Regaining confluence in λ^{Gtz}-calculus

Jelena Ivetić

Faculty of Technical Sciences, University of Novi Sad, Serbia
jelena@imft.ftn.uns.ac.rs

CALCOjnr 2009, Udine, September 2009

Outline

1. Motivation
2. λ^{Gtz}-calculus
   - Untyped λ^{Gtz}
   - Typed λ^{Gtz}
3. Confluence
   - Non-confluence of λ^{Gtz}
   - Regaining confluence

Curry-Howard correspondence

Intuitionistic natural deduction and simply typed λ-calculus are corresponding in the following ways:

- types of the closed λ-terms correspond to the theorems of implicational fragment of intuitionistic logic;
- type assignment corresponds to the proof of the theorem in the formal system *ND*;
- term reduction corresponds to the proof normalization in the system *ND*.

Expanding C-H...

- Natural deduction $\leftrightsquigarrow$ λ-calculus;
- Hilbert's axiomatic system $\leftrightsquigarrow$ combinators;
- Sequent calculus $\leftrightsquigarrow$???

Sequent calculus - LJ

$(axiom)$ $$\frac{}{\Gamma, A \vdash A}$$

$(\rightarrow_{left})$ $$\frac{\Gamma \vdash A \quad \Gamma, B \vdash C}{\Gamma, A \rightarrow B \vdash C}$$

$(\rightarrow_{right})$ $$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}$$

(cut) $$\frac{\Gamma \vdash A \quad \Gamma, A \vdash B}{\Gamma \vdash B}$$

$\bar{\lambda}$-calculus (Herbelin, 1995)

Syntax:

$$
\begin{array}{lrcl}
\text{(Terms)} & t, u, v & ::= & xl \,|\, \lambda x.t \,|\, tl \,|\, t\langle x = v\rangle \\
\text{(Lists)} & l, l' & ::= & [\,] \,|\, t :: l \,|\, l@l' \,|\, l\langle x = t\rangle
\end{array}
$$

Reductions:

$$
\begin{array}{rrcl}
(\beta_{cons}) & \lambda x.u(v :: l) & \rightarrow & u\langle x = v\rangle l \\
(\beta_{nil}) & \lambda x.u[\,] & \rightarrow & \lambda x.u \\
(C_{var}) & (tl)l' & \rightarrow & t(l@l') \\
(C_{cons}) & (t :: l)@l' & \rightarrow & t :: (l@l') \\
(C_{nil}) & [\,]@l & \rightarrow & l \\
(S_{yes}) & (xl)\langle x = v\rangle & \rightarrow & vl\langle x = v\rangle \\
(S_{no}) & (yl)\langle x = v\rangle & \rightarrow & yl\langle x = v\rangle \\
(S_{\lambda}) & (\lambda y.u)\langle x = v\rangle & \rightarrow & \lambda y.(u\langle x = v\rangle) \\
(S_{nil}) & [\,]\langle x = v\rangle & \rightarrow & [\,] \\
(S_{cons}) & (u :: l)\langle x = v\rangle & \rightarrow & u\langle x = v\rangle :: l\langle x = v\rangle.
\end{array}
$$

Simply typed $\bar{\lambda}$

$$\frac{}{\Gamma; . : A \vdash (\,.[\,]) : A}\,(Ax) \qquad \frac{\Gamma, x : A; . : A \vdash (\,.l) : B}{\Gamma, x : A; \vdash xl : B}\,(Cont)$$

$$\frac{\Gamma, x : A; \vdash t : B}{\Gamma; \vdash \lambda x.t : A \to B}\,(\to_R) \qquad \frac{\Gamma; \vdash t : A \quad \Gamma; . : B \vdash (\,.l) : C}{\Gamma; . : A \to B \vdash (\,.(t :: l)) : C}\,(\to_L)$$

$$\frac{\Gamma; \vdash t : A \quad \Gamma; . : A \vdash (\,.l) : B}{\Gamma; \vdash tl : B}\,(C_{H1}) \qquad \frac{\Gamma; . : C \vdash (\,.l) : A \quad \Gamma; . : A \vdash (\,.l') : B}{\Gamma; . : C \vdash (\,.l @ l') : B}\,(C_{H2})$$

$$\frac{\Gamma; \vdash t : A \quad \Gamma, x : A; \vdash u : B}{\Gamma; \vdash u\langle x = t\rangle : B}\,(C_{M1}) \qquad \frac{\Gamma; \vdash t : A \quad \Gamma, x : A; . : C \vdash (\,.l) : B}{\Gamma; . : C \vdash (\,.l\langle x = t\rangle) : B}\,(C_{M2})$$

C-H correspondence...

...is satisfied, but not completely!

Normal forms of the $\bar{\lambda}$-calculus correspond to cut-free proofs of the *LJT* (restricted *LJ*).

The syntax

- Proposed by Espirito Santo [2006];
- fully corresponds to intuitionistic sequent calculus (with cut rule).

The syntax:

$$\begin{array}{lccl} \text{(Terms)} & t & ::= & x \,|\, \lambda x.t \,|\, tk \\ \text{(Contexts)} & k & ::= & \widehat{x}.t \,|\, t :: k \end{array}$$

- term - a variable, an abstraction or an application (*cut*);
- context - a selection or a context constructor (*cons*);
- terms and contexts are together called expressions, denoted by E;
- $\widehat{x}.x$ represents an empty list.

Reduction rules:

$$\begin{array}{llcl}
(\beta) & (\lambda x.t)(u :: k) & \rightarrow & u\widehat{x}.(tk) \\
(\pi) & (tk)k' & \rightarrow & t(k@k') \\
(\sigma) & t\widehat{x}.v & \rightarrow & v[x := t] \\
(\mu) & \widehat{x}.xk & \rightarrow & k, \text{ ako } x \notin k
\end{array}$$

- $v[x := t]$ is meta-substitution;
- $k@k'$ is defined with:

$$(u :: k)@k' = u :: (k@k') \qquad (\widehat{x}.t)@k' = \widehat{x}.tk'.$$

Normal forms:

$$\begin{array}{llcl}
\text{(Terms)} & t_{nf} & = & x_{nf} \mid \lambda x.t_{nf} \mid x(t_{nf} :: k_{nf}) \\
\text{(Contexts)} & k_{nf} & = & \widehat{x}.t_{nf} \mid t_{nf} :: k_{nf}.
\end{array}$$

Example

Reducing $T : (\lambda x.y)(y(\widehat{z}.z) :: \widehat{u}.\lambda y.u)$.

I way:

$$
\begin{array}{lll}
T & \to_\beta & (y\widehat{z}.z)\widehat{x}.(y\widehat{u}.\lambda y.u) \\
 & \to_\pi & y\,((\widehat{z}.z)@\widehat{x}.(y\widehat{u}.\lambda y.u)) \\
 & \leftrightarrow & y\widehat{z}.(z\widehat{x}.(y\widehat{u}.\lambda y.u) \\
 & \to_\sigma & y\widehat{z}.\ y\widehat{u}.(\lambda y.u)[x := z] \\
 & \leftrightarrow & y\widehat{z}.(y\widehat{u}.\lambda y.u) \\
 & \to_\sigma & y\widehat{z}.\ (\lambda y.u)[u := y] \\
 & \leftrightarrow & y\widehat{z}.\ (\lambda y'.u)[u := y] \\
 & \leftrightarrow & y\widehat{z}.\lambda y'.y \\
 & \to_\sigma & (\lambda y'.y)[z := y] \\
 & \leftrightarrow & \lambda y'.y.
\end{array}
$$

Example

II way:

$$\begin{array}{lll} T & \to_\beta & (y\widehat{z}.z)\widehat{x}.(y\widehat{u}.\lambda y.u) \\ & \to_\sigma & (y\widehat{u}.\lambda y.u)[x := y\widehat{z}.z] \\ & \leftrightarrow & y\widehat{u}.\lambda y.u \\ & \to_\sigma & (\lambda y'.u)[u := y] \\ & \leftrightarrow & \lambda y'.y. \end{array}$$

Call-by-value and Call-by-name!

Properties

What makes a formal calculus suitable for implementation?

- Confluence;
- Preservation of type under reduction (Subject reduction);
- Strong normalization;
- Characterization of strong normalization.

Simply typed λ^{Gtz}

$$\frac{}{\Gamma, x : A \vdash x : A}\,(Ax) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,(\to_R)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma; B \vdash k : C}{\Gamma; A \to B \vdash t :: k : C}\,(\to_L) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma; A \vdash \widehat{x}.t : B}\,(Sel)$$

$$\frac{\Gamma \vdash t : A \quad \Gamma; A \vdash k : B}{\Gamma \vdash tk : B}\,(Cut)$$

- Properties: subject reduction; strong normalization.

λ^{Gtz} with intersection types

$$\frac{}{\Gamma, x : \cap A_i \vdash x : A_i \quad i \geq 1}\,(Ax) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\,(\to_R)$$

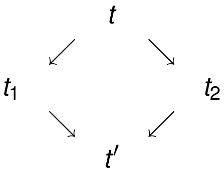
$$\frac{\Gamma \vdash t : A_i \quad i = 1, ..., n \ \Gamma; B \vdash k : C}{\Gamma; \cap A_i \to B \vdash t :: k : C}\,(\to_L) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma; A \vdash \widehat{x}.t : B}\,(Sel)$$

$$\frac{\Gamma \vdash t : A_i, \ \forall i \ \Gamma; \cap A_i \vdash k : B}{\Gamma \vdash tk : B}\,(Cut)$$

- Properties: subject reduction; strong normalization; characterization of SN.

Confluence

The confluence (or Church-Rosser property) is one of the desired properties of a formal calculus.



- In the diagram, arrows stand for multiple occurrences of the union of all reductions of the particular formal calculus.

λ^{Gtz}-calculus is not confluent, unlike most of intuitionistic formal calculi. The reason is a critical pair, consisting of π and σ reductions.

$$(tk)(\widehat{x}.t')$$

$$t(k@\widehat{x}.t') \;\swarrow_{\pi} \qquad \searrow_{\sigma}\; t'[x := tk]$$

- In a lot of particular cases these two terms can not be reduced to the same normal form.

There are two basic directions in solving this problem:

- eliminate the critical pair by restricting the reduction rules;
- expand the syntax and create an appropriate type assignment system.

"Call-by-value" sub-calculus

- obtained by forbidding σ reduction to perform on the term $(tk)(\widehat{x}.v)$;
- the syntax of λ^{Gtz}_V:

$$\begin{array}{lccl} \text{Values} & T & ::= & x \,|\, \lambda x.t \\ \text{Terms} & t & ::= & T \,|\, tk \\ \text{Contexts} & k & ::= & \widehat{x}.t \,|\, t :: k \end{array}$$

- introduces *values* as a syntactic category;
- reduction rules of λ^{Gtz}_V are β, π, μ of λ^{Gtz} and

$$(\sigma_V) \quad T(\widehat{x}.t) \quad \rightarrow \quad t[x := T].$$

"Call-by-name" sub-calculus

- obtained by forbidding π reduction to perform on the term $(tk)(\widehat{x}.v)$;
- the syntax of λ_L^{Gtz}:

$$\begin{array}{lrcl} \text{Terms} & t & ::= & x \,|\, \lambda x.t \,|\, tk \\ \text{Lists} & l & ::= & \widehat{x}.x \,|\, t :: l \\ \text{Contexts} & k & ::= & l \,|\, \widehat{x}.t \end{array}$$

- introduces *lists* as a syntactic category;
- reduction rules of λ_L^{Gtz} are β, σ, μ of λ^{Gtz} and

$$(\pi_L) \quad (tk)l \quad \rightarrow \quad t(k@l).$$

- $(tk)(\widehat{x}.x)$ is at the same time σ-redex and (π_L)-redex, but applying each of these two reductions leads to the result tk.

Proving confluence

- after elimination of the critical pair, we can prove confluence;
- we use parallel reductions method;
- developed by Takahashi (1995);
- adapted by Likavec (2004) for proving confluence of $\lambda\mu\tilde{\mu}$ sub-calculi;
- we will sketch the proof for the confluence of λ_V^{Gtz}.

Parallel reductions for λ_V^{Gtz}

$$\frac{}{x \Rightarrow x}\ (g1) \qquad \frac{t \Rightarrow t'}{\lambda x.t \Rightarrow \lambda x.t'}\ (g2) \qquad \frac{t \Rightarrow t',\ k \Rightarrow k'}{tk \Rightarrow t'k'}\ (g3)$$

$$\frac{t \Rightarrow t'}{\widehat{x}.t \Rightarrow \widehat{x}.t'}\ (g4) \qquad \frac{t \Rightarrow t',\ k \Rightarrow k'}{t :: k \Rightarrow t' :: k'}\ (g5)$$

$$\frac{t \Rightarrow t',\ u \Rightarrow u',\ k \Rightarrow k'}{(\lambda x.t)(u :: k) \Rightarrow u'\widehat{x}.(t'k')}\ (g6) \qquad \frac{T \Rightarrow T',\ t \Rightarrow t'}{T(\widehat{x}.t) \Rightarrow t'[x := T']}\ (g7)$$

$$\frac{t \Rightarrow t',\ k \Rightarrow k',\ k_1 \Rightarrow k_1'}{(tk)k_1 \Rightarrow t'(k'@k_1')}\ (g8) \qquad \frac{k \Rightarrow k'}{\widehat{x}.xk \Rightarrow k'}\ (g9)$$

Properties of $\Rightarrow$

(i) For all E, $E \Rightarrow E'$.

(ii) If $E \longrightarrow E'$ then $E \Rightarrow E'$.

(iii) If $E \Rightarrow E'$ then $E \twoheadrightarrow E'$.

(iv) If $E \Rightarrow E'$ and $F \Rightarrow F'$, then
$E\langle x := F\rangle \Rightarrow E'\langle x := F'\rangle$.

G^*

Expression E^* is obtained from E by simultaneous reducing of all existing redexes of E.

(∗1) $x^* \equiv x$

(∗2) $(\lambda x.t)^* \equiv \lambda x.t^*$

(∗3) $(\widehat{x}.t)^* \equiv \widehat{x}.t^*$

(∗4) $(t :: k)^* \equiv t^* :: k^*$

(∗5) $(tk)^* \equiv t^* k^*$ if $tk \neq (\lambda x.v)(u :: k_1)$ and $tk \neq T(\widehat{x}.v)$ i $tk \neq (uk_1)k_2$

(∗6) $((\lambda x.t)(u :: k))^* \equiv u^*(\widehat{x}.t^* k^*)$

(∗7) $(T(\widehat{x}.v))^* \equiv v^*[x := T^*]$

(∗8) $((tk)k_1)^* \equiv t^*(k^* @ k_1^*)$.

Theorem (Star-property of $\Rightarrow$)

If $E \Rightarrow E'$, then $E' \Rightarrow E^*$.

Theorem (Diamond-property of $\Rightarrow$)

If $E_1 \Leftarrow E \Rightarrow E_2$, then $E_1 \Rightarrow E' \Leftarrow E_2$ for some G'.

Theorem (Confluence of λ_V^{Gtz}-calculus)

If $E_1 \twoheadleftarrow E \twoheadrightarrow E_2$, then $E_1 \twoheadrightarrow E' \twoheadleftarrow E_2$ for some E'.

Ongoing and future work

- further investigation of the λ_V^{Gtz} and λ_L^{Gtz};
- expanding the syntax of λ^{Gtz} with explicit structural rules.