdot [\,] \cdot (b' \otimes c') & \end{array}$$

as required. ■

4.5 Elementary actions In the inductive proof below that all members of $C_{m\to n}$ converge, we need (for clause 5 only) that $C_{k\to\ell}$ is non-empty for some $k \to \ell \prec m \to n$. The only purpose of the following sets $E_{m\to n}$ of elementary actions (clearly non-empty) is to prove simultaneously in the induction that they are included in the $C_{m\to n}$, which must therefore be non-empty.

$$\begin{array}{rcll} E_{0\to 0} & \stackrel{\text{def}}{=} & \{\mathbf{id}_0\} & \\ E_{0\to 1} & \stackrel{\text{def}}{=} & \{\langle x\rangle \mid x : 1\} & \\ E_{0\to m\Rightarrow n} & \stackrel{\text{def}}{=} & \{\ulcorner a \urcorner \mid a \in E_{m\to n}\} & \\ E_{0\to p_1\otimes\cdots\otimes p_r} & \stackrel{\text{def}}{=} & \{a_1 \otimes \cdots \otimes a_r \mid \forall i \leq r.\ a_i \in E_{0\to p_i}\} & \quad (r > 1) \\ E_{m\to n} & \stackrel{\text{def}}{=} & \{(\vec{x})a \mid a \in E_{0\to n},\ \vec{x} : m \text{ not free in } a\} & \quad (m \neq 0)\ . \end{array}$$

We are now ready for the main result of this section.

4.6 Proposition (Convergence of C)

(1) If $a \in C_{m \to n}$ then a converges.

(2) $E_{m \to n} \subseteq C_{m \to n}$.

Proof Simultaneously by induction on $m \to n$. The base cases $0 \to 0$ and $0 \to 1$ are obvious, and we proceed to consider the induction step.

For (1), first suppose $a \in C_{0 \to m \Rightarrow n}$. Then $(a \otimes \mathbf{id}_m) \cdot \mathbf{ap} \in C_{m \to n}$. In molecular form, if $a = \vec{\mu}\,\langle f \rangle$ then $(a \otimes \mathbf{id}_m) \cdot \mathbf{ap} \in C_{m \to n} = (\vec{x})\,\vec{\mu},\, \langle f\vec{x} \rangle \mathbf{ap}\,(\vec{y})\,\langle \vec{y} \rangle$. Clearly a divergence of the former implies a divergence of the latter. But by induction the latter converges, hence a converges.

Next suppose $a \in C_{0 \to p_1 \otimes \cdots \otimes p_r}$ where $r > 1$. Let $a = \vec{\mu}\,\langle \vec{d} \rangle$. If a diverges, then clearly $a \cdot \downarrow_i$ also diverges for some i. but the latter is in $C_{0 \to p_i}$, hence converges by induction. Therefore a also converges.

Finally suppose $a \in C_{m \to n}$ where $m \neq 0$. Since by induction $C_{0 \to m}$ includes $E_{0 \to m}$ and is thus non-empty, pick any $b \in C_{0 \to m}$. Then $b \cdot a \in C_{0 \to n}$ converges by induction. From 3.1(3a) we deduce that a converges.

For (2), first suppose $a \in E_{0 \to m \Rightarrow n}$. Then $a = \ulcorner a' \urcorner$, $a' \in E_{m \to n}$. By induction, $a' \in C_{m \to n}$, hence by 4.3 $\mathcal{G}'[a']$ converges whenever $\mathcal{G}'$ grounds $m \to n$. But by 4.4 any such $\mathcal{G}'$ is of form $b \cdot [\,] \cdot c$ where $b \in C_{0 \to m}$. Now both b and a' converge by induction, so we can apply 3.4 to deduce that $\mathcal{G}'[(a \otimes \mathbf{id}_m) \cdot \mathbf{ap}]$ converges. But $\mathcal{G}'$ was arbitrary, hence we have shown that $\mathcal{G}[a]$ converges whenever $\mathcal{G}$ grounds $0 \to m \Rightarrow n$. It follows by 4.3 that a converges.

Next suppose $a \in E_{0 \to p_1 \otimes \cdots \otimes p_r}$ where $r > 1$. Then $a = a_1 \otimes \cdots \otimes a_r$, with $a_i \in E_{0 \to p_i}$ for each i, hence $a_i \in C_{0 \to p_i}$ and converges by induction. To show $a \in C$ we require for each i (by 4.3) that $\mathcal{G}[a \cdot \downarrow_i]$ converges whenever $\mathcal{G}$ grounds $0 \to p_i$. Now

$$a \cdot \downarrow_i = a_1 \cdot \boldsymbol{\omega} \otimes \cdots \otimes a_{i-1} \cdot \boldsymbol{\omega} \otimes a_i \otimes a_{i+1} \cdot \boldsymbol{\omega} \otimes \cdots \otimes a_r \cdot \boldsymbol{\omega}\ ,$$

and since $a_i \cdot \boldsymbol{\omega} : 0 \to 0$ we have with the help of 4.4

$$\mathcal{G}[a \cdot \downarrow_i] = a_1 \cdot \boldsymbol{\omega} \otimes \cdots \otimes \mathcal{G}[a_i] \otimes \cdots \otimes a_r \cdot \boldsymbol{\omega}\ .$$

But the right-hand side converges by 3.6, since all its factors converge, and we are done.

Finally suppose $a \in E_{m \to n}$ where $m \neq 0$. Then $a = (\vec{x})a'$, with $\vec{x}$ not free in a'. By induction, $a' \in C_{0 \to n}$. To prove $a \in C_{m \to n}$ we require by 4.3 that $\mathcal{G}[a]$ converges, where $\mathcal{G}$ is any grounding for $m \to n$. Now $\mathcal{G}[\,] = \mathcal{G}'[b \cdot [\,]]$ for some $b \in C_{0 \to m}$ and some grounding $\mathcal{G}'$ for $0 \to n$. So

$$\begin{array}{rcll} \mathcal{G}[a] & = & \mathcal{G}'[b \cdot a] & \\ & = & \mathcal{G}'[b \cdot \boldsymbol{\omega} \otimes a'] & \text{since } \vec{x} \text{ not free in } a' \\ & = & b \cdot \boldsymbol{\omega} \otimes \mathcal{G}'[a'] & \text{since } b \cdot \boldsymbol{\omega} : 0 \to 0\ . \end{array}$$

Now b converges by induction, $\mathcal{G}'[a']$ converges by 4.3, and so $\mathcal{G}[a]$ converges by 3.6 as required.

This completes the proof. ∎

5 Convergence and confluence of actions

The purpose of this section is to prove by induction on the structure of actions that every action is in C, and hence convergent by 4.6. In fact, just as in the standard proof [2] for the typed λ-calculus, we have to prove inductively a stronger property –that every C-instance of an action is in C. This notion is defined as follows: A substitution $\sigma = \{\vec{d}/\vec{x}\}$ *respects* C if $\langle\vec{d}\rangle \in C$; a C*-instance* of a is σa where σ respects C.

The proof will be done by induction on what we call the *molecular structure* of actions. This is defined in 5.2; informally, it reflects the way in which a molecular form may be built from right to left, starting with the export vector, prefixing the molecules one by one and ending by abstracting the imports.

First, knowing now that all members of C to converge, we prove a stronger form of Lemma 4.4 which we shall need.

5.1 Lemma (Convergent groundings)

(1) If $b : 0 \to m$ and $c : n \to 0$ converge then $(\mathbf{id}_{m\Rightarrow n} \otimes b) \cdot \mathbf{ap} \cdot c$ converges.

(2) Every grounding $\mathcal{G}$ for $m \to n$ takes the form $b \cdot [\;] \cdot c$ where $b \in C_{0\to m}$ and $c : n \to 0$ converges.

Proof For (1), let $b = \vec{\lambda}\,\langle\vec{d}\rangle$, $c = (\vec{y})\,\vec{\mu}$ and $a = (\mathbf{id}_{m\Rightarrow n} \otimes b) \cdot \mathbf{ap} \cdot c$. Then

$$a = (z)\,\vec{\lambda},\ \langle z\vec{d}\rangle\mathbf{ap}\,(\vec{y}),\ \vec{\mu}\ .$$

Now in reduction of a the residue of the **ap** molecule never becomes a redex; moreover, $\vec{\mu}$ contains no name bound by $\vec{\lambda}$. Hence if a diverges then either b or c must diverge, contra hypothesis.

For (2), we have to adapt the proof of Lemma 4.4 to obtain the extra condition that c converges. On inspecting the proof we see that this requires us to show that whenever b and c converge then so do $\downarrow_i \cdot\, c$, $b \otimes c$ and $(\mathbf{id}_{m\Rightarrow n} \otimes b) \cdot \mathbf{ap} \cdot c$. The first two are easy (see 3.6), and we have proved the third in (1) above. ■

We are now ready to define our constructions on molecular forms.

5.2 Molecular structure We consider a molecular form to be built from right to left. The main construction $\mu^\frown a$ is called *formation*; it prefixes a molecule μ to an action a with source arity 0, thus binding some of its variables. Formation can be expressed in terms of more elementary operations by $\langle\vec{d}\rangle K\vec{c}\,(\vec{y})^\frown a = \langle\vec{d}\rangle \cdot K\vec{c} \cdot (\vec{y})a$, but we adopt it since it represents molecular structure faithfully –and also since composition $(\cdot)$ is only needed in this restricted form. Indeed, the following constructions clearly build all actions:

$$\text{Data}: \quad \frac{x : p}{\langle x\rangle : 0 \to p} \qquad \frac{a : m \to n}{\ulcorner a \urcorner : 0 \to m \Rightarrow n}$$

$$\text{Tupling}: \quad \frac{\langle d_1\rangle : 0 \to p_1 \quad \cdots \quad \langle d_r\rangle : 0 \to p_r}{\langle d_1 \cdots d_r\rangle : 0 \to p_1 \otimes \cdots \otimes p_r}$$

$$\text{Formation}: \quad \frac{\langle\vec{d}\rangle : 0 \to m \qquad \vec{c} : \vec{m} \to \vec{n} \qquad \vec{y} : n \qquad a : 0 \to \ell}{\langle\vec{d}\rangle K \vec{c}(\vec{y})^\frown a : 0 \to \ell}$$

$$\text{Abstraction}: \quad \frac{\vec{x} : m \qquad a : 0 \to n}{(\vec{x})a : m \to n}\ .$$

In the formation rule $\vec{c} : \vec{m} \to \vec{n}$ means $c_i : m_1 \to n_i$ $(1 \le i \le r)$; we assume that r, m_i, n_i, m, n satisfy any side condition χ in the arity rule for $K \in \mathcal{K} \cup \{\mathbf{ap}\}$.

We now show that the data and tupling constructions preserve membership in C. The first part 5.3(1) concerning names is vital; recall that for any $x : 1$ we have $\langle x \rangle \in C$ by definition, but a proof is needed for the case $x : m \Rightarrow n$.

5.3 Proposition (Data and tupling)

(1) $\langle x \rangle \in C$.

(2) $a \in C$ iff $\ulcorner a \urcorner \in C$.

(3) $\langle d_1 \cdots d_r \rangle \in C$ iff $d_i \in C$ for each $i \le r$.

Proof For (1) the result holds by definition in the case $x : 1$, so suppose $x : m \Rightarrow n$. We require $\mathcal{G}[\langle x \rangle]$ convergent for every grounding $\mathcal{G}$ for $0 \to m \Rightarrow n$. By definition of grounding and by 5.1(2), this amounts to requiring that $a = (\langle x \rangle \otimes b) \cdot \mathbf{ap} \cdot c$ converges whenever $b : 0 \to m$ converges and $c : n \to 0$ converges. But a converges iff $(x)a$ converges, and in this case we find by the algebra of higher-order action calculi [5] that $(x)a = (\mathbf{id}_{m \Rightarrow n} \otimes b) \cdot \mathbf{ap} \cdot c$; the requirement is then assured by 5.1(1).

For (2), let $a : m \to n$ and $a^\dagger = (\ulcorner a \urcorner \otimes \mathbf{id}_m) \cdot \mathbf{ap}$. First observe that $\ulcorner a \urcorner \in C$ iff $a^\dagger \in C$, by Definition 4.1. Now let $\mathcal{G}$ ground $m \to n$, so that $\mathcal{G}[\,] = b \cdot [\,] \cdot c$ with $b : 0 \to m$ convergent and $c : n \to 0$. Hence, by the Coding lemma (3.4), if a and $\mathcal{G}[a]$ converge so does $\mathcal{G}[a^\dagger]$. For the other direction, since $a^\dagger \longrightarrow a$ we have $\mathcal{G}[a^\dagger] \longrightarrow^* \mathcal{G}[a]$ by 3.1; hence if $\mathcal{G}[a^\dagger]$ converges so does $\mathcal{G}[a]$. Since $\mathcal{G}$ was arbitrary, the result follows by 4.3.

Finally, (3) follows directly from $\langle d_1 \cdots d_r \rangle \cdot \downarrow_i = \langle d_i \rangle$ by the definition of C. ■

We next prove a few more closure properties of C, then the main result:

5.4 Lemma (Closure of C)

(1) If $a : 0 \to m$ and $a, b \in C$ then $a \cdot b \in C$.

(2) If $a \longrightarrow a'$ and $a \in C$ then $a' \in C$.

(3) If $a : 0 \to n$ and $\mu^\frown a \in C$ then $a \in C$.

Proof For (1), the result is immediate by definition of C if $m \ne 0$. Let $m = 0$ and $b : 0 \to n$, and let $\mathcal{G}$ ground $0 \to n$. Since $a : 0 \to 0$ we deduce from 4.4 that $\mathcal{G}[a \cdot b] = \mathcal{G}[a \otimes b] = a \otimes \mathcal{G}[b]$, which converges since $a, b \in C$ and by 3.6. But $\mathcal{G}$ was arbitrary, hence $a \cdot b \in C$.

For (2), we first deduce from 5.1(2) and 3.1(3) that $a \longrightarrow a'$ implies $\mathcal{G}[a] \overset{*}{\longrightarrow} \mathcal{G}[a']$ for any grounding. Thus if $\mathcal{G}[a]$ converges then $\mathcal{G}[a']$ converges; the result follows from 4.3.

For (3), by 4.4 every grounding $\mathcal{G}$ for $0 \to n$ is of the form $[\,] \cdot c$, so $\mathcal{G}[\mu^\frown a] = \mu^\frown \mathcal{G}[a]$. Thus if $\mathcal{G}[\mu^\frown a]$ converges then $\mathcal{G}[a]$ also converges; the rest is as in (2). ∎

5.5 Proposition (Convergence) For all actions a, if σ respects C then $\sigma a \in C$.

Proof We proceed by induction on the molecular structure of a as defined in 5.2, treating each of the constructions separately. If a is of the form $\langle x \rangle$, $\ulcorner b \urcorner$ or $\langle \vec{d} \rangle$ then the result follows directly from 5.3, together with the fact that substitution distributes over tupling and coding. For the remaining cases, assume that $\sigma = \{\vec{d}/\vec{x}\}$.

Abstraction: $a = (\vec{y})a'$ with $a : m \to n$, $\vec{y} : m$ and $a' : 0 \to n$. Suppose $\sigma a \notin C$; by 4.3 it follows that $\mathcal{G}[\sigma a]$ diverges for some grounding $\mathcal{G}$ for $m \to n$. We have

$$\begin{array}{rcll} \mathcal{G}[\,] & = & \mathcal{G}'[b \cdot [\,]] & \text{where } b \in C_{0 \to m},\ \mathcal{G}' \text{ grounds } 0 \to n \\ \mathcal{G}'[\,] & = & [\,] \cdot c & \text{with } c : m \to 0\ . \end{array}$$

So $b \cdot \sigma a \cdot c$ diverges, and hence by 3.5 for some $\vec{e}$

$$b \overset{*}{\longrightarrow} \vec{\mu}\,\langle \vec{e} \rangle \text{ and } \langle \vec{e} \rangle \cdot (\vec{y})\sigma a' \cdot c \text{ diverges.}$$

Now $\langle \vec{e} \rangle \cdot (\vec{y})\sigma a' = \sigma' a'$ where $\sigma' = \{\vec{e}/\vec{y}, \vec{d}/\vec{x}\}$, since by convention the names $\vec{y}$ do not occur in σ. Thus $\mathcal{G}'[\sigma' a']$ diverges. But $\langle \vec{e} \rangle \in C$ by Lemma 5.4, so σ' respects C and by induction $\sigma' a' \in C$. But this is a contradiction since $\mathcal{G}'$ is a grounding. Since this was deduced from the assumption that $\sigma a \notin C$, we have $\sigma a \in C$ as required.

Formation: $a = \lambda^\frown a'$ with $a : 0 \to n$ and $a' : 0 \to n$. We have to show that if $\mathcal{G}$ grounds $0 \to n$ then $\mathcal{G}[\sigma a]$ converges. Since $\mathcal{G}$ takes the form $[\,] \cdot c$ we immediately have

$$\mathcal{G}[\sigma a] = \sigma \lambda^\frown \mathcal{G}[\sigma a']\ .$$

Take first the case that $\lambda = \langle \vec{e} \rangle K \vec{b}\,(\vec{y})$ with $K \neq \mathbf{ap}$; then $\sigma\lambda = \langle \sigma\vec{e} \rangle K \sigma\vec{b}\,(\vec{y})$. Consider a reduction of $\sigma\lambda^\frown \mathcal{G}[\sigma a']$; at no time can a substitution be made for $\vec{y}$, so the reduction consists of independent reductions of $\langle \sigma\vec{e} \rangle$, of σb for each $b \in \vec{b}$, and of $\mathcal{G}[\sigma a']$. But by induction these all converge; hence also $\mathcal{G}[\sigma a]$ converges as required.

Now take the case that $\lambda = \langle g\vec{f} \rangle \mathbf{ap}\,(\vec{y})$, where $g, \vec{f}$ are data. If σg is a name, then $\sigma\lambda$ can never become a redex in a reduction of $\mathcal{G}[\sigma a]$, and the argument is as in the case $K \neq \mathbf{ap}$. Otherwise let $\sigma g = \ulcorner b \urcorner$; this is in C either because σ respects C (if g is a name) or by induction (if g is a code). By simple manipulation we find

$$\mathcal{G}[\sigma a] = \langle \sigma\vec{f} \rangle \cdot b^\dagger \cdot (\vec{y})\sigma a' \cdot c \text{ where } b^\dagger = (\ulcorner b \urcorner \otimes \mathbf{id}) \cdot \mathbf{ap}\ .$$

Now $\langle \sigma\vec{f} \rangle \in C$ by induction, and $b^\dagger \in C$ (since $\ulcorner b \urcorner \in C$) by Definition 4.1, hence $\langle \sigma\vec{f} \rangle \cdot b^\dagger \in C$ by Lemma 5.4(1). Suppose that $\mathcal{G}[\sigma a]$ diverges; then by 3.5 for some $\vec{e}$

$$\langle \sigma\vec{f} \rangle \cdot b^\dagger \overset{*}{\longrightarrow} \vec{\mu}\,\langle \vec{e} \rangle \text{ and } \vec{e} \cdot (\vec{y})\sigma a' \cdot c \text{ diverges.}$$

We then proceed to a contradiction exactly as in the abstraction case. Hence we conclude that $\mathcal{G}[\sigma a]$ converges; but $\mathcal{G}$ was arbitrarily chosen, hence $\sigma a \in C$ as required. ∎

5.6 Theorem (Convergence and Confluence) Every action is convergent; that is, the reduction relation $\longrightarrow_\beta$ is strongly normalising. Further, $\longrightarrow_\beta$ is also confluent; that is, Whenever $a \overset{*}{\longrightarrow} b$ and $a \overset{*}{\longrightarrow} c$ there exists a' which completes the following diagram:

$$\begin{array}{ccc} a & \overset{*}{\longrightarrow} & b \\ \downarrow_* & & \downarrow_* \\ c & \overset{*}{\longrightarrow} & a' \end{array}$$

Proof For the first part, we need only take σ to be the empty substitution in Proposition 5.5. The second part follows from Proposition 3.3 (Weak Confluence), by the standard result (see for example Barendregt [2], Proposition 3.1.25) that weak confluence and strong normalisation together imply strong confluence. ■

References

[1] M. Abadi, L. Cardelli, P.-L. Curien, J.-J. Lévy, *Explicit Substitutions*, ACM POPL Conference, San Francisco, 1990.

[2] Barendregt, H., **The Lambda Calculus**, North Holland 1981.

[3] Milner, R., *Action structures*, Research Report LFCS–92–249, Laboratory for Foundations of Computer Science, Edinburgh University, 1992.

[4] Milner, R., *Action calculi I: axioms and applications*, Computer Science Department, Edinburgh University, 1993. Also appeared as *Action calculi, or syntactic action structures*, Proceedings of MFCS '93, Lecture Notes in Computer Science Vol 711, Springer-Verlag, pp105–121, 1993.

[5] Milner, R., *Action calculi III: Higher-order action calculi*, Submitted for publication.

[6] Ritter, E., *Normalization for Typed Lambda Calculi with Explicit Substitution*, Paper presented at the 1993 Annual Conference of the European Association for Computer Science Logic, Swansea 1993.