Computational Interpretation for Natural Deduction plus cut and structural rules

The $\lambda\mathtt{lxr}$-calculus

$$
\begin{array}{lrll}
(Terms) & t, u ::= & x & variable \\
 & | & \lambda x.t & abstraction \\
 & | & t\ u & application \\
 & | & t[x/u] & substitution \\
 & | & W_x(t) & weakening \\
 & | & C_x^{y,z}(t) & contraction
\end{array}
$$

We only consider *well-formed* terms :

– Linearity

– Compulsory presence

– Barendregt's convention

Typing Rules for the λlxr-calculus

$$\frac{}{x : A \vdash x : A}\ (ax) \qquad \frac{\Delta \vdash u : B \quad \Gamma, x : B \vdash t : A}{\Gamma, \Delta \vdash t[x/u] : A}\ (cut)$$

$$\frac{\Gamma \vdash t : A \to B \quad \Delta \vdash u : A}{\Gamma, \Delta \vdash (t\ u) : B}\ (\to e) \qquad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\ (\to i)$$

$$\frac{\Gamma, x : A, y : A \vdash t : B}{\Gamma, z : A \vdash C_z^{x,y}(t) : B}\ (c) \qquad \frac{\Gamma \vdash t : A}{\Gamma, x : B \vdash W_x(t) : A}\ (w)$$

Congruence I

AC of contraction :

$$\begin{array}{lcll} C_w^{x,v}(C_x^{z,y}(t)) & \equiv & C_w^{x,y}(C_x^{z,v}(t)) & \text{if } x \neq y, v \\ C_x^{y,z}(t) & \equiv & C_x^{z,y}(t) & \\ C_{x'}^{y',z'}(C_x^{y,z}(t)) & \equiv & C_x^{y,z}(C_{x'}^{y',z'}(t)) & \text{if } x \neq y', z' \ \&\ x' \neq y, z \end{array}$$

C of weakening :

$$W_x(W_y(t)) \equiv W_y(W_x(t))$$

Congruence II

Commutativity of substitutions :

$$t[x/u][y/v] \equiv t[y/v][x/u] \text{ if } y \notin FV(u) \;\&\; x \notin FV(v)$$

Contraction and substitution have the same status :

$$C_w^{y,z}(t)[x/u] \equiv C_w^{y,z}(t[x/u]) \text{if } x \neq w \;\&\; y, z \notin FV(u)$$

Reduction Rules for the $\lambda\texttt{lxr}$-calculus

$$(B) \quad (\lambda x.t)\ u \to\ t[x/u]$$

SubSystem `x`

$$
\begin{array}{llcll}
(Abs) & (\lambda y.t)[x/u] & \rightarrow & \lambda y.t[x/u] & \\
(App1) & (t\ v)[x/u] & \rightarrow & t[x/v]\ v & \text{if } x \in FV(t) \\
(App2) & (t\ v)[x/u] & \rightarrow & t\ v[x/u] & \text{if } x \in FV(v) \\
(Var) & x[x/u] & \rightarrow & u & \\
(Weak1) & W_x(t)[x/u] & \rightarrow & W_{FV(u)}(t) & \\
(Weak2) & W_y(t)[x/u] & \rightarrow & W_y(t[x/u]) & \text{if } x \neq y \\
(Cont1) & C_x^{y,z}(t)[x/u] & \rightarrow & C_\Phi^{\Delta,\Pi}(t[y/u_1][z/u_2]) & \\
 & & & \text{where } \Phi := FV(u) & \\
 & & & u_1 = R_\Delta^\Phi(u) \text{ and } u_2 = R_\Pi^\Phi(u) & \\
(Comp) & t[y/v][x/u] & \rightarrow & t[y/v[x/u]] \text{ if } x \in FV(v) &
\end{array}
$$

SubSystem `t`

$$
\begin{array}{llcll}
(WAbs) & \lambda x.W_y(t) & \rightarrow & W_y(\lambda x.t) & x \neq y \\
(WApp1) & W_y(u)\ v & \rightarrow & W_y(u\ v) & \\
(WApp2) & u\ W_y(v) & \rightarrow & W_y(u\ v) & \\
(WSubs) & t[x/W_y(u)] & \rightarrow & W_y(t[x/u]) & \\
(Merge) & C_w^{y,z}(W_y(t)) & \rightarrow & R_w^z(t) & \\
(Cross) & C_w^{y,z}(W_x(t)) & \rightarrow & W_x(C_w^{y,z}(t)) & x \neq y,\ \ x \neq z \\
(CAbs) & C_w^{y,z}(\lambda x.t) & \rightarrow & \lambda x.C_w^{y,z}(t) & \\
(CApp1) & C_w^{y,z}(t\ u) & \rightarrow & C_w^{y,z}(t)\ u & y,z \in FV(t) \\
(CApp2) & C_w^{y,z}(t\ u) & \rightarrow & t\ C_w^{y,z}(u) & y,z \in FV(u) \\
(CSubs) & C_w^{y,z}(t[x/u]) & \rightarrow & t[x/C_w^{y,z}(u)] & y,z \in FV(u)
\end{array}
$$

The reduction relation $\lambda\mathtt{lxr}$

The reduction relation is generated by the previous rewriting rules and congruence axioms :

$$t \longrightarrow_{\lambda\mathtt{lxr}} t' \text{ iff } \exists t_1, t_2 \; t \equiv t_1 \longrightarrow_{B+\mathtt{x}+\mathtt{t}} t_2 \equiv t'$$

Properties of the $\lambda\mathtt{lxr}$-calculus

1. (Full composition) $t[x/v] \longrightarrow^* t\{x = v\}$
   even when t contains non-evaluated substitutions
2. (Free variables are preserved) If $t \longrightarrow_{\lambda\mathtt{lxr}} t'$, then
   $FV(t) = FV(t')$
3. (Subject reduction) If [illegible] $t' : A$.
   confluent and terminating
4. (Convergence) $\mathtt{xt} = \mathtt{x} \cup \mathtt{t}$ is convergent.
   Which is the form of a term in $\mathtt{xt}$-normal form ?

Connexion with λ-calculus

$\mathcal{B}()$ hides resource control

$$t \to^*_{\texttt{xt}} W_\Pi(\mathcal{A}(\mathcal{B}(t))) \qquad \lambda\texttt{lxr} \; \underset{\mathcal{A}()}{\overset{\mathcal{B}()}{\rightleftarrows}} \; \lambda \qquad t = \mathcal{B}(\mathcal{A}(t))$$

$\mathcal{A}()$ introduces resource operators

Example : $\lambda x.y\ y$ is represented by $\lambda x.W_x(C_y^{z,z'}(z\ z'))$

From $\lambda\texttt{lxr}$ to λ

$$\begin{array}{lcl}
\mathcal{B}(x) & = & x \\
\mathcal{B}(\lambda x.t) & = & \lambda x.\mathcal{B}(t) \\
\mathcal{B}(W_x(t)) & = & \mathcal{B}(t) \\
\mathcal{B}(C_x^{y,z}(t)) & = & \mathcal{B}(t)\{y/x\}\{z/x\} \\
\mathcal{B}(t\ u) & = & \mathcal{B}(t)\ \mathcal{B}(u) \\
\mathcal{B}(t[x/u]) & = & \mathcal{B}(t)\{x/\mathcal{B}(u)\}
\end{array}$$

Relating $\lambda\mathtt{lxr}$ and λ

Proposition [Simulating $\lambda\mathtt{lxr}$-reductions]

1. If $M \equiv N$, then $\mathcal{B}(M) = \mathcal{B}(N)$.
2. If $M \to_B N$, then $\mathcal{B}(M) \to_\beta^* \mathcal{B}(N)$.
3. If $M \to_{\mathtt{xt}} N$, then $\mathcal{B}(M) = \mathcal{B}(N)$.

Corollary $M \to_{\lambda\mathtt{lxr}} N$, then $\mathcal{B}(M) \to_\beta^* \mathcal{B}(N)$.

From λ to $\lambda\mathtt{lxr}$

$$
\begin{array}{lcll}
\mathcal{A}(x) & := & x & \\
\mathcal{A}(\lambda x.t) & := & \lambda x.\mathcal{A}(t) & \text{if } x \in FV(t) \\
\mathcal{A}(\lambda x.t) & := & \lambda x.W_x(\mathcal{A}(t)) & \text{if } x \notin FV(t) \\
\mathcal{A}(tu) & := & C_\Phi^{\Delta,\Pi}(R_\Delta^\Phi(\mathcal{A}(t)))\ R_\Pi^\Phi(\mathcal{A}(u)) & \text{where } \Phi := FV(t) \cap FV(u)
\end{array}
$$

Relating λ and $\lambda\mathtt{lxr}$

Lemma For all λ-terms t and u such that $x \in FV(t)$, we have

$$C_{\Phi}^{\Delta,\Pi}(R_{\Delta}^{\Phi}(\mathcal{A}(t))[x/R_{\Pi}^{\Phi}(\mathcal{A}(u))]) \rightarrow_{\mathtt{xt}}^{*} \mathcal{A}(t\{x/u\})$$

where $\Phi := (FV(t) \setminus \{x\}) \cap FV(u)$.

Corollary If $t \rightarrow_{\beta} t'$, then $\mathcal{A}(t) \rightarrow^{+}_{\lambda\mathtt{lxr}} W_{FV(t)\setminus FV(t')}(\mathcal{A}(t'))$.

Exemple $t = (\lambda x.y)z \rightarrow_{\beta} y = t'$ and
$\mathcal{A}(t) = (\lambda x.W_x(y))z \rightarrow^{+}_{\lambda\mathtt{lxr}} W_z(\mathcal{A}(y))$.

Confluence

Lemma The $\mathtt{xt}$-normal form of t is $W_{FV(t)\setminus FV(\mathcal{B}(t))}(\mathcal{A}(\mathcal{B}(t)))$.

Exemple Let $t = C_x^{x_1,x_2}((\lambda y.x_1\ (x_2\ W_y(z)))\ W_k(w))$. Then
$\mathtt{xt}(t) = W_k((\lambda y.W_y(C_x^{x_1,x_2}(x_1\ (x_2\ z))))\ w)$.

Théorème [Confluence modulo] The reduction relation $\lambda\mathtt{lxr}$ is confluent (even on terms with meta-variables).

More Properties of the $\lambda\mathtt{lxr}$-calculus

1. (Preservation of typing)

   (a) If $\Gamma \vdash_{\lambda} t : A$ then $\Gamma \vdash_{\lambda\mathtt{lxr}} W_{\Gamma \setminus FV(t)}(\mathcal{A}(t)) : A$

   (b) If $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$ then $\Gamma \vdash_{\lambda} \mathcal{B}(t) : A$

2. (PSN) If $M \in SN^{\beta}$ then $\mathcal{A}(M) \in SN^{\lambda\mathtt{lxr}}$.

   breaks Melliès' counter-example of non-termination
   (with $t[y/v][x/u] \rightarrow t[y/v[x/u]]$ if $x \notin t$)

Strong Normalization and Proof Nets

Translating types :

$$\begin{array}{lcll} A^* & = & A & \text{for atomic types} \\ (A \rightarrow B)^* & = & ?((A^*)^{\perp}) \parr B^* & \text{otherwise} \end{array}$$

Translating terms :

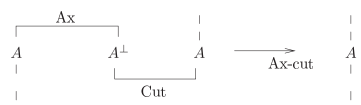
$T(B_1, \ldots, B_n \vdash t : A)$ gives a proof-net having wires labelled with $?((B_1^*)^{\perp})$, ..., $(?(B_n^*)^{\perp})$, A^*.

Proof Nets

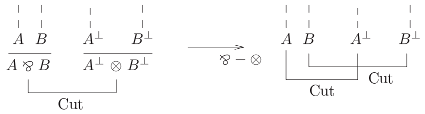
Proof nets with

- Cut elimination rules (6 rules)
- Elim reduction rule
- Pull reduction rule
- Axiom for associativity of contraction
- Axiom for contraction-box

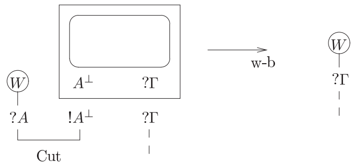
Cut elimination for MELL proof nets (1)



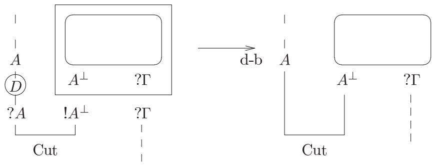
Cut elimination for MELL proof nets (2)



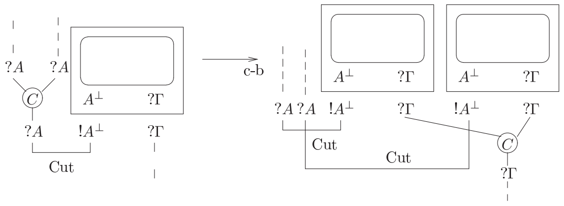
Cut elimination for MELL proof nets (3)



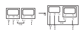
Cut elimination for MELL proof nets (4)



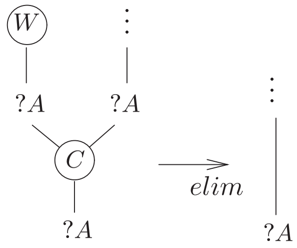
Cut elimination for MELL proof nets (5)



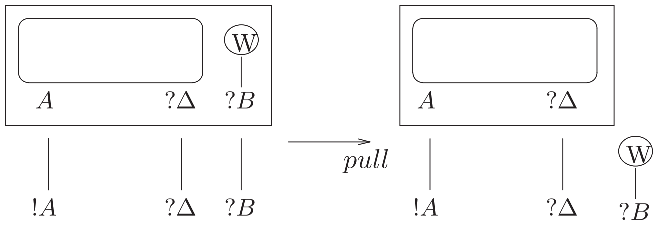
Cut elimination for MELL proof nets (6)



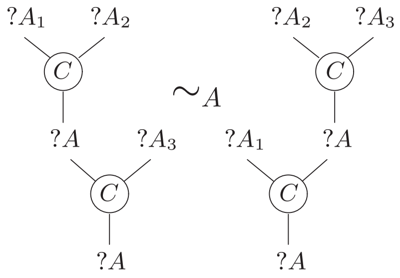
Elim reduction rule



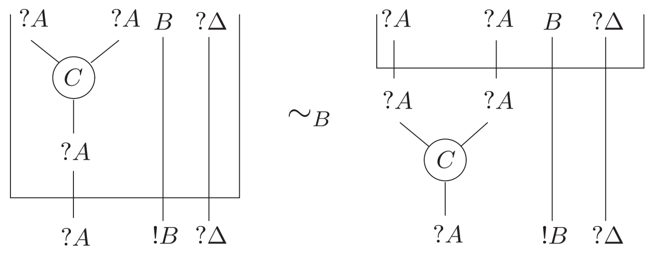
Pull reduction rule



Axiom for associativity of contraction



Axiom for contraction-box



Reduction relation modulo for proof-nets

Let us consider :

$$E = A \cup B$$
$$R = \{Ax\text{-}cut, \parr\text{-}\otimes, w\text{-}b, d\text{-}b, c\text{-}b, b\text{-}b, elim, pull\}$$

We consider the reduction relation on proof-nets generated by the reduction rules R and congruence axioms E :

$$p \rightarrow_{R/E} p' \text{ iff } \exists p_1, p_2\ p \sim_E p_1 \rightarrow_R p_2 \sim_E p'$$

Termination properties of proof-nets

(Girard)

The cut elimination rules are SN.

(DiCosmo-Guerrini)

The cut elimination rules + pull + axioms are SN.

(Polonovski)

The reduction relation R/E is SN.

Simulating $\lambda\mathtt{lxr}$ with Proof Nets

Lemma Let t be a $\lambda\mathtt{lxr}$-typed term.

- If $t \equiv t'$, then $T(t) \sim_E T(t')$.
- If $t \to_B t'$, then $T(t) \to^+_{R/E} T(t')$.
- If $t \to_{\mathtt{xt}} t'$, then $T(t) \to^*_{R/E} T(t')$.

So that $\lambda\mathtt{lxr}$ is sound w.r.t proof-nets :
If t is $\lambda\mathtt{lxr}$-typed, then $t \to_{\lambda\mathtt{lxr}} t'$ implies $T(t) \to^*_{R/E} T(t')$.

Théorème [Strong Normalisation] The relation $\lambda\mathtt{lxr}$ is strongly normalising on well-typed $\lambda\mathtt{lxr}$-terms.

Towards completeness

- Define a congruence $\approx$ for proof-nets.
- Define a congruence $\cong$ for $\lambda\mathtt{lxr}$-terms.
- Show that $T(t_1) \approx T(t_2)$ implies $t_1 \cong t_2$.

Summary

The $\lambda\mathtt{lxr}$-calculus is a computational interpretation of natural deduction plus cut and structural rules enjoying the following properties :

- Confluence on all the terms.
- Simulation of one-step β-reduction.
- Preservation of β-strong normalization.
- Strong normalization of well-typed terms.
- Admits extensions with (full)composition rules.
- Sound and complete with respect to proof-nets.
- Explicit operators for implementation issues.