Equivalence of Algebraic λ-calculi

Alejandro Díaz-Caro[1] Simon Perdrix[1,2]
Christine Tasson[3] Benoît Valiron[1]

[1]LIG, Université de Grenoble

[2]CNRS

[3]CEA-LIST, MeASI

HOR • July 14th, 2010 • Edinburgh

Algebraic calculi

- Algebraic $\equiv$ Module structure

$$(\lambda x.yx + \frac{1}{2} \cdot \lambda x.y(yx))\ \lambda z.z$$

- Two origins
  - $\lambda_{\text{alg}} \longrightarrow$ differential calculus
  - $\lambda_{\text{lin}} \longrightarrow$ quantum computation
- Differences in
  - the encoding of the module structure
  - how the reduction rules apply

λ_{alg} and λ_{lin}

share the same set of terms

$$\begin{array}{rcl} M,N,L & ::= & V \mid (M)\,N \mid M+N \mid \alpha\cdot M \\ U,V,W & ::= & B \mid \alpha\cdot B \mid V+W \mid \mathbf{0} \\ B & ::= & x \mid \lambda x.M \end{array}$$

closed under associativity and conmutativity

$$M+N=N+M \qquad (M+N)+L=M+(N+L)$$

- base term is either a variable or abstraction
- a value is either $\mathbf{0}$ or of the form $\sum \alpha_i \cdot B_i$;
- a general term can be anything.

Rules for λ_{alg}

Group E

$$\begin{array}{rcl} \alpha \cdot \mathbf{0} & =_a & \mathbf{0} \\ \alpha \cdot (\beta \cdot M) & =_a & (\alpha \times \beta) \cdot M \\ 1 \cdot M & =_a & M \end{array} \qquad \begin{array}{rcl} \mathbf{0} + M & =_a & M \\ 0 \cdot M & =_a & \mathbf{0} \\ \alpha \cdot (M + N) & =_a & \alpha \cdot M + \alpha \cdot N \end{array}$$

Group F

$$\alpha \cdot M + \beta \cdot M =_a (\alpha + \beta) \cdot M$$

Group A

$$\begin{array}{rl} (M + N)\ L =_a & (M)\ L + (N)\ L \\ (\alpha \cdot M)\ N =_a & \alpha \cdot (M)\ N \\ (\mathbf{0})\ M =_a & \mathbf{0} \end{array}$$

Group B

$$(\lambda x.M)\ N \to_a M[x := N]$$

Rules for λ_{lin}

Group E

$$\begin{array}{rcl} \alpha \cdot \mathbf{0} & \to_\ell & \mathbf{0} \\ \alpha \cdot (\beta \cdot M) & \to_\ell & (\alpha \times \beta) \cdot M \\ 1 \cdot M & \to_\ell & M \end{array} \qquad \begin{array}{rcl} \mathbf{0} + M & \to_\ell & M \\ 0 \cdot M & \to_\ell & \mathbf{0} \\ \alpha \cdot (M + N) & \to_\ell & \alpha \cdot M + \alpha \cdot N \end{array}$$

Group F

$$\begin{array}{rcl} \alpha \cdot M + \beta \cdot M & \to_\ell & (\alpha + \beta) \cdot M \\ \alpha \cdot M + M & \to_\ell & (\alpha + 1) \cdot M \\ M + M & \to_\ell & (1 + 1) \cdot M \end{array}$$

Group A

$$\begin{array}{c|c} (M + N)\ L \to_\ell (M)\ L + (N)\ L & (M)\ (N + L) \to_\ell (M)\ N + (M)\ L \\ (\alpha \cdot M)\ N \to_\ell \alpha \cdot (M)\ N & (M)\ \alpha \cdot N \to_\ell \alpha \cdot (M)\ N \\ (\mathbf{0})\ M \to_\ell \mathbf{0} & (M)\ \mathbf{0} \to_\ell \mathbf{0} \end{array}$$

Group B

$$(\lambda x.M)\ B \to_a M[x := B]$$

Context rules

Some common rules

$$\frac{M \to M'}{(M)\ N \to (M')\ N} \qquad \frac{M \to M'}{M + N \to M' + N}$$

$$\frac{N \to N'}{M + N \to M + N'} \qquad \frac{M \to M'}{\alpha \cdot M \to \alpha \cdot M'}$$

plus one additional rule for λ_{lin}

$$\frac{M \to_\ell M'}{(V)\ M \to_\ell (V)\ M'}$$

Modifications to the original calculi

The most important

- No reduction under lambda (*à la* Plotkin)
- $\lambda x.M$ is a base term for any M

$$\lambda x.(\alpha \cdot M + \beta \cdot N) \neq \alpha \cdot \lambda x.M + \beta \cdot \lambda x.N$$

Confluence

This system is not trivially confluent

Example

Let $Y_b = (\lambda x.(b + (x)\ x))\ (\lambda x.(b + (x)\ x))$, then

$$\mathbf{0} \leftarrow Y_b - Y_b \rightarrow Y_b + b - Y_b \rightarrow b$$

Several possible solutions

- Type system $\rightarrow$ strong normalisation
- Take scalars as positive reals
- Restrict some of the reduction rules

For this talk, we simply assume that confluence holds.

Question

Can we simulate λ_{lin} with λ_{alg} and λ_{alg} with λ_{lin} ?

Simulating λ_{alg} with λ_{lin}

Thunks

Idea: encapsulating terms M as $B = \lambda f.M$. So $M \equiv (B)\ f$

Simulating λ_{alg} with λ_{lin}

Thunks

Idea: encapsulating terms M as $B = \lambda f.M$. So $M \equiv (B)\ f$

$$
\begin{array}{rcl}
(\!| x |\!)_f & = & (x)\ f \\
(\!| 0 |\!)_f & = & 0 \\
(\!| \lambda x.M |\!)_f & = & \lambda x.(\!| M |\!)_f \\
(\!| (M)\ N |\!)_f & = & ((\!| M |\!)_f)\ \lambda z.(\!| N |\!)_f \\
(\!| M + N |\!)_f & = & (\!| M |\!)_f + (\!| N |\!)_f \\
(\!| \alpha \cdot M |\!)_f & = & \alpha \cdot (\!| M |\!)_f
\end{array}
$$

If the encoding $\lambda_{\text{alg}} \to \lambda_{\text{lin}}$ is denoted with $(\!| - |\!)_f$, one could expect the result

$$M \to_a^* N \qquad \Rightarrow \qquad (\!| M |\!)_f \to_\ell^* (\!| N |\!)_f$$

Problem

Example

$$(\lambda x.\lambda y.(y)\ x)\ I \to_a \lambda y.(y)\ I,$$

$$\begin{aligned}
(\!|(\lambda x.\lambda y.(y)\ x)\ I|\!)_f &= (\lambda x.\lambda y.((y)\ f)\ (\lambda z.(x)\ f))\ (\lambda z\,\lambda x.(x)\ f) \\
&\to_\ell^* \lambda y.((y)\ f)\ (\lambda z.(\lambda z.\lambda x.(x)\ f)\ f), \\
(\!|\lambda y.(y)\ I|\!)_f &= \lambda y.((y)\ f)\ (\lambda z.\lambda x.(x)\ f)
\end{aligned}$$

"Administrative" redex hidden in the first one

Removing administrative redexes

$$
\begin{array}{rcl}
Admin_f\, 0 & = & 0 \\
Admin_f\, x & = & x \\
Admin_f\, (\lambda f.M)\ f & = & Admin_f\, M \\
Admin_f\, (M)\ N & = & (Admin_f\, M)\ Admin_f\, N \\
Admin_f\, \lambda x.M & = & \lambda x.Admin_f\, M \qquad (x \neq f) \\
Admin_f\, M + N & = & Admin_f\, M + Admin_f\, N \\
Admin_f\, \alpha \cdot M & = & \alpha \cdot Admin_f\, M
\end{array}
$$

Theorem (Simulation)

For any program (i.e. closed term) M, and value V, if $M \to_a^* V$ then exists a value W such that

$$(\!|M|\!)_f \to_\ell^* W \text{ and } Admin_f\, W = Admin_f\, (\!|V|\!)_f$$

First,

$$M \xrightarrow[\ell]{} N$$

$$\equiv_{Admin_f}$$

$$M'$$

Lemma

If $Admin_f\, M = Admin_f\, M'$ and $M \to_\ell N$ then there exists N' such that $Admin_f\, N = Admin_f\, N'$ and such that $M' \to_\ell^* N'$.

First,

$$
\begin{array}{ccc}
M & \xrightarrow[\ell]{} & N \\
\equiv_{Admin_f} & & \equiv_{Admin_f} \\
M' & \xrightarrow[\ell]{*} & N'
\end{array}
$$

Lemma

If $Admin_f\, M = Admin_f\, M'$ and $M \to_\ell N$ then there exists N' such that $Admin_f\, N = Admin_f\, N'$ and such that $M' \to_\ell^* N'$.

Then,

M

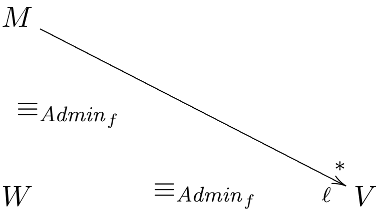
$\equiv_{Admin_f}$

W

Lemma

If W is a value and M a term such that $Admin_f\, W = Admin_f\, M$, then there exists a value V such that $M \to_\ell^* V$ and $Admin_f\, W = Admin_f\, V$.

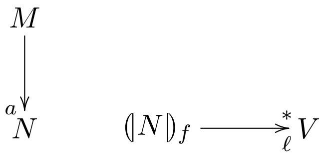
Then,



Lemma

If W is a value and M a term such that $Admin_f\,W = Admin_f\,M$, then there exists a value V such that $M \to_\ell^* V$ and $Admin_f\,W = Admin_f\,V$.

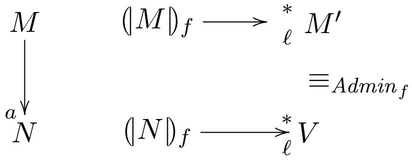
Finally,



Lemma

For any program (i.e. closed term) M*, if* $M \to_a N$ *and* $(\!|N|\!)_f \to_\ell^* V$ *for a value V, then there exists* M' *such that* $(\!|M|\!)_f \to_\ell^* M'$ *such that* $Admin_f\, M' = Admin_f\, V$*.*

Finally,



Lemma

For any program (i.e. closed term) M*, if* $M \to_a N$ *and* $(|N|)_f \to_\ell^* V$ *for a value V, then there exists* M' *such that* $(|M|)_f \to_\ell^* M'$ *such that* $Admin_f\, M' = Admin_f\, V$*.*

Now, we want

$$M \xrightarrow[a]{*} V$$

$$(\!|M|\!)_f \qquad\qquad (\!|V|\!)_f$$

Now, we want

$$M \longrightarrow_a^{*} V$$

$$(\!|M|\!)_f \qquad\qquad (\!|V|\!)_f$$

$$(\!|M|\!)_f \longrightarrow_\ell^{*} W \equiv_{Admin_f} (\!|V|\!)_f$$

Base case

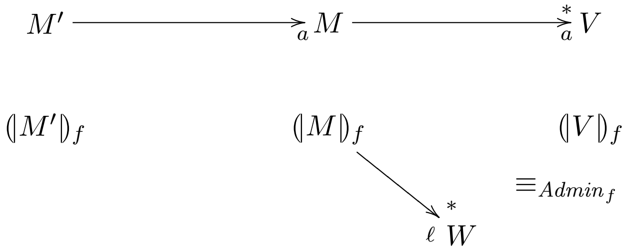
$$V \qquad = \qquad V$$

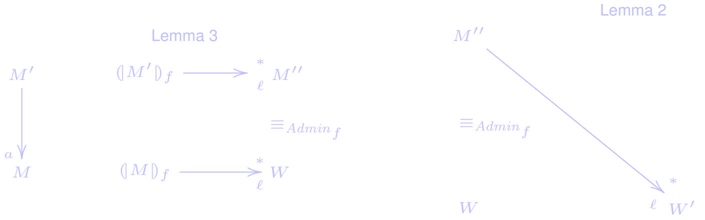
$$(\!|V|\!)_f \qquad\qquad (\!|V|\!)_f$$

$$= \qquad\qquad =$$

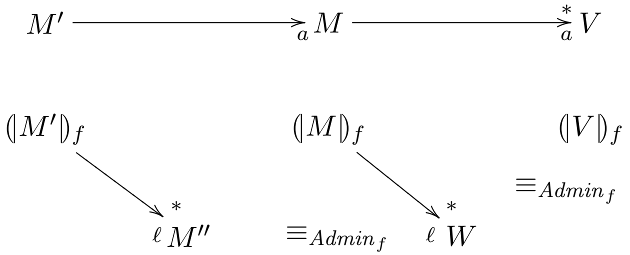
$$(\!|V|\!)_f$$

Inductive case

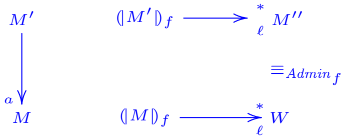




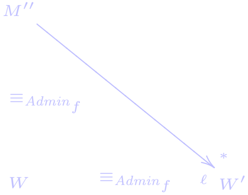
Inductive case



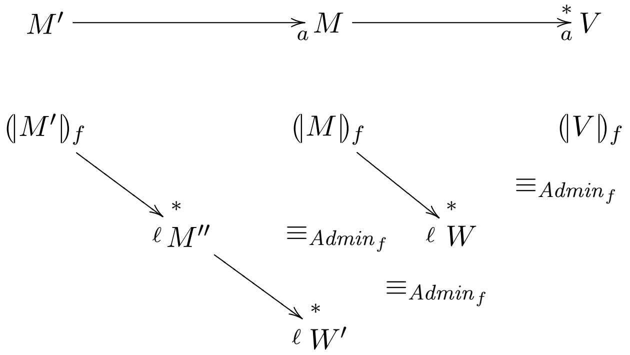
Lemma 3



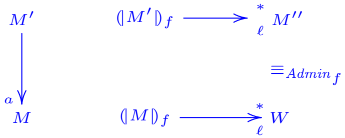
Lemma 2



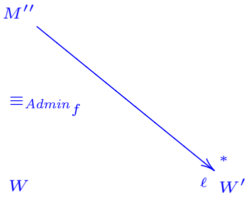
Inductive case



Lemma 3



Lemma 2



Simulating λ_{lin} with λ_{alg}

CPS encoding

We define $[\![-]\!] : \lambda_{\text{lin}} \to \lambda_{\text{alg}}$

$$
\begin{array}{rcl}
[\![x]\!] & = & \lambda f.(f)\ x \\
[\![\mathbf{0}]\!] & = & \mathbf{0} \\
[\![\lambda x.M]\!] & = & \lambda f.(f)\ \lambda x.[\![M]\!] \\
[\![(M)\ N]\!] & = & \lambda f.([\![M]\!])\ \lambda g.([\![N]\!])\ \lambda h.((g)\ h)\ f \\
[\![\alpha \cdot M]\!] & = & \lambda f.(\alpha \cdot [\![M]\!])\ f \\
[\![M + N]\!] & = & \lambda f.([\![M]\!] + [\![N]\!])\ f
\end{array}
$$

Now, given the continuation $I = \lambda x.x$, if $M \to_\ell V$ we want $([\![M]\!])\ I \to_a W$ for some W related to $[\![V]\!]$

The encoding for values

$$\begin{array}{rcl}\Psi(x) & = & x,\\ \Psi(0) & = & 0,\\ \Psi(\lambda x.M) & = & \lambda x.\llbracket M\rrbracket,\\ \Psi(\alpha\cdot V) & = & \alpha\cdot\Psi(V),\\ \Psi(V+W) & = & \Psi(V)+\Psi(W).\end{array}$$

- $\Psi(V)$ is such that $\llbracket V\rrbracket \to_a^* \Psi(V)$.

Theorem (Simulation)

$$M \to_\ell^* V \qquad \Rightarrow \qquad (\llbracket M\rrbracket)\ \lambda x.x \to_a^* \Psi(V).$$

An auxiliary definition

Note that

$$(\llbracket B \rrbracket)\ K = (\lambda f.(f)\ \Psi(B))\ K \to_a^* (K)\ \Psi(B).$$

- We define the term $B : K = (K)\ \Psi(B)$.
- We extend this definition to other terms in some not-so-easy way due to the algebraic properties of the language.
- It is the main difficulty in this proof.

The auxiliary definition

Let $(:) : \lambda_{\text{lin}} \times \lambda_{\text{alg}} \to \lambda_{\text{alg}}$:

$$
\begin{aligned}
&B : K = (K)\ \Psi(B) \\
&(M + N) : K = M : K + N : K \\
&\alpha \cdot M : K = \alpha \cdot (M : K) \\
&\mathbf{0} : K = \mathbf{0} \\
&(U)\ B : K = ((\Psi(U))\ \Psi(B))\ K \\
&(U)\ (V + W) : K = ((U)\ V + (U)\ W) : K \\
&(U)\ (\alpha \cdot B) : K = \alpha \cdot (U)\ B : K \\
&(U)\ \mathbf{0} : K = \mathbf{0} \\
&(U)\ N : K = N : \lambda f.((\Psi(U))\ f)\ K \\
&(M)\ N : K = M : \lambda g.(\llbracket N \rrbracket)\ \lambda h.((g)\ h)\ K
\end{aligned}
$$

Proof

If $M \to_\ell^* V$,

$$(\![M]\!)\ (\lambda x.x)$$

$$\Psi(V)$$

Proof

If K is a value:

- For all M, $(\llbracket M \rrbracket)\ K \to_a^* M : K$
- If $M \to_\ell N$ then $M : K \to_a^* N : K$
- If V is a value, $V : K \to_a^* (K)\ \Psi(V)$

If $M \to_\ell^* V$,

$$(\llbracket M \rrbracket)\ (\lambda x.x) \quad \to_a^* M : \lambda x.x$$

$$\Psi(V)$$

Proof

If K is a value:

- For all M, $(\llbracket M \rrbracket)\ K \to_a^* M : K$
- If $M \to_\ell N$ then $M : K \to_a^* N : K$
- If V is a value, $V : K \to_a^* (K)\ \Psi(V)$

If $M \to_\ell^* V$,

$$\begin{aligned}(\llbracket M \rrbracket)\ (\lambda x.x) &\to_a^* M : \lambda x.x \\ &\to_a^* V : \lambda x.x\end{aligned}$$

$$\Psi(V)$$

Proof

If K is a value:

- For all M, $(\llbracket M \rrbracket)\ K \to_a^* M : K$
- If $M \to_\ell N$ then $M : K \to_a^* N : K$
- If V is a value, $V : K \to_a^* (K)\ \Psi(V)$

If $M \to_\ell^* V$,

$$\begin{aligned}(\llbracket M \rrbracket)\ (\lambda x.x) \quad & \to_a^* M : \lambda x.x \\ & \to_a^* V : \lambda x.x \\ & \to_a^* (\lambda x.x)\ \Psi(V) \\ & \quad\ \Psi(V)\end{aligned}$$

Proof

If K is a value:

- For all M, $(\llbracket M \rrbracket)\ K \to_a^* M : K$
- If $M \to_\ell N$ then $M : K \to_a^* N : K$
- If V is a value, $V : K \to_a^* (K)\ \Psi(V)$

If $M \to_\ell^* V$,

$$\begin{aligned}(\llbracket M \rrbracket)\ (\lambda x.x) \quad & \to_a^* M : \lambda x.x \\ & \to_a^* V : \lambda x.x \\ & \to_a^* (\lambda x.x)\ \Psi(V) \\ & \to_a \Psi(V)\end{aligned}$$

Future and ongoing work

In classical lambda-calculus:

- CPS can also be done to interpret call-by-name in call-by-value [Plotkin 75].
- The thunk construction can be related to CPS [Hatcliff and Danvy 96].

Can we relate to these?

Future and ongoing work

Other interesting questions

- The choice between equality and reduction rules seems arbitrary.

Can we exchange them?
Are the simulations still valid?