Unity in structural proof theory and structural extensions of the $\lambda$-calculus

José Espírito Santo
Departamento de Matemática
Universidade do Minho
Portugal

4th International Workshop on Proof Theory, Computation, Complexity (PCC'05)
Lisbon, 16-17 July 2005
Motivation

Two closely related problems

1. Unity in structural proof theory

2. Structural extensions of the $\lambda$-calculus

Goal: fully realize the programme suggested by

1. von Plato’s natural deduction with general elimination rules

2. Herbelin’s “$\lambda$-calculus structure” for sequent calculus

3. JES’ isomorphism between a fragment of $LJT$ and an extension of natural deduction
Sequent calculi (left) and natural deduction systems (right)
The $\lambda^{gm}$-calculus (1)

Expressions

(Terms) $t, u, v ::= x | \lambda x. t | tl$

(Applicative contexts) $l ::= u \cdot (x)v | u :: l$

Abbreviations $[u] \equiv u \cdot (x)x$

Extracting hidden information

Match

$t(u_1 :: ... :: u_m \cdot (x)v)$ \hspace{1cm} (m \geq 1)$

with $\Psi(F, u_m \cdot (x)v)$ for some $F$, or

with $\Psi(F', u_{m-1} :: u_m \cdot (x)v)$ for some $F'$, etc.
The $\lambda^{gm}$-calculus (2)

Sequents

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

Typing rules

\[ \Gamma, x : A \vdash x : A \quad \text{Axiom} \]

\[ \Gamma, x : A \vdash t : B \quad \frac{}{\Gamma \vdash \lambda x. t : A \supset B} \quad \text{Right} \]

\[ \frac{}{\Gamma \vdash tl : B} \quad h - \text{Cut} \]

\[ \frac{}{\Gamma ; A \supset B \vdash u :: l : C} \quad lc - \text{Left} \]

\[ \frac{}{\Gamma ; A \supset B \vdash u \cdot (x)v : C} \quad l - \text{Left} \]
The $\lambda^g_m$-calculus (3)

Reduction rules

$$(\beta 1) \quad (\lambda x.t)(u \cdot (y)v) \rightarrow s(s(u, x, t), y, v)$$

$$(\beta 2) \quad (\lambda x.t)(u :: l) \rightarrow s(u, x, t)l$$

$$(\pi) \quad \Psi(F, u \cdot (x)v)l \rightarrow \Psi(F, u \cdot (x)vl)$$

$$(\mu) \quad u \cdot (x)xl \rightarrow u :: l, \text{ if } x \notin l$$

t is in $[\beta\pi\text{-nf}]$ iff every cut occurring in $t$ is of the form $xl$ (morally a left introduction)

1) For typable terms, reduction is SN

2) Any combination of the reduction rules is confluent

Proof: Because this holds of $\lambda Jm$. Think of

$$t(u_1 :: \ldots :: u_m \cdot (x)v)$$

as

$$t(u_1, u_2 :: \ldots :: u_m :: [], (x)v)$$
The $\lambda^{gm}$-calculus (4)

Permutation rules

\[(p) \quad \Psi(F, u \cdot (x)v) \rightarrow s(\Psi(F, [u]), x, v), \text{if } v \neq x\]
\[(q) \quad \Psi(F, u :: l) \rightarrow \Psi(F, [u])l\]

$t$ is in $q$-nf iff every cut occurring in $t$ is of the form $t(u \cdot (x)v)$ (morally a general application)

$t$ is in $pq$-nf iff every cut occurring in $t$ is of the form $t[u] \equiv t(u \cdot (x)x)$ (morally an ordinary application)
The \( \lambda_{gs} \)-calculus (1)

Expressions

(Terms) \( M, N, P ::= x \mid \lambda x.M \mid app(F, N, (x)P) \)

(Functions) \( F ::= hd(M) \mid FN \)

Abbreviations \( app(F, N) \equiv app(F, N, (x)x) \)

Extracting hidden information

Match

\[ app(hd(M)N_1\ldots N_{m-1}, N_m, (x)P) \quad (m \geq 1) \]

with \( \Theta(hd(M), l) \) for some \( l \), or

with \( \Theta(hd(M)N_1, l') \) for some \( l' \), etc.
The $\lambda_{gs}$-calculus (2)

Sequents

$\Gamma \vdash M : A$  \hspace{1cm} $\Gamma \gg F : A$

Typing rules

$$\Gamma, x : A \vdash x : A$$  \hspace{1cm} Assumption

$$\Gamma, x : A \vdash M : B$$  \hspace{1cm} $\Gamma \vdash \lambda x. M : A \supset B$ \hspace{1cm} Intro

$$\Gamma \gg F : A \supset B$$  \hspace{1cm} $\Gamma \vdash N : A$  \hspace{1cm} inner – Elim

$$\Gamma \gg F : A \supset B$$  \hspace{1cm} $\Gamma \vdash N : A$  \hspace{1cm} $\Gamma, x : B \vdash P : C$  \hspace{1cm} outer – Elim

$$\Gamma \vdash M : A$$  \hspace{1cm} $\Gamma \gg hd(M) : A$  \hspace{1cm} Coercion

8
The $\lambda_{gs}$-calculus (3)

Reduction rules

(\beta_1) \quad app(hd(\lambda x. M), N, (y) P) \rightarrow [[N/x]M/y]P

(\beta_2) \quad hd(\lambda x. M) N \rightarrow hd([N/x]M)

(\pi) \quad \Theta(hd(app(F, N, (x) P), l)) \rightarrow
\rightarrow app(F, N, (x) \Theta(hd(P), l))

(\mu) \quad app(F, N, (x) \Theta(hd(x), l)) \rightarrow \Theta(FN, l), \text{ if } x \notin l

$M$ is in $\beta\pi$-nf iff every coercion occurring in $M$ is of the form $hd(x)$

A derivation $D$ in $\lambda_{gs}$ is $\beta\pi$-normal iff every coercion formula occurring in $D$ is an assumption
The $\lambda_{gs}$-calculus (4)

Permutation rules

\[(p) \quad \text{app}(F, N, (x)P) \rightarrow [\text{app}(F, N)/x]P, \ P \neq x\]

\[(q) \quad FN \rightarrow \text{hd}(\text{app}(F, N))\]

$M$ is in $\underline{q}$-nf iff every $gs$-application occurrence in $M$ is of the form $\text{app}(\text{hd}(M), N, (x)P)$ (morally a general application)

$M$ is in $\underline{pq}$-nf iff every $gs$ occurring in $M$ is of the form $\text{app}(\text{hd}(M), N) \equiv \text{app}(\text{hd}(M), N, (x)x)$
Isomorphism (1)

Idea:

\[ t(u_1 :: ... :: u_m \cdot (x)v) \]

\[ \sim \]

\[ app(hd(M)N_1...N_{m-1}, N_m, (x)P) \]

as long as \( t \sim M, u_i \sim N_i \) and \( v \sim P \)

Case \( m > 1 \):

1) Sequent calculus: right-associativity, head at the surface

2) Natural deduction: left-associativity, tail at the surface

3) To each occurrence of \( u :: l \) corresponds one occurrence of \( FN \)

4) Inversion of associativity of applicative terms
Isomorphism (2)

Case $m = 1$

$$t(u \cdot (x)v)$$

$$\sim$$

$$app(hd(M), N, (x)P)$$

1) Translation between notational variants of generalised application, inducing a translation between two copies of $\Lambda J$

2) Neutral associativity, both head and tail at the surface

3) $\Lambda J$ (and hence the $\Lambda$) are neutral w.r.t. the characterization of sequent calculus and natural deduction in terms of associativity
Mappings $\Psi$ and $\Theta$ are sound, mutually inverse bijections. For each $R \in \{\beta_1, \beta_2, \pi, \mu, p, q\}$:

1) $M \rightarrow_R M'$ in $\lambda_{gs}$ iff $\Psi M \rightarrow_R \Psi M'$ in $\lambda_{gm}$.

2) $t \rightarrow_R t'$ in $\lambda_{gm}$ iff $\Theta t \rightarrow_R \Theta t'$ in $\lambda_{gs}$.

Corollary: $\lambda_{gs}$ inherits the properties of $\lambda_{gm}$.
Mappings $\Psi$ and $\Theta$ translate between two copies of $\Lambda J$.

$\Psi(app(hd(M), N, (x)P)) = (\Psi M)(\Psi N \cdot (x)\Psi P)$

$\Theta(t(u \cdot (x)v)) = app(\Theta(t), \Theta(u), (x)\Theta(v))$
Roadmap again

\[ \begin{aligned}
\lambda^G & \quad \Theta \quad \psi \quad \lambda_N \\
\lambda^g_{gm} & \quad \Theta \quad \psi \quad \lambda^g_{gs} \\
\lambda^g & \quad \Theta \quad \psi \quad \lambda^g \\
& \quad \quad \quad G \quad \Lambda J
\end{aligned} \]
The $\lambda^G$-calculus (1)

Expressions

(Terms) \[ t, u, v := x | \lambda x.t | tk \]

(Contexts) \[ k := (x)v | u :: k \]

Abbreviations \[ [u] = u :: (z)z \]

Typing rules

\[
\begin{align*}
\Gamma, x : A & \vdash x : A \quad \text{Axiom} \\
\Gamma, x : A & \vdash t : B \\
\Gamma & \vdash \lambda x.t : A \supset B \quad \text{Right} \\
\Gamma & \vdash u : A \\
\Gamma & \vdash k : C \\
\Gamma ; A \supset B & \vdash u :: k : C \quad \text{Left} \\
\Gamma & \vdash t : A \\
\Gamma & \vdash k : B \\
\Gamma & \vdash tk : B \quad \text{Cut} \\
\Gamma, x : A & \vdash v : B \\
\Gamma ; A & \vdash (x)v : B \quad \text{Selection}
\end{align*}
\]
The $\lambda^G$-calculus (2)

Reduction rules

\begin{align*}
(\sigma) \quad t((x)v) & \rightarrow s(t, x, v) \\
(\beta) \quad (\lambda x.t)(u :: k) & \rightarrow u((x)t)k \\
(\pi) \quad \Psi(E, (x)v)(u :: k) & \rightarrow \Psi(E, (x)v(u :: k)) \\
(\mu) \quad (x) xk & \rightarrow k, \text{ if } x \notin k
\end{align*}

Permutation rules

\begin{align*}
(p) \quad \Psi(EN, (x)v) & \rightarrow s(\Psi(E, [\Psi N]), x, v), \text{ if } v \neq x \\
(q) \quad \Psi(EN, u :: k) & \rightarrow \Psi(E, [\Psi N])(u :: k)
\end{align*}

$t$ is in $pq\text{-nf}$ iff every cut occurring in $t$ is of the form $t[u] \equiv t(u :: (x)x)$ or $t((x)v)$. Hence, $t$ is in $pq\text{-nf}$ iff $t$ is morally a $\lambda x$-term.
The $\lambda_N$-calculus (1)

**Expressions**

(Terms) $M, N, P ::= x | \lambda x. M | \{E/x\} P$

(FAP-Expressions) $E ::= h_d(M) | EN$

**Abbreviations** $ap(E) = \{E/z\}z$

**Typing rules**

\[
\begin{align*}
\Gamma, x : A &\vdash x : A & \text{Assumption} \\
\Gamma, x : A &\vdash M : B \quad \Gamma \vdash \lambda x. M : A \supset B & \text{Intro} \\
\Gamma \triangleright E : A \supset B &\quad \Gamma \vdash N : A \quad \Gamma \vdash EN : B & \text{Elim} \\
\Gamma \triangleright E : A &\quad \Gamma, x : A \vdash P : B \quad \Gamma \vdash \{E/x\} P : B & \text{Subst} \\
\Gamma \vdash M : A & \quad \Gamma \triangleright h_d(M) : A & \text{Coercion}
\end{align*}
\]
The $\lambda_N$-calculus (2)

Reduction rules

$$(\sigma) \quad \{\text{hd}(M)/x\}P \rightarrow \text{[M/x]} \text{P}$$
$$(\beta) \quad \text{hd}(\lambda x. M)N \rightarrow \text{hd}\{\text{hd}(N)/x\}M$$

$$(\pi) \quad \Theta(\text{hd}\{E/x\}P), u :: k) \rightarrow \rightarrow \{E/x\}\Theta(\text{hd}(P), u :: k)$$
$$(\mu) \quad \{E/x\}\Theta(\text{hd}(x), k) \rightarrow \Theta(E, k), \text{ if } x \notin k$$

Permutation rules

$$(p) \quad \{\text{EN}/x\}P \rightarrow \text{[ap(EN)/x]}P, \text{ if } P \neq x$$
$$(q) \quad \text{ENN}^\prime \rightarrow \text{hd(ap(EN))}N^\prime$$

A derivation $\mathcal{D}$ in $\lambda_N$ is $\beta\pi\sigma$-normal iff every coercion formula occurring in $\mathcal{D}$ is an assumption and the main premiss of an elimination

$M$ is in $pq$-nf iff every substitution occurring in $M$ is of the form $\text{ap(hd}(M)M) \equiv \{\text{hd}(M)N/x\}x$ or $\{\text{hd}(M)/x\}P$. Hence, $M$ is in $pq$-nf iff $M$ is morally a $\lambda x$-term.
Decompositions of general elimination

\[ \Gamma \vdash M : A \supset B \quad \Gamma \vdash N : A \quad \Gamma, x : B \vdash P : C \]
\[ \Gamma \vdash M(N, x.P) : C \]

From \( \Lambda J = \lambda^g \subset \lambda^{gm} \subset \lambda^G \)

\[ \Gamma \vdash t : A \supset B \quad \Gamma \vdash u : A \quad \Gamma ; B \vdash (x)v \]
\[ \Gamma \vdash t(u :: (x)v) : C \]

(1) Selection (2) Left (3) Cut

From \( \Lambda J = \lambda^g \subset \lambda^{gs} \subset \lambda_N \)

\[ \Gamma \vdash M : A \supset B \]
\[ \Gamma \triangleright hd(M) : A \supset B \]
\[ \Gamma \vdash N : A \]
\[ \Gamma \triangleright hd(M)N : B \]
\[ \Gamma \vdash \{hd(M)N/x\}P : C \]

(1) Coercion (2) Elim (3) Subst
Conclusions and future work

1) Natural deduction isomorphic to full sequent calculus

2) Both proof system presented as (meaningful) extensions of the simply typed $\lambda$-calculus

3) Difference reduced to associativity of applicative terms

4) Isomorphism: at the logical level

   left-introduction $\sim$ elimination
   cut $\sim$ primitive substitution

5) Isomorphism: at the $\lambda$-calculi level, inversion of associativity

6) Future work: unification of sequent calculus and natural deduction