Action Calculi IV : Molecular forms

Robin Milner
Edinburgh, November 1993

D R A F T 18/11/93. Comments are welcome.

1 Introduction

Action calculi were introduced in [3] as a subclass of action structures [2] comprising a variety of computational calculi. Each action calculus $\mathsf{AC}(\mathcal{K})$ is determined by a set $\mathcal{K}$ of *control* operators, together with their *reaction rules*.

The claim that action calculi provide a valuable framework rests upon a technical result: that each $\mathsf{AC}(\mathcal{K})$ can be presented either as a term algebra subject to a few simple equational axioms (independent of $\mathcal{K}$), or as an algebra of *molecular forms*. The latter play a dual role; thay act as *normal forms* for the term algebra under the axioms (and hence play an important part in the theory of action calculi), and they serve as a natural and modular mode of expression, even suitable for programming.

The technical result is that these two presentations are isomorphic. It was asserted in [3]; it is proved in detail here. For this purpose it is unnecessary to consider the dynamics (i.e. the reaction rules) associated with each specific set $\mathcal{K}$ of controls.

Higher-order action calculi $\mathsf{AC}^h(\mathcal{K})$ are introduced in [4]. They have a richer algebraic theory than action calculi; one axiom is strengthened and a further one is added. They also have a molecular presentation. Again, the isomorphism of the two presentations of $\mathsf{AC}^h(\mathcal{K})$, asserted in [4], is proved here in detail.

1.1 Outline The paper is entirely technical, and does not contribute either motivation or examples. It presumes some familiarity with [2, 3], and in Sections 8 and 9 with [4]. We begin in Section 1.2 with some preparatory definitions; then Sections 2 to 6 deal with the two presentations of $\mathsf{AC}(\mathcal{K})$ and their isomorphism. Section 7 briefly discusses the equational axioms. Section 8 defines *action calculi with coding* $\mathsf{AC}^c(\mathcal{K})$ –a halfway house to higher-order– and lifts the isomorphism to these. Finally, Section 9 deals with the two presentations of $\mathsf{AC}^h(\mathcal{K})$ and their isomorphism.

1.2 Arities, names and controls An action structure is a monoidal category with some extra structure; its morphisms are the *actions*, and its objects –a monoid $(M, \otimes, \epsilon)$– are called *arities*. We shall use k, ℓ, m, n to range over M.

For action calculi we assume that the monoid M *has primes*, i.e. that every arity factors uniquely into prime arities. More precisely, 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 $m_2 = \epsilon$. Then we assume that every $m \in M$ is a product of primes, and that if $p_1 \otimes \cdots \otimes p_r = q_1 \otimes \cdots \otimes q_s$ where each p_i and q_j is prime, then $r = s$ and $p_i = q_i$ $(1 \leq i \leq r)$.

We assume a *name-set* X. We shall use u, v, w, x, y, z to range over X. We assume that to each name $x \in X$ is associated a prime arity $p \in M$; 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 $\vec{x} : p_1 \otimes \cdots \otimes p_r$.

Each action calculus $\mathsf{AC}(\mathcal{K})$ is determined by a set $\mathcal{K}$ of *controls*, together with their reaction rules. Each control $K \in \mathcal{K}$ allows the construction of an action $K(a_1, \ldots, a_r)$ from actions $a_1, \ldots, a_r$, subject to an arity rule which may be presented in the form

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

where the 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 those of its arguments. We often write the construction as $K\vec{a}$; we may refer to it as a K-*action*, or $\mathcal{K}$-*action*, or *control action*.

2 Terms $\mathcal{T}(\mathcal{K})$

2.1 Terms The *terms* $\mathcal{T}(\mathcal{K})$ are generated from the *data* $\langle x \rangle : \epsilon \to p$ $(x \in X)$ and the *discard* $\boldsymbol{\omega} : p \to \epsilon$, using the *control* constructions $K\vec{t}$ $(K \in \mathcal{K})$ and the action structure operators. We shall use s, t to stand for terms. The abstractors $\mathbf{ab}_x$ $(x \in X)$ are the only operators which bind names; we denote the set of names free in t by $\mathrm{fn}\, t$.

2.2 Abbreviations We make the abbreviations

$$\begin{array}{rcll} \mathbf{ab}_{x_1 \cdots x_r} t & \stackrel{\text{def}}{=} & \mathbf{ab}_{x_1} \cdots \mathbf{ab}_{x_r} t & \\ \langle x_1 \cdots x_r \rangle & \stackrel{\text{def}}{=} & \langle x_1 \rangle \otimes \cdots \otimes \langle x_r \rangle \otimes \mathbf{id}_\epsilon & \qquad (r \geq 0)\ . \end{array}$$

2.3 Defined operators We define an alternative form of abstraction, and the family $\mathbf{i}_{km}$ of permutations:

$$\begin{array}{rrcll} \mathrm{D1}: & (x)t & \stackrel{\text{def}}{=} & \mathbf{ab}_x t \cdot (\boldsymbol{\omega} \otimes \mathbf{id}_n) & (t : m \to n) \\ & (\vec{x})t & \stackrel{\text{def}}{=} & (x_1) \cdots (x_r)t & (\vec{x} = x_1 \cdots x_r,\ r \geq 0) \\ \mathrm{D2}: & \mathbf{i}_{km} & \stackrel{\text{def}}{=} & (\vec{x}\vec{y})\langle \vec{y}\vec{x} \rangle & (\vec{x} : k, \vec{y} : m)\ . \end{array}$$

2.4 Axioms

The theory AC		
$\boldsymbol{\gamma}$:	$(x)t = \boldsymbol{\omega} \otimes t$	(x not free in t)
$\boldsymbol{\delta}$:	$(x)(\langle x \rangle \otimes \mathbf{id}_m) = \mathbf{id}_{p \otimes m}$	$(x : p)$
$\boldsymbol{\zeta}$:	$\mathbf{i}_{km} \cdot (t \otimes s) = (s \otimes t) \cdot \mathbf{i}_{ln}$	$(s : k \to l,\ t : m \to n)$
$\boldsymbol{\sigma}$:	$(\langle y \rangle \otimes \mathbf{id}_m) \cdot (x)t = \{y/x\}t$	$(t : m \to n)$.

We shall write $\mathsf{AC} \vdash s = t$ to mean that $s = t$ is provable in AC.

2.5 Proposition The following are provable in AC:

$$
\begin{array}{rll}
 & (1) \quad \langle x\rangle \cdot \boldsymbol{\omega} = \mathbf{id}_\epsilon & \\
 & (2) \quad (x)(s \cdot t) = (x)s \cdot t & (x \text{ not free in } t) \\
\boldsymbol{\alpha}: & (3) \quad (x)t = (y)\{y/x\}t & (y \text{ not free in } t) \\
 & (4) \quad (x)(t \otimes \mathbf{id}) = (x)t \otimes \mathbf{id} & \\
 & (5) \quad (x)(s \otimes t) = (x)s \otimes t & (x \text{ not free in } t) \\
 & (6) \quad \mathbf{ab}_x t = (x)(\langle x\rangle \otimes a) & \\
 & (7) \quad \mathbf{ab}_x t = \mathbf{id} \otimes t & (x \text{ not free in } t) \\
 & (8) \quad (\vec{x}\vec{y})t = (\mathbf{i}_{mn} \otimes \mathbf{id}) \cdot (\vec{y}\vec{x})t & (\vec{x} : m, \vec{y} : n) \\
 & (9) \quad (\vec{x})\langle\vec{x}\rangle = \mathbf{id}\ . &
\end{array}
$$

Proof

(1)
$$
\begin{array}{rcll}
\mathsf{AC} \vdash \langle x\rangle \cdot \boldsymbol{\omega} & = & \langle x\rangle \cdot (x)\mathbf{id}_\epsilon & \boldsymbol{\gamma} \\
 & = & \mathbf{id}_\epsilon\ . & \boldsymbol{\sigma}
\end{array}
$$

(2)
$$
\begin{array}{rcll}
\mathsf{AC} \vdash (x)(s \cdot t) & = & \mathbf{ab}_x s \cdot (x)t & \text{D1} \\
 & = & \mathbf{ab}_x s \cdot (\boldsymbol{\omega} \otimes t) & \boldsymbol{\gamma} \\
 & = & \mathbf{ab}_x s \cdot (\boldsymbol{\omega} \otimes \mathbf{id}) \cdot t & \\
 & = & (x)s \cdot t\ . & \text{D1}
\end{array}
$$

(3)
$$
\begin{array}{rcll}
\mathsf{AC} \vdash (y)\{y/x\}t & = & (y)((\langle y\rangle \otimes \mathbf{id}) \cdot (x)t) & \boldsymbol{\sigma} \\
 & = & (y)(\langle y\rangle \otimes \mathbf{id}) \cdot (x)t & (2) \\
 & = & (x)t\ . & \boldsymbol{\delta}
\end{array}
$$

(4)
$$
\begin{array}{rcll}
\mathsf{AC} \vdash (x)(t \otimes \mathbf{id}) & = & (x)(\{x/x\}t \otimes \mathbf{id}) & \\
 & = & (x)((\langle x\rangle \otimes \mathbf{id}) \cdot (x)t \otimes \mathbf{id}) & \boldsymbol{\sigma} \\
 & = & (x)((\langle x\rangle \otimes \mathbf{id} \otimes \mathbf{id}) \cdot ((x)t \otimes \mathbf{id})) & \\
 & = & (x)(\langle x\rangle \otimes \mathbf{id} \otimes \mathbf{id}) \cdot ((x)t \otimes \mathbf{id}) & (2) \\
 & = & (x)t \otimes \mathbf{id}\ . & \boldsymbol{\delta}
\end{array}
$$

(5)
$$
\begin{array}{rcll}
\mathsf{AC} \vdash (x)(s \otimes t) & = & (x)((s \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t)) & \\
 & = & (x)(s \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t) & (2) \\
 & = & ((x)s \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t) & (4) \\
 & = & (x)s \otimes t\ . &
\end{array}
$$

(6)
$$
\begin{array}{rcll}
\mathsf{AC} \vdash \mathbf{ab}_x t & = & \mathbf{ab}_x t \cdot (x)(\langle x\rangle \otimes \mathbf{id}) & \boldsymbol{\delta} \\
 & = & (x)(t \cdot (\langle x\rangle \otimes \mathbf{id})) & \text{D1} \\
 & = & (x)(\langle x\rangle \otimes t)\ . &
\end{array}
$$

(7)
$$
\begin{array}{rcll}
\mathsf{AC} \vdash \mathbf{ab}_x t & = & (x)(\langle x\rangle \otimes t) & (6) \\
 & = & (x)((\langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t)) & \\
 & = & (x)(\langle x\rangle \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes t) & (2) \\
 & = & \mathbf{id} \otimes t\ . & \boldsymbol{\delta}
\end{array}
$$

$$
\begin{array}{llll}
(8) & \mathsf{AC} \vdash (\mathbf{i}_{mn} \otimes \mathbf{id}) \cdot (\vec{y}\vec{x})t & = (\vec{x}\vec{y})(\langle\vec{y}\vec{x}\rangle \otimes \mathbf{id}) \cdot (\vec{y}\vec{x})t & \text{D2}, (5)^* \\
 & & = (\vec{x}\vec{y})((\langle\vec{y}\vec{x}\rangle \otimes \mathbf{id}) \cdot (\vec{y}\vec{x})t) & (2)^* \\
 & & = (\vec{x}\vec{y})t & \boldsymbol{\sigma}^*
\end{array}
$$

(9) Induction on length of $\vec{x}$. Basis true by definition. Step:

$$
\begin{array}{lll}
\mathsf{AC} \vdash (x\vec{y})\langle x\vec{y}\rangle & = (x)(\vec{y})(\langle\vec{y}\rangle \cdot (\langle x\rangle \otimes \mathbf{id})) & \\
 & = (x)((\vec{y})\langle\vec{y}\rangle \cdot (\langle x\rangle \otimes \mathbf{id})) & (2) \\
 & = (x)(\langle x\rangle \otimes \mathbf{id}) & \text{induction} \\
 & = \mathbf{id} & \boldsymbol{\delta}
\end{array}
$$

■

3 Molecular forms $\mathcal{M}(\mathcal{K})$

An *action* $a : m \to n$ in $\mathcal{M}(\mathcal{K})$ takes the form

$$(\vec{x})\ \mu_1 \cdots \mu_r\ \langle\vec{u}\rangle \qquad (\vec{x} : m, \vec{u} : n);$$

the sequence $\mu_1 \cdots \mu_r$ is called the *body*, and consists of molecules. We let a, b, c stand for actions. A *molecule* μ takes the form

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

where $K\vec{b} : k \to \ell$ is a $\mathcal{K}$-action. We shall let κ, λ, μ stand for molecules.

The *subactions* of an action a comprise a itself and the subactions of each action in a molecule of a. The *size* of a is the number of molecules of a and of its subactions. (We often need to argue by induction on size.) In the action $a = (\vec{x})\,\vec{\mu}\,\langle\vec{u}\rangle$ the *imported* names $\vec{x}$ (which must be distinct) are binding; the (distinct) names $\vec{y}$ of the molecule $\mu = \langle\vec{v}\rangle\ K\vec{b}\,(\vec{y})$ are also binding. The scope of each binding extends to its right, to the end of the smallest subaction containing it. Thus if μ is a molecule of a then the scope of $\vec{y}$ extends to the end of a, while the scope of any name bound within a member of $\vec{b}$ is confined to that subaction; however, any name free in $\vec{b}$ is also free in μ. We denote the set of names free in a by $\mathrm{fn}\,a$. Alpha conversion is allowed. The body of an action a is a *partial sequence*, i.e. the commutation $\lambda\mu = \mu\lambda$ is allowed if neither molecule binds a name occurring free in the other.

We now define the action structure operations over $\mathcal{M}(\mathcal{K})$; we also define data and discard as molecular forms and the control operations (for each $K \in \mathcal{K}$) over molecular forms. If $a = (\vec{u})\,\vec{\lambda}\,\langle\vec{v}\rangle$ and $b = (\vec{x})\,\vec{\mu}\,\langle\vec{y}\rangle$ then

$$
\begin{array}{rcll}
\mathbf{id}_k^{\mathcal{M}} & \stackrel{\text{def}}{=} & (\vec{x})\langle\vec{x}\rangle & (|\vec{x}| = m) \\
a \cdot^{\mathcal{M}} b & \stackrel{\text{def}}{=} & (\vec{u})\,\vec{\lambda}\,\sigma^{\mathcal{M}}\vec{\mu}\,\langle\sigma^{\mathcal{M}}\vec{y}\rangle & (\sigma = \{\vec{v}/\vec{x}\}) \\
a \otimes^{\mathcal{M}} b & \stackrel{\text{def}}{=} & (\vec{u}\vec{x})\vec{\lambda}\,\vec{\mu}\,\langle\vec{v}\vec{y}\rangle & \\
\mathbf{ab}_x^{\mathcal{M}} a & \stackrel{\text{def}}{=} & (x\vec{u})\,\vec{\lambda}\,\langle x\vec{v}\rangle & \\
\langle x\rangle^{\mathcal{M}} & \stackrel{\text{def}}{=} & (\,)\langle x\rangle & \\
\boldsymbol{\omega}^{\mathcal{M}} & \stackrel{\text{def}}{=} & (x)\langle\,\rangle & \\
K^{\mathcal{M}}(\vec{c}) & \stackrel{\text{def}}{=} & (\vec{x})\,\langle\vec{x}\rangle K\,\vec{c}(\vec{y})\,\langle\vec{y}\rangle & (\vec{x}, \vec{y} \text{ not free in } \vec{c})\,.
\end{array}
$$

Finally, we take substitutions $\{\vec{y}/\vec{x}\}^{\mathcal{M}}$ to be defined in the expected way (respecting bound names) over molecular forms. We shall be pedantic here, maintaining these superscripts ${}^{\mathcal{M}}$ for all operations upon molecular forms until we have proven the promised isomorphism; then we can thankfully drop them!

The following is straightforward:

3.1 Proposition $\mathcal{M}(\mathcal{K})$ is an action structure, and satisfies the axioms of AC. ∎

4 Translating molecular forms to terms

We define a translation $\widehat{(-)} : \mathcal{M}(\mathcal{K}) \rightarrow \mathcal{T}(\mathcal{K})/\mathsf{AC}$. The translation $\widehat{a}$ is defined inductively on a. First we translate each $a \in \mathcal{M}(\mathcal{K})$ to a term without assuming that a is subject to alpha-conversion and commutation of molecules. Then we show that, if a can be transformed to b by these operations, their translations are provably equal in AC. The translation $\widehat{(-)}$ is therefore well-defined from $\mathcal{M}(\mathcal{K})$ to $\mathcal{T}(\mathcal{K})/\mathsf{AC}$.

Note that some notations –e.g. $\langle x \rangle$– are used both within for molecular forms and for terms. This abuse of notation is always consistent with our translation; it sometimes makes the translation look like an identity function. The function $\widehat{(-)}$ is defined inductively as follows:

$$\widehat{(\vec{x})\,\vec{\mu}\,\langle\vec{y}\rangle} \stackrel{\text{def}}{=} \begin{cases} (\vec{x})\langle\vec{y}\rangle & \text{if } \vec{\mu} \text{ is empty} \\ (\vec{x})(\langle\vec{u}\rangle \cdot K\widehat{\vec{c}} \cdot \widehat{(\vec{x'})\vec{\mu'}\langle\vec{y}\rangle}) & \text{if } \vec{\mu} = \lambda\vec{\mu'}, \text{ where } \lambda = \langle\vec{u}\rangle K\vec{c}(\vec{x'}). \end{cases}$$

The following is easily proved by induction on a:

4.1 Proposition $\mathrm{fn}\,\widehat{a} = \mathrm{fn}\,a$. ∎

Using this, the following can be proved simultaneously by induction on a:

4.2 Proposition The translation $\widehat{(-)}$ preserves alpha-convertibility and substitution, i.e. (1) If b is an alpha-variant of a then $\boldsymbol{\alpha} \vdash \widehat{b} = \widehat{a}$; (2) If σ is a substitution $\{\vec{y}/\vec{x}\}$ then $\boldsymbol{\alpha} \vdash \widehat{\sigma^{\mathcal{M}} a} = \sigma\widehat{a}$. ∎

Now we complete the proof that

4.3 Proposition The translation $\widehat{(-)} : \mathcal{M}(\mathcal{K}) \rightarrow \mathcal{T}(\mathcal{K})/\mathsf{AC}$ is well-defined.

Proof We have already shown that the translation of alpha-variant molecular forms are provably equal terms (by $\boldsymbol{\alpha}$). It remains to show that the translations of forms which only differ by admissible commutation of molecules are provably equal in AC.

It will be enough to show that

$$\text{if } a_1 = \mu_1\,\mu_2\,a \text{ and } a_2 = \mu_2\,\mu_1\,a \text{ then } \mathsf{AC} \vdash \widehat{a_1} = \widehat{a_2}\ ,$$

provided μ_1 and μ_2 are commutable. So let

$$\mu_1 = \langle\vec{u_1}\rangle K_1 \vec{c_1}(\vec{v_1}) \text{ and } \mu_2 = \langle\vec{u_2}\rangle K_2 \vec{c_2}(\vec{v_2}) \ ,$$

where $\vec{v_1}$ are not free in μ_2 and $\vec{v_2}$ are not free in μ_1. Then

$$\begin{array}{rcll}
\mathsf{AC} \vdash \ \widehat{a_1} & = & \langle\vec{u_1}\rangle \cdot K_1\widehat{\vec{c_1}} \cdot (\vec{v_1})(\langle\vec{u_2}\rangle \cdot K_2\widehat{\vec{c_2}} \cdot (\vec{v_2})\widehat{a}) & \\
& = & \langle\vec{u_1}\rangle \cdot K_1\widehat{\vec{c_1}} \cdot \mathbf{ab}_{v_1}(\langle\vec{u_2}\rangle \cdot K_2\widehat{\vec{c_2}}) \cdot (\vec{v_1}\vec{v_2})\widehat{a} & \\
& = & \langle\vec{u_1}\rangle \cdot K_1\widehat{\vec{c_1}} \cdot (\mathbf{id} \otimes \langle\vec{u_2}\rangle \cdot K_2\widehat{\vec{c_2}}) \cdot (\vec{v_1}\vec{v_2})\widehat{a} & \text{by 2.5(7)} \\
& = & (\langle\vec{u_1}\rangle \cdot K_1\widehat{\vec{c_1}} \otimes \langle\vec{u_2}\rangle \cdot K_2\widehat{\vec{c_2}}) \cdot (\vec{v_1}\vec{v_2})\widehat{a} & \\
& = & (\langle\vec{u_2}\rangle \cdot K_2\widehat{\vec{c_2}} \otimes \langle\vec{u_1}\rangle \cdot K_1\widehat{\vec{c_1}}) \cdot \mathbf{i}_{m_2m_1} \cdot (\vec{v_1}\vec{v_2})\widehat{a} & \text{by } \boldsymbol{\zeta} \\
& = & (\langle\vec{u_2}\rangle \cdot K_2\widehat{\vec{c_2}} \otimes \langle\vec{u_1}\rangle \cdot K_1\widehat{\vec{c_1}}) \cdot (\vec{v_2}\vec{v_1})\widehat{a} & \text{by 2.5(8)} \\
& = & \widehat{a_2} & \text{by symmetry.}
\end{array}$$

■

It is worth noting that we have used all four axioms $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\sigma}$ in establishing this well-definedness.

From now on, when working in AC we shall often abbreviate $\mathsf{AC} \vdash t = u$ to $t = u$, giving reasons (e.g. the axiom used) where necessary.

4.4 Proposition $\widehat{(-)}$ preserves the action structure operations, i.e. (1) $\widehat{\mathbf{id}^{\mathcal{M}}} = \mathbf{id}$; (2) $\widehat{a \cdot^{\mathcal{M}} b} = \widehat{a} \cdot \widehat{b}$; (3) $\widehat{a \otimes^{\mathcal{M}} b} = \widehat{a} \otimes \widehat{b}$; (4) $\widehat{\mathbf{ab}_x^{\mathcal{M}} a} = \mathbf{ab}_x \widehat{a}$.

Proof

(1) By 2.5(9).

(2) Let $a = (\vec{u})\,\vec{\lambda}\,\langle\vec{x}\rangle$ and $b = (\vec{v})\,\vec{\mu}\,\langle\vec{y}\rangle$. We use induction on the length of $\vec{\lambda}$.

If $\vec{\lambda}$ is empty then $a = (\vec{u})\langle\vec{x}\rangle$ and $\widehat{a} = (\vec{u})\langle\vec{x}\rangle$. Also $\widehat{b} = (\vec{v})\widehat{b_0}$ where $b_0 = \vec{\mu}\,\langle\vec{y}\rangle$. Now $a \cdot^{\mathcal{M}} b = (\vec{u})\,\sigma^{\mathcal{M}}\vec{\mu}\,\langle\sigma\vec{y}\rangle$, where $\sigma = \{\vec{x}/\vec{v}\}$; so $\widehat{a \cdot^{\mathcal{M}} b} = (\vec{u})\,\widehat{\sigma^{\mathcal{M}} b_0} = (\vec{u})\,\sigma\widehat{b_0}$ by 4.2(2). On the other hand

$$\begin{array}{rcll}
\widehat{a} \cdot \widehat{b} & = & (\vec{u})\langle\vec{x}\rangle \cdot (\vec{v})\widehat{b_0} & \\
& = & (\vec{u})(\langle\vec{x}\rangle \cdot (\vec{v})\widehat{b_0}) & \text{by 2.5(2)} \\
& = & (\vec{u})\,\sigma\widehat{b_0} & \text{by } \boldsymbol{\sigma} \ .
\end{array}$$

If $\vec{\lambda} = \lambda\vec{\lambda}'$, where $\lambda = \langle\vec{w}\rangle K\,\vec{c}(\vec{u'})$, then let $a' = (\vec{u'})\,\vec{\lambda}'\,\langle\vec{x}\rangle$. Then

$$\begin{array}{rcll}
\widehat{a \cdot^{\mathcal{M}} b} & = & (\vec{u})(\langle\vec{w}\rangle \cdot K\widehat{\vec{c}} \cdot \widehat{a' \cdot^{\mathcal{M}} b}) & \text{by definition of } \widehat{(-)} \\
& = & (\vec{u})(\langle\vec{w}\rangle \cdot K\widehat{\vec{c}} \cdot \widehat{a'} \cdot \widehat{b}) & \text{by induction} \\
& = & (\vec{u})(\langle\vec{w}\rangle \cdot K\widehat{\vec{c}} \cdot \widehat{a'}) \cdot \widehat{b} & \text{by 2.5(2)} \\
& = & \widehat{a} \cdot \widehat{b} \ . &
\end{array}$$

(3) It is an easy induction to prove that $\widehat{a \otimes^{\mathcal{M}} \mathbf{id}^{\mathcal{M}}} = \widehat{a} \otimes \mathbf{id}$, and that $\widehat{\mathbf{id}^{\mathcal{M}} \otimes^{\mathcal{M}} b} = \mathbf{id} \otimes \widehat{b}$. Then we can use case (2) above to deduce $\widehat{a \otimes^{\mathcal{M}} b} = (\widehat{a} \otimes \mathbf{id}) \cdot (\mathbf{id} \otimes \widehat{b}) = \widehat{a} \otimes \widehat{b}$.

(4) We have from 3.1 and 2.5(6) that $\mathbf{ab}_x^{\mathcal{M}} a = (x)^{\mathcal{M}}(\langle x\rangle^{\mathcal{M}} \otimes^{\mathcal{M}} a)$. So

$$\begin{array}{rcll}
\widehat{\mathbf{ab}_x a} & = & (x)\widehat{\langle x\rangle^{\mathcal{M}} \otimes^{\mathcal{M}} a} & \text{by definition} \\
& = & (x)(\widehat{\langle x\rangle^{\mathcal{M}}} \otimes \widehat{a}) & \text{by case (3)} \\
& = & (x)(\langle x\rangle \otimes \widehat{a}) & \text{by 4.5 below} \\
& = & \mathbf{ab}_x \widehat{a} & \text{by 2.5(6).}
\end{array}$$

■

To complete the preservation properties, we state the following (whose proof is immediate):

4.5 Proposition $\widehat{(-)}$ preserves data, discard and control operations, i.e. $\widehat{\langle x\rangle^{\mathcal{M}}} = \langle x\rangle$, $\widehat{\boldsymbol{\omega}^{\mathcal{M}}} = \boldsymbol{\omega}$ and $\widehat{K^{\mathcal{M}}(\vec{c})} = K\widehat{\vec{c}}$. ■

5 Translating terms to molecular forms

We define a translation $[\![-]\!] : \mathcal{T}(\mathcal{K}) \to \mathcal{M}(\mathcal{K})$ inductively as follows:

$$\begin{array}{rcl@{\qquad\qquad}rcl}
[\![\mathbf{id}]\!] & \stackrel{\text{def}}{=} & (x)\langle x\rangle & [\![\langle x\rangle]\!] & \stackrel{\text{def}}{=} & \langle x\rangle^{\mathcal{M}} \\
[\![s \cdot t]\!] & \stackrel{\text{def}}{=} & [\![s]\!] \cdot^{\mathcal{M}} [\![t]\!] & [\![\boldsymbol{\omega}]\!] & \stackrel{\text{def}}{=} & \boldsymbol{\omega}^{\mathcal{M}} \\
[\![s \otimes t]\!] & \stackrel{\text{def}}{=} & [\![s]\!] \otimes^{\mathcal{M}} [\![t]\!] & [\![K(\vec{t})]\!] & \stackrel{\text{def}}{=} & K^{\mathcal{M}}[\![\vec{t}]\!]\ . \\
[\![\mathbf{ab}_x t]\!] & \stackrel{\text{def}}{=} & \mathbf{ab}_x^{\mathcal{M}}[\![t]\!] & & &
\end{array}$$

The following is an easy induction:

5.1 Proposition $\mathrm{fn}\,[\![t]\!] \subseteq \mathrm{fn}\,t$. ■

We cannot expect equality here; note that for terms we have $\mathrm{fn}\,(s \cdot t) = \mathrm{fn}\,s \cup \mathrm{fn}\,t$, but in general for molecular forms $\mathrm{fn}\,(a \cdot b) \neq \mathrm{fn}\,a \cup \mathrm{fn}\,b$; consider $a = \langle x\rangle^{\mathcal{M}}$, $b = \boldsymbol{\omega}^{\mathcal{M}}$.

The following is also an easy induction, since $[\![-]\!]$ is defined inductively on term-structure:

5.2 Proposition The translation $[\![-]\!]$ preserves the action structure operations, data, discard and substitution. ■

We now come to the important property of $[\![-]\!]$:

5.3 Proposition The translation $[\![-]\!]$ is a well-defined function from $\mathcal{T}(\mathcal{K})/\mathsf{AC}$ to $\mathcal{M}(\mathcal{K})$; that is, if $\mathsf{AC} \vdash s = t$ then $[\![s]\!] = [\![t]\!]$.

Proof It is enough to prove the result when $\vdash s = t$ is an instance of a basic axiom of action structures or one of the AC axioms. In each case the result is straightforward because $[\![-]\!]$ is defined inductively on term structure and because molecular forms also satisfy all the axioms (Proposition 3.1). In the case of $\boldsymbol{\sigma}$, one also needs $[\![\{y/x\}t]\!] = \{y/x\}^{\mathcal{M}}[\![t]\!]$ (Proposition 5.2). Finally, the case for $\boldsymbol{\gamma} \vdash (x)t = \boldsymbol{\omega} \otimes t$ ($x \notin \mathrm{fn}\, t$) works just because $\mathrm{fn}\,[\![t]\!] \subseteq \mathrm{fn}\, t$ (Proposition 5.1). For since also $x \notin \mathrm{fn}\,[\![t]\!]$, we have

$$\begin{array}{rcll} [\![(x)t]\!] & = & (x)^{\mathcal{M}}[\![t]\!] & \text{by definition} \\ & = & \boldsymbol{\omega}^{\mathcal{M}} \otimes^{\mathcal{M}} [\![t]\!] & \text{by Props 3.1, 5.1} \\ & = & [\![\boldsymbol{\omega} \otimes t]\!] & \text{by definition.} \end{array}$$

■

6 The action calculus AC($\mathcal{K}$)

We are now ready to prove that the molecular forms really are normal forms for the theory AC. We have already established that the two translations $\widehat{(-)}$ and $[\![-]\!]$ are well-defined between AC($\mathcal{K}$) and $\mathcal{T}(\mathcal{K})$/AC. We next show that they are mutually inverse:

6.1 Proposition For all terms t, $\mathsf{AC} \vdash \widehat{[\![t]\!]} = t$.

Proof Straightforward, since $[\![-]\!]$ is defined inductively via the operations, and $\widehat{(-)}$ preserves all of them (Propositions 4.4, 4.5). ■

6.2 Proposition For all molecular forms a, $[\![\widehat{a}]\!] = a$.

Proof We proceed inductively on the size of a. The base case is easy. Now suppose $a = (\vec{x})\,\lambda\vec{\lambda'}\,\langle\vec{y}\rangle$. If $\lambda = \langle\vec{z}\rangle K\vec{c}(\vec{x'})$ then

$$\widehat{a} \;=\; (\vec{x})(\langle\vec{z}\rangle \cdot K\widehat{\vec{c}} \cdot \widehat{a'}) \qquad \text{where } a' = (\vec{x'})\,\vec{\lambda'}\,\langle\vec{y}\rangle\,.$$

So we have

$$\begin{array}{rcll} [\![\widehat{a}]\!] & = & (\vec{x})^{\mathcal{M}}(\langle\vec{z}\rangle^{\mathcal{M}} \cdot^{\mathcal{M}} K^{\mathcal{M}}[\![\widehat{\vec{c}}]\!] \cdot^{\mathcal{M}} [\![\widehat{a'}]\!]) & \\ & = & (\vec{x})^{\mathcal{M}}(\langle\vec{z}\rangle^{\mathcal{M}} \cdot^{\mathcal{M}} K^{\mathcal{M}}\vec{c} \cdot^{\mathcal{M}} a') & \text{by induction} \\ & = & a & \text{by calculation.} \end{array}$$

■

Finally, with Propositions 4.2–4.5, 5.2–5.3 and 6.1–6.2 we have established

6.3 Theorem The translations $\widehat{(-)}$ and $[\![-]\!]$ between $\mathcal{M}(\mathcal{K})$ and $\mathcal{T}(\mathcal{K})$/AC constitute a static isomorphism of action structures, which moreover preserves the control operations $K \in \mathcal{K}$, the data $\langle x\rangle$, the discard $\boldsymbol{\omega}$ and the substitutions $\{\vec{y}/\vec{x}\}$. ■

This static isomorphism may be extended to a dynamic one by defining a reaction relation $\searrow$ either on $\mathcal{M}(\mathcal{K})$ or on $\mathcal{T}(\mathcal{K})$ by reaction rules, and transferring it to the other by use of the static isomorphism.

We shall refer to the isomorphic structures $\mathcal{M}(\mathcal{K})$ and $\mathcal{T}(\mathcal{K})$/AC indifferently as AC($\mathcal{K}$), the *action calculus over* $\mathcal{K}$. At this point we are justified in omitting the superscripts $^{\mathcal{M}}$ from the operations upon molecular forms.

7 Discussion of Axioms

The axioms AC originally given in [3] were six in number. They included $\boldsymbol{\alpha}$ which has now been derived as 2.5(3), and $\boldsymbol{\epsilon}$ which has been derived in more general form as 2.5(9); there was also a different $\boldsymbol{\delta}$, now derived as 2.5(4).

The present four –γ, $\boldsymbol{\delta}$, $\boldsymbol{\zeta}$ and $\boldsymbol{\sigma}$– appear to be independent. We shall not attempt to prove this for each of them here. But note that γ is the only one with a side-condition about free names; it alone expresses that abstraction is "vacuous" if there are no free occurrences. $\boldsymbol{\delta}$ is the only axiom which asserts the effect of abstracting a free occurrence of a name. $\boldsymbol{\zeta}$ asserts, essentially, that the monoidal category is symmetric; this is unlikely to be a consequence of the other axioms.

For $\boldsymbol{\sigma}$ there is a quick and easy independence proof. By examining each axiom, one can show that if $\mathsf{AC} \vdash s = t$ without use of $\boldsymbol{\sigma}$ then $\mathrm{fn}\, s = \mathrm{fn}\, t$. Hence without $\boldsymbol{\sigma}$ one cannot prove, for example, 2.5(1): $\langle x\rangle \cdot \boldsymbol{\omega} = \mathbf{id}$.

8 Action calculi with coding

Higher-order action calculi $\mathsf{AC}^h(\mathcal{K})$ are defined in [4] by adding to the controls $\mathcal{K}$ two special controls, *coding* and *application*. The dynamics of higher-order calculi is presented as a reaction rule for these controls. In the statics, however, there is nothing special about the control for application; therefore to unsderstand the static algebra (including molecular forms) for $\mathsf{AC}^h(\mathcal{K})$ we consider first only coding. In this section we show that the algebra we have developed, including the molecular form theorem 6.3, go through almost unchanged with the addition of coding; the only (and essential) change is that a datum is no longer just $\langle x\rangle$ for any name $x \in X$; instead, it may also take the form $\ulcorner a \urcorner$ (or $\ulcorner t \urcorner$), where a is a molecular form (or t is a term).

In this section, we outline the corresponding changes in the definitions, arguments and results of Sections 1–6. Subsection 8.n corresponds to Section n. We shall use $\mathsf{AC}^c(\mathcal{K})$ to stand for the action calculus over $\mathcal{K}$ with codes. We shall define terms $\mathcal{T}^c(\mathcal{K})$, a theory AC^c and molecular forms $\mathcal{M}^c(\mathcal{K})$; the main result is the isomorphism between $\mathcal{M}^c(\mathcal{K})$ and $\mathcal{T}^c(\mathcal{K})/\mathsf{AC}^c$.

8.1 Arities, names and controls In order to give the correct arity to a code we must extend the arity monoid $(M, \otimes, \epsilon)$ to a monoid $M^{\Rightarrow}$ by freely adding binary exponentiation $\Rightarrow$ of arities; thus if M has primes P then $M^{\Rightarrow}$ has primes $P^{\Rightarrow} \stackrel{\text{def}}{=} P \cup \{m \Rightarrow n \mid m, n \in M^{\Rightarrow}\}$. We also extend X to $X^{\Rightarrow}$ by adding an infinity of names at each new prime arity $m \Rightarrow n$.

We now add the *coding* control $\ulcorner\ \urcorner$, whose arity rule is

$$\frac{a : m \to n}{\ulcorner a \urcorner : \epsilon \to (m \Rightarrow n)}\ .$$

8.2 Terms The terms $\mathcal{T}^c(\mathcal{K})$ are just $\mathcal{T}(\mathcal{K}, \ulcorner\ \urcorner)$. However, to define the theory it is convenient to generalise the notion of *datum*. A *bare datum* (which is not a term) is either a name $x : p$, or a code $\ulcorner t \urcorner : m \Rightarrow n$ for any term $t : m \to n$. (Note that a bare datum has prime arity.) We use d, e to range over bare data. A *datum* is then the term $\langle d \rangle : \epsilon \to p$, for a bare datum $d : p$. Thus the *term* $\ulcorner t \urcorner$ is an abbreviation for $\langle \ulcorner t \urcorner \rangle$.

We allow a more general form of substitution $\{d/x\}$; it replaces every free occurrence $\langle x \rangle$ by the datum $\langle d \rangle$, using alpha conversion to avoid capture of free names in d.

The theory AC^c now differs from AC only in the $\boldsymbol{\sigma}$ rule:

The theory AC^c		
$\boldsymbol{\gamma}$:	$(x)t = \boldsymbol{\omega} \otimes t$	(x not free in t)
$\boldsymbol{\delta}$:	$(x)(\langle x \rangle \otimes \mathbf{id}_m) = \mathbf{id}_{p \otimes m}$	$(x : p)$
$\boldsymbol{\zeta}$:	$\mathbf{i}_{km} \cdot (t \otimes s) = (s \otimes t) \cdot \mathbf{i}_{ln}$	$(s : k \to l,\ t : m \to n)$
$\boldsymbol{\sigma}$:	$(\langle d \rangle \otimes \mathbf{id}_m) \cdot (x)t = \{d/x\}t$	$(t : m \to n)$.

The consequences of this theory listed in Lemma 2.5 need to be generalised in only one place; 2.5(1) becomes

$$\mathsf{AC}^c \vdash \langle d \rangle \cdot \boldsymbol{\omega} = \mathbf{id}_\epsilon \ .$$

8.3 Molecular forms The molecular forms $\mathcal{M}^c(\mathcal{K})$ are not just $\mathcal{M}(\mathcal{K}, \ulcorner\ \urcorner)$; the reason is that they are intended to be normal forms for the theory AC^c, which is richer than AC (because of $\boldsymbol{\sigma}$). However, they too are obtained by generalising the notion of datum. For molecular forms, a bare datum is either a name $x : p$, or a code $\ulcorner a \urcorner^{\mathcal{M}} : m \Rightarrow n$ for any molecular form $a : m \to n$; data and substitutions are then generalised exactly as they were for terms. We shall use d, e for data in molecular forms. (Strictly we should use different letters from what was used for data in terms; but we are running short of Roman letters.) Then actions and molecules take the respective forms

$$\begin{array}{rcl} a & = & (\vec{x})\, \mu_1 \cdots \mu_r\, \langle \vec{d} \rangle \\ \mu & = & \langle \vec{e} \rangle\, K \vec{b}\, (\vec{y}) \ , \end{array}$$

where previously the data-vectors $\langle \vec{d} \rangle, \langle \vec{e} \rangle$ contained only names. Note that a may now have subactions which occur within a datum, as well as occurring as action parameters to arbitrary controls.

Analogous to Proposition 3.1, we now have

Proposition $\mathcal{M}^c(\mathcal{K})$ is an action structure, and satisfies the axioms of AC^c. ■

8.4 Translating molecular forms to terms The translation $\widehat{(-)} : \mathcal{M}^c(\mathcal{K}) \to \mathcal{T}^c(\mathcal{K})$ is defined inductively on molecular forms; it must now taking care of translation of codes as well. Thus for actions, the translation is

$$\widehat{(\vec{x})\, \vec{\mu}\, \langle \vec{d} \rangle} \ \stackrel{\text{def}}{=} \ \begin{cases} (\vec{x}) \langle \widehat{\vec{d}} \rangle & \text{if } \vec{\mu} \text{ is empty} \\ (\vec{x})(\langle \widehat{\vec{e}} \rangle \cdot K \widehat{\vec{c}} \cdot \widehat{(\vec{x'}) \vec{\mu'} \langle \vec{d} \rangle}) & \text{if } \vec{\mu} = \lambda \vec{\mu'}, \text{ where } \lambda = \langle \vec{e} \rangle K \vec{c} (\vec{x'}) \ ; \end{cases}$$

the translation of (bare) data is

$$\widehat{x} = x\ , \qquad \widehat{\ulcorner a \urcorner^{\mathcal{M}}} = \ulcorner \widehat{a} \urcorner\ .$$

Then all the propositions of Section 4 go through, suitably adapted. Note that to say that the translation "preserves substitution" now means that

$$\widehat{\{\vec{d}/\vec{x}\}^{\mathcal{M}} a} = \{\widehat{\vec{d}}/\vec{x}\}\widehat{a}\ .$$

In summary, we have

Proposition The translation $\widehat{(-)} : \mathcal{M}^c(\mathcal{K}) \to \mathcal{T}^c(\mathcal{K})/\mathsf{AC}^c$ is well-defined; it preserves substitution, the action structure operations, data, discard, the control operations and coding. ■

8.5 Translating terms to molecular forms The only new feature of the translation $[\![-]\!] : \mathcal{T}^c(\mathcal{K}) \to \mathcal{M}^c(\mathcal{K})$ is, again, that it preserves coding. Thus there are the defining clauses

$$\begin{array}{rcl} [\![\langle d \rangle]\!] & \stackrel{\text{def}}{=} & \langle [\![d]\!] \rangle^{\mathcal{M}} \\ [\![x]\!] & \stackrel{\text{def}}{=} & x \\ [\![\ulcorner t \urcorner]\!] & \stackrel{\text{def}}{=} & \ulcorner [\![t]\!] \urcorner^{\mathcal{M}}\ . \end{array}$$

Again we summarise the analogous results:

Proposition The translation $[\![(-)]\!] : \mathcal{T}^c(\mathcal{K})/\mathsf{AC}^c \to \mathcal{M}^c(\mathcal{K})$ is well-defined; it preserves substitution, the action structure operations, data, discard, the control operations and coding. ■

8.6 Isomorphism of $\mathcal{M}^c(\mathcal{K})$ and $\mathcal{T}^c(\mathcal{K})/\mathsf{AC}^c$ Collecting together the preceding results, we can also assert

Theorem The translations $\widehat{(-)}$ and $[\![-]\!]$ between $\mathcal{M}^c(\mathcal{K})$ and $\mathcal{T}^c(\mathcal{K})/\mathsf{AC}^c$ constitute a static isomorphism of action structures, which moreover preserves the control operations $K \in \mathcal{K}$, the data $\langle x \rangle$, the discard $\boldsymbol{\omega}$, coding $\ulcorner\ \urcorner$ and the substitutions $\{\vec{d}/\vec{x}\}$ of data for names. ■

We shall refer to the isomorphic structures $\mathcal{M}^c(\mathcal{K})$ and $\mathcal{T}^c(\mathcal{K})/\mathsf{AC}^c$ indifferently as $\mathsf{AC}^c(\mathcal{K})$, the *action calculus with codes over* $\mathcal{K}$. Again, we are now free to drop the superscripts $^{\mathcal{M}}$ from the operation on molecular forms.

9 The higher-order action calculus $\mathsf{AC}^h(\mathcal{K})$

By adding codes to $\mathsf{AC}(\mathcal{K})$ we have allowed actions to be *encoded* as data, and passed as parameters to other actions; but this does not yet allow such parameters to be *decoded* and used. This second step is what entitles an action calculus to be called *higher-order*.

For this purpose, we add to $\mathsf{AC}^c(\mathcal{K})$ the *application* control **ap** : $(m \Rightarrow n) \otimes m \to n$. An **ap** molecule takes the form $\langle e\vec{d}\rangle\mathbf{ap}(\vec{y})$, with $e : m \Rightarrow n$, $\vec{d} : m$ and $\vec{y} : n$.

9.1 Dynamics We shall present the dynamics of coding and application for $\mathsf{AC}^c(\mathcal{K},\mathbf{ap})$ in two ways; a reduction relation for the molecular forms $\mathcal{M}^c(\mathcal{K},\mathbf{ap})$ and a reaction preorder for the term algebra $\mathcal{T}^c(\mathcal{K},\mathbf{ap})/\mathsf{AC}^c$. Of course these algebras are isomorphic, as we have just proved; the main aim of this subsection is to show that the dynamic presentations agree too.

$\mathsf{AC}^c(\mathcal{K},\mathbf{ap})$ inherits from $\mathsf{AC}(\mathcal{K})$ a reaction relation $\searrow_{\mathcal{K}}$ which represents the meaning of the controls $\mathcal{K}$. The interplay between $\searrow_{\mathcal{K}}$ and the dynamics of coding and application is analysed in [4]; it need not concern us here.

Definition ($\boldsymbol{\beta}$-reduction) The reduction relation $\longrightarrow_\beta$ over $\mathcal{M}^c(\mathcal{K},\mathbf{ap})$ is the least relation such that [1]

(1) If $a = (\ldots)[\ \vec{\kappa},\ \langle\ulcorner b\urcorner\vec{d}\rangle\mathbf{ap}(\vec{x}),\ \vec{\lambda}\ ]\langle\vec{f}\rangle$ and $b = (\vec{y})\,\vec{\mu}\,\langle\vec{e}\rangle$, then $a \longrightarrow_\beta a'$ where

$$a' \;=\; (\ldots)[\ \vec{\kappa},\ \sigma\vec{\mu},\ \tau\vec{\lambda}\ ]\langle\tau\vec{f}\rangle \qquad (\sigma = \{\vec{d}/\vec{y}\}, \tau = \{\sigma\vec{e}/\vec{x}\})\ ;$$

(2) If $b \longrightarrow_\beta b'$ and b is a subaction of a, then $a \longrightarrow_\beta a'$ where a' results from replacing b by b' in a. ■

In (1) the **ap** molecule is called a *redex* of a; but note that the reduction invokes a substitution outside the redex as well as within it.

Definition ($\boldsymbol{\beta}$-reaction) The reaction relation $\searrow_\beta$ over $\mathcal{T}^c(\mathcal{K},\mathbf{ap})$ is the least preorder such that

$$(\ulcorner t\urcorner \otimes \mathbf{id}_k)\cdot\mathbf{ap} \searrow_\beta t \qquad (t : k \to \ell) \qquad (\searrow\boldsymbol{\beta})$$

which is preserved by the action structure operations, the controls and coding, and also closed under provable equality in AC^c.

From the latter property, $\searrow_\beta$ is also a preorder over $\mathcal{T}^c(\mathcal{K},\mathbf{ap})/\mathsf{AC}^c$. ■

Proposition $\longrightarrow_\beta^*$ is preserved by substitution, the action structure operations, the controls and coding; more precisely if $a \longrightarrow_\beta a'$ then (1) $\sigma a \longrightarrow_\beta \sigma a'$; (2) $a \cdot b \longrightarrow_\beta^* a' \cdot b$ and $b \cdot a \longrightarrow_\beta b \cdot a'$; (3) $a \otimes b \longrightarrow_\beta a' \otimes b$ and $b \otimes a \longrightarrow_\beta b \otimes a'$; (4) $\mathbf{ab}_x a \longrightarrow_\beta \mathbf{ab}_x a'$; (5) $K(\cdots a \cdots) \longrightarrow_\beta K(\cdots a' \cdots)$; (6) $\langle\ulcorner a\urcorner\rangle \longrightarrow_\beta \langle\ulcorner a'\urcorner\rangle$.

[1] Sometimes, as here, we group the molecules of an action in square brackets for clarity.

Proof (1) and (2) are proved by standard analysis of the effect of substitutions; (3) and (4) are straightforward since the redex in a does not interact with any substitution; (5) and (6) are true by definition of $\longrightarrow_\beta$. ■

Note particularly that the first part of (2) is the only case where the *single-step* reduction $\longrightarrow_\beta$ is not preserved; this is because the redex of a may lie in one of its export data, which may be either duplicated or discarded by the composition with b.

We can now deduce

Proposition In the isomorphic structures $\mathcal{M}^c(\mathcal{K},\mathbf{ap})$ and $\mathcal{T}^c(\mathcal{K},\mathbf{ap})/\mathsf{AC}^c$ the relations $\longrightarrow_\beta^*$ and $\searrow_\beta$ coincide.

Proof It is enough to prove that $a \longrightarrow_\beta a'$ implies $\mathsf{AC}^c \vdash \widehat{a} \searrow_\beta \widehat{a'}$, and that $t \searrow_\beta t'$ implies $[\![t]\!] \longrightarrow_\beta^* [\![t']\!]$. For the former it is enough to treat the case that $a = \mu a_1$ where μ is the redex, and then rely on the fact that $\searrow_\beta$ is preserved by all operations. For the latter it is enough to treat the case that $t \searrow_\beta t'$ is an instance of the rule ${}^{\searrow}\boldsymbol{\beta}$, and then rely on the fact that $\longrightarrow_\beta^*$ is preserved by all operations. The details are straightforward. ■

Equational theory of higher-order action calculi In the λ-calculus, one gains a static or equational theory of β-conversion by imposing the congruence induced by β-reduction. We do the same here. First, we define the theory AC^h by converting the rule ${}^{\searrow}\boldsymbol{\beta}$ into an equation:

The theory AC^h		
$\boldsymbol{\beta}:$	$(\ulcorner t \urcorner \otimes \mathbf{id}_k)\cdot \mathbf{ap} = t$	$(t : k \to \ell)$
$\boldsymbol{\gamma}:$	$(x)t = \boldsymbol{\omega} \otimes t$	(x not free in t)
$\boldsymbol{\delta}:$	$(x)(\langle x\rangle \otimes \mathbf{id}_m) = \mathbf{id}_{p\otimes m}$	$(x : p)$
$\boldsymbol{\zeta}:$	$\mathbf{i}_{km}\cdot(t\otimes s) = (s\otimes t)\cdot \mathbf{i}_{ln}$	$(s : k\to l,\ t : m\to n)$
$\boldsymbol{\sigma}:$	$(\langle d\rangle \otimes \mathbf{id}_m)\cdot (x)t = \{{}^{d}\!/\!_{x}\}t$	$(t : m \to n)$.

We define $\mathsf{AC}^h(\mathcal{K})$, the *higher-order action calculus over* $\mathcal{K}$, to be the factored term algebra $\mathcal{T}^h(\mathcal{K})/\mathsf{AC}^h$. We now look for its molecular presentation.

We know that the preorder $\searrow_\beta$ over $\mathcal{T}^c(\mathcal{K},\mathbf{ap})/\mathsf{AC}^c$, or equivalently over $\mathcal{M}^c(\mathcal{K},\mathbf{ap})$, is preserved by all operations; therefore the equivalence $\equiv_\beta \stackrel{\text{def}}{=} (\searrow_\beta \cup \searrow_\beta^{-1})^*$ is a congruence. But clearly $\mathsf{AC}^h(\mathcal{K})$ is just the quotient of $\mathsf{AC}^c(\mathcal{K},\mathbf{ap})$ by $\equiv_\beta$, i.e.

$$\mathsf{AC}^h(\mathcal{K}) \;=\; \mathsf{AC}^c(\mathcal{K},\mathbf{ap})/\equiv_\beta \;.$$

Now we may take advantage of the equation $\searrow_\beta \;=\; \longrightarrow_\beta^*$ to discover the molecular presentation of $\mathsf{AC}^h(\mathcal{K})$. First we need an important property of $\longrightarrow_\beta$:

Theorem The reduction relation $\longrightarrow_\beta$ over $\mathcal{M}^c(\mathcal{K},\mathbf{ap})$ is both strongly normalizing and confluent.

Proof For strong normalization, one can (with care) adapt a standard method for the typed λ-calculus; see for example [1]. We omit the details here.

For confluence, it is simplest first to prove weak confluence, namely that if $a \longrightarrow_\beta a_1$ and $a \longrightarrow_\beta a_2$ then there exists a' for which $a_1 \longrightarrow_\beta^* a'$ and $a_2 \longrightarrow_\beta^* a'$. But by a well-known argument, weak confluence with strong normalization implies confluence. ■

It follows from this theorem that every action $a \in \mathcal{M}^c(\mathcal{K}, \mathbf{ap})$ has a unique normal form under β-reduction; we shall write it as $\underline{a}$. The normal forms are exactly those actions with no redex, i.e. every **ap** molecule takes the form $\langle z\vec{d}\rangle\mathbf{ap}(x)$, having a *name* (not a code) as its leading datum.

Thus, since $\equiv_\beta$ is also the congruence $(\longrightarrow_\beta \cup \longrightarrow_\beta^{-1})^*$ induced by β-reduction, we make the following definition:

Definition The action structure $\mathcal{M}^h(\mathcal{K})$ is given as follows: its actions are the normal forms of $\mathcal{M}^c(\mathcal{K},\mathbf{ap})$, and its operations are those of $\mathcal{M}^c(\mathcal{K},\mathbf{ap})$ followed by normalisation. For example, the composition of a with b is $\underline{a \cdot b}$; the operation of the substitution σ upon a is $\underline{\sigma a}$. ■

(Note that product, abstraction and coding preserve normal forms.) Then the isomorphism between $\mathcal{M}^c(\mathcal{K},\mathbf{ap})$ and $\mathcal{T}^c(\mathcal{K},\mathbf{ap})/\mathsf{AC}^c$ is easily adapted to yield the following:

Theorem $\mathcal{M}^h(\mathcal{K})$ is a presentation of $\mathsf{AC}^h(\mathcal{K})$. That is, there is a static isomorphism between $\mathcal{M}^h(\mathcal{K})$ and $\mathcal{T}^h(\mathcal{K})/\mathsf{AC}^h$ which preserves the action structure operations, the control operations $K \in \mathcal{K}$, the data $\langle x\rangle$, the discard $\boldsymbol{\omega}$, coding $\ulcorner\ \urcorner$ and the substitutions $\{\vec{d}/\vec{x}\}$ of data for names. ■

References

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

[2] Milner, R., *Action structures and the π-calculus*, to appear in Proceedings of NATO Advanced Study Institute on **Proof and Computation** (held at Marktoberdorf, 1993), about 60 pages. (This paper combines of two previous reports: *Action structures*, Research Report ECS–LFCS–92–249, and *Action structures for the π-calculus*, Research Report ECS–LFCS–93–264, Laboratory for Foundations of Computer Science, Computer Science Department, Edinburgh University.

[3] 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.

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