The multiary version of the λ-calculus with generalised application

José Espírito Santo and Luís Pinto

14 April 2004

Plan

1. The system $\lambda\mathbf{J^m}$
   - (a) multiary sequent terms (Schwichtenberg)
   - (b) general elimination rules (von Plato)
   - (c) ΛJ (Joachimski and Matthes)
   - (d) $\lambda\mathbf{J^m}$ and its subsystems
   - (e) Sequent calculus vs natural deduction
2. Relationship between $\lambda\mathbf{J^m}$ and ΛJ
   - (a) overlap between multiarity and generality
3. By-product: confluence and SN for reduction in $\lambda\mathbf{J^m}$

Multiarity

$$\frac{\vdash A \quad C \vdash D}{A \supset C \vdash D}\, Left = 0 - Left$$

$$\frac{\vdash A \quad \vdash B_1 \quad \ldots \quad \vdash B_k \quad C \vdash D}{A \supset B_1 \supset \ldots \supset B_k \supset C \vdash D}\, k - Left$$

Instead of a rule for each k, a single rule

$$\frac{\vdash A \quad ; B \vdash C \quad C \vdash D}{A \supset B \vdash D}\, m - Left$$

The latter uses Herbelin's trick: B is in the stoup, hence $B = B_1 \supset \ldots \supset B_k \supset C$, for some $k \geq 0$.

Computationally, multiarity is the ability of applying fuctions to lists of arguments

$\boxed{\Lambda J\text{: the }\lambda\text{-calculus with generalised application}}$

Expressions

$$t, u, v ::= x \mid \lambda x.t \mid \quad \underbrace{t(u, x.v)}_{g-application}$$

Typing generalised application

$$\frac{\Gamma \vdash t : A \supset B \quad \Gamma \vdash u : A \quad \Gamma, x : B \vdash v : C}{\Gamma \vdash t(u, x.v) : C}\, g - Elim$$

Proviso: $x \notin \Gamma$

Reduction rules

$$\begin{array}{rcl} (\lambda x.t)(u, y.v) & \rightarrow_\beta & s(s(u, x, t), y, v) \\ t(u, x.v)(u', y.v') & \rightarrow_\pi & t(u, x.v(u', y.v')) \end{array}$$

(s stands for substitution)

$\boxed{\lambda\mathbf{J^m}\text{: the generalised multiary } \lambda\text{-calculus}}$

Expressions

$$
\begin{array}{rcl}
t,u,v & ::= & x \mid \lambda x.t \mid \underbrace{t(u,l,(x)v)}_{\textit{gm-application}} \\
l & ::= & t::l \mid []
\end{array}
$$

Sequents

$\Gamma \vdash t : A \qquad \Gamma; B \vdash l : A$

Typing rules

$$
\frac{}{x:A,\Gamma \vdash x:A}\,Axiom \qquad \frac{x:A,\Gamma \vdash t:B}{\Gamma \vdash \lambda x.t : A \supset B}\,Right
$$

$$
\frac{\Gamma \vdash t:A \supset B \qquad \Gamma \vdash u:A \quad \Gamma;B \vdash l:C \quad x:C,\Gamma \vdash v:D}{\Gamma \vdash t(u,l,(x)v):D}\,gm-Elim
$$

$$
\frac{}{\Gamma;C \vdash [] : C}\,Ax \qquad \frac{\Gamma \vdash u:A \quad \Gamma;B \vdash l:C}{\Gamma;A \supset B \vdash u::l:C}\,Lft
$$

Proviso: $x \notin \Gamma$ in $Right$ and $gm - Elim$

Recovering left rules

$$\dfrac{\overbrace{\Gamma \vdash t : A \supset B}^{(1)} \quad \Gamma \vdash u : A \quad \overbrace{\Gamma ; B \vdash l : C}^{(2)} \quad x : C, \Gamma \vdash v : D}{\Gamma \vdash t(u, l, (x)v) : D}\, gm - Elim$$

Multiary left introduction

(1) must be an instance of $Axiom$, $(y : A \supset B) \in \Gamma$ and $t = y$

$$\dfrac{\Gamma \vdash u : A \quad \Gamma ; B \vdash l : C \quad x : C, \Gamma \vdash v : D}{\Gamma \vdash y(u, l, (x)v) : D}\, \textit{m-Left}$$

Left introduction

In addition to the restrictions above, (2) is an instance of Ax, $l = []$ and $B = C$

$$\dfrac{\Gamma \vdash u : A \quad x : B, \Gamma \vdash v : D}{\Gamma \vdash y(u, [], (x)v) : D}\, \textit{Left}$$

Reduction rules

$$\begin{array}{rcl}
(\lambda x.t)(u,[],(y)v) & \rightarrow_{\beta_1} & \mathbf{s}(\mathbf{s}(u,x,t),y,v) \\
(\lambda x.t)(u,v::l,(y)v) & \rightarrow_{\beta_2} & \mathbf{s}(u,x,t)(v,l,(y)v) \\
t(u,l,(x)v)(u',l',(y)v') & \rightarrow_{\pi} & t(u,l,(x)v(u',l',(y)v')) \\
t(u,l,(x)x(u',l',(y)v')) & \rightarrow_{\mu} & t(u,\mathbf{append}(l,u',l'),(y)v')
\end{array}$$

Proviso for μ: $x \notin u', l', v'$

(s stands for substitution)

$(\beta_1), (\beta_2), (\pi)$-normal forms:

$$\begin{array}{rcl}
t,u,v & ::= & x \mid \lambda x.t \mid x(u,l,(y)v) \\
l & ::= & u::l \mid []
\end{array}$$

(μ)-normal forms: in every gm-application $t(u,l,(y)v)$:

if $v = y(u',l',(y')v')$, y must occur either in u', l' or v'

("if v introduces y, it must do so with contraction")

Elimination rules for subsystems of $\lambda\mathbf{J^m}$

$$\frac{\Gamma \vdash t : A \supset B \qquad \Gamma \vdash u : A \qquad \overbrace{\Gamma ; B \vdash l : C}^{(1)} \qquad \overbrace{x : C, \Gamma \vdash v : D}^{(2)}}{\Gamma \vdash t(u, l, (x)v) : D}\, gm - Elim$$

Subsystem $\lambda\mathbf{J}$

(1) must be an instance of Ax, $l = []$ and $B = C$. $t(u{\cdot}(x)v)$ abbreviates $t(u, [], (x)v)$

$$\frac{\Gamma \vdash t : A \supset B \qquad \Gamma \vdash u : A \qquad x : B, \Gamma \vdash v : D}{\Gamma \vdash t(u{\cdot}(x)v) : D}\, g - Elim$$

Subsystem $\lambda\mathcal{G}$

In addition to the restrictions defining $\lambda\mathbf{J}$, (2) must be an instance of $Axiom$, $v = x$ and $C = D$. $t[u]$ abbreviates $t(u, [], (x)x)$

$$\frac{\Gamma \vdash t : A \supset B \quad \Gamma \vdash u : A}{\Gamma \vdash t[u] : B}\, Elim$$

$\boxed{\text{Sequent calculus view on } g - Elim}$

View

$$\frac{\vdash A \supset B \quad \vdash A \quad B \vdash C}{\vdash C}$$

as

$$\frac{\vdash A \supset B \qquad \dfrac{\vdash A \quad B \vdash C}{A \supset B \vdash C}\,(1)}{\vdash C}\,(2)$$

(1) linear left introduction of $A \supset B$

(2) right-permuted cut

- This slightly extends Gentzen's mapping $\mathcal{G}$ of natural deduction into sequent calculus.
- Reduction in $\lambda\mathbf{J}$ is about the elimination of these cuts. Reduction in $\lambda\mathbf{J^m}$ is about the elimination of cuts like (2), except that (1) should be a linear *multiary* left introduction

Sequent calculus vs natural deduction

$$\begin{tikzcd}
& \lambda\mathbf{J}^{\mathbf{m}} \arrow[dl, "\mu"'] \arrow[dr, "\nu, q"] & & \\
\lambda\mathbf{J}^{\mathbf{m}}_{\mu} & & \lambda\mathbf{J} \arrow[ll, "\cong"', "\mu"] \arrow[d, "p"'] & \Lambda J \arrow[l, "\cong"', "\mathcal{G}'"] \arrow[d, "p"'] \\
& & \lambda\mathcal{G} & \lambda \arrow[l, "\cong"', "\mathcal{G}"]
\end{tikzcd}$$

sequent
calculus

natural
deduction

μ = μ-normal-form-of
q = express multiary application with iterated application
ν = express multiarity with generality
p = execute explicit substitution
$\mathcal{G}$ = Gentzen's mapping
$\mathcal{G}'$ = Extension of $\mathcal{G}$ to ΛJ
$\lambda\mathbf{J}^{\mathbf{m}}_{\mu}$ = μ-normal-forms

Expressing t applied to u_0, u_1, u_2

$$\begin{tikzcd}
t u_0 u_1 u_2 \arrow[d, equal] & \lambda \\
t(u_0 \cdot (x_1)x_1)(u_1 \cdot (x_2)x_2)(u_2 \cdot (x_3)x_3) \arrow[d, two heads, "\pi"'] & \lambda\mathbf{J} \\
t(u_0 \cdot (x_1)x_1(u_1 \cdot (x_2)x_2(u_2 \cdot (x_3)x_3))) \arrow[d, equal] & \\
t(u_0, [], (x_1)x_1(u_1, [], (x_2)x_2(u_2, [](x_3)x_3))) \arrow[d, two heads, "\mu"'] & \lambda\mathbf{J^m} \\
t(u_0, [u_1, u_2], (x_3)x_3) &
\end{tikzcd}$$

Overlap between multiarity and generality

$$t(u,[],(x)x(u',l',(y)v)) \underset{\xrightarrow{\mu}}{\xleftarrow{\nu}} t(u,u'::l',(y)v)$$

or, more generally,

$$t(u,l,(x)x(u',l',(y)v)) \underset{\xrightarrow{\mu}}{\xleftarrow{\nu}} t(u,\mathbf{append}(l,u'::l'),(y)v)$$

proviso: $x \notin u', l', v$

- ν: express multiarity (=non-empty lists) with general application
- μ: use multiarity as a shorthand, whenever possible

$\boxed{\text{Mappings } \mu \text{ and } \nu}$

$$\nu : \mathbf{\Lambda J^m} \longrightarrow \mathbf{\Lambda J}$$

$$\begin{array}{rcl}
\nu(x) & = & x \\
\nu(\lambda x.t) & = & \lambda x.\nu(t) \\
\nu(t(u,l,(x)v)) & = & \nu(t)(\nu(u)\cdot(z)\nu'(z,l,x,\nu(v))), \quad z \text{ fresh} \\
\\
\nu'(z,[],x,v) & = & \mathrm{s}(z,x,v) \\
\nu'(z,u::l,x,v) & = & z(\nu(u)\cdot(w)\nu'(w,l,x,v)), \quad w \text{ fresh}
\end{array}$$

$$\mu : \mathbf{\Lambda J^m} \longrightarrow \mathbf{\Lambda J^m_\mu}$$

$$\begin{array}{rcl}
\mu(x) & = & x \\
\mu(\lambda x.t) & = & \lambda x.\mu(t) \\
\mu(t(u,l,(x)v)) & = & \begin{cases} \mu(t)(\mu(u),\mathbf{append}(\mu'(l),u',l'),(y)v'), \\ \qquad\qquad \text{if } \mu(v) = x(u',l',(y)v') \text{ and} \\ \qquad\qquad x \notin u',l',v' \\ \\ \mu(t)(\mu(u),\mu'(l),(x)\mu(v)), \quad \text{otherwise} \end{cases} \\
\\
\mu'([]) & = & [] \\
\mu'(u::l) & = & \mu(u)::\mu'(l)
\end{array}$$

Results

1. Preliminaries

- μ is SN and confluent and $\mu(t)$ is the unique μ-normal-form of t
- The $\lambda\mathbf{J}$-terms are the fixed-points of ν
- $t \rightarrow_{\mu}^{*} t'$ iff $\nu(t') = t$ $(t \in \lambda\mathbf{J})$
- μ and ν establish a bijection between $\lambda\mathbf{J}$-terms and μ-normal-forms

2. Preservation of reduction

$t \rightarrow_{\beta} t'$ in $\lambda\mathbf{J}^{\mathbf{m}}$	$\mu(t) \rightarrow_{\beta}\rightarrow_{\mu}^{*} \mu(t')$	$\nu(t) \rightarrow_{\beta} \nu(t')$
$t \rightarrow_{\pi} t'$ in $\lambda\mathbf{J}^{\mathbf{m}}$	no preservation	$\nu(t) \rightarrow_{\pi}^{+} \nu(t')$
$t_0 \rightarrow_{\pi} t'$ in $\lambda\mathbf{J}^{\mathbf{m}}$		$\nu(t_0) \rightarrow_{\pi'} \nu(t')$
$t \rightarrow_{\pi'} t'$ in $\lambda\mathbf{J}$	$\mu(t) \rightarrow_{\pi}\rightarrow_{\mu}^{*} \mu(t')$	
$t \rightarrow_{\mu} t'$ in $\lambda\mathbf{J}^{\mathbf{m}}$	$\mu(t) = \mu(t')$	$\nu(t) = \nu(t')$

($\beta = \beta_1 \cup \beta_2$; t_0 μ-normal; π' a variation on π s.t. $\rightarrow_{\pi'} \subseteq \rightarrow_{\pi}^{+}$ and $\rightarrow_{\pi'}$ commutes with $\rightarrow_{\mu}$)

3. Isomorphism

    - Let $\lambda\mathbf{J}'$ be $\lambda\mathbf{J}$ where π' replaces π;

    - Equip the set of μ-normal-forms with reduction rules β_μ and π_μ, where R_μ is R followed by reduction to μ-normal-form;

    - Then, the bijection between $\lambda\mathbf{J}'$-terms and μ-normal-forms is extended to an isomorphism of reduction relations

$$\begin{tikzcd} \lambda\mathbf{J}^{\mathbf{m}}_{\mu}[\beta_\mu,\pi_\mu] \arrow[r, shift right, "\nu"'] \arrow[r, phantom, "\cong"] & \lambda\mathbf{J}'[\beta,\pi'] \arrow[l, shift right, "\mu"'] \end{tikzcd}$$

4. Reduction in $\lambda\mathbf{J}^{\mathbf{m}}$ is SN for typable terms and confluent (strong normalisation and confluence of $\rightarrow_{R,\mu}$ ($R = \beta$ or π or $\beta \cup \pi$) are lifted from ΛJ).

5. Designing the multiary version of the λ-calculus with generalised application

   - In $\lambda\mathbf{J}^{\mathbf{m}}$ there is an overlap between multiarity and generality

   - $\lambda\mathbf{J}^{\mathbf{m}}_{\mu}$ and $\lambda\mathbf{J}'$ are two "isomorphic" ways of avoiding this overlap; however,

   - In $\lambda\mathbf{J}'$, multiarity is not available

   - $\lambda\mathbf{J}^{\mathbf{m}}_{\mu}$ is defined via $\lambda\mathbf{J}^{\mathbf{m}}$; the latter is much simpler to define than the former