# From MELL Proof-Nets to Explicit Substitution Calculi

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

| | | | |
|---|---|---|---|
| $(Terms)$ | $t, u ::=$ | $x$ | *variable* |
| | $\mid$ | $\lambda x.t$ | *abstraction* |
| | $\mid$ | $t\ u$ | *application* |
| | $\mid$ | $t[x/u]$ | *substitution* |
| | $\mid$ | $W_x(t)$ | *weakening* |
| | $\mid$ | $C_x^{y,z}(t)$ | *contraction* |

**Free Variables:**

$$
\begin{array}{lllllll}
\mathbf{fv}(x) & := & \{x\} & \quad & \mathbf{fv}(\lambda x.t) & := & \mathbf{fv}(t) \setminus \{x\} \\
\mathbf{fv}(tu) & := & \mathbf{fv}(t) \cup \mathbf{fv}(u) & & \mathbf{fv}(t[x/u]) & := & (\mathbf{fv}(t) \setminus \{x\}) \cup \mathbf{fv}(u) \\
\mathbf{fv}(W_x(t)) & := & \mathbf{fv}(t) \cup \{x\} & & \mathbf{fv}(C_x^{y,z}(t)) & := & (\mathbf{fv}(t) \setminus \{y,z\}) \cup \{x\}
\end{array}
$$

We only consider *well-formed* terms:

- Linearity
- Compulsory presence
- Barendregt's convention

**Notation** Given $\Phi \subseteq \mathbf{fv}(t)$, $R_{\Delta}^{\Phi}(t)$ denotes the renaming of $\Phi$ by $\Delta$. Example: $R_{y_1 y_2}^{x_1 x_2}(x_1 x_2 x_3) = y_1 y_2 x_3$.

# Typing Rules for the $\lambda\mathtt{lxr}$-calculus

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

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

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

where $\Gamma, \Delta$ is only defined if $\Gamma$ and $\Delta$ do not share variables.

**Notation** We write $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$ if $\Gamma \vdash t : A$ is derivable in this system.

**Remark** If $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$, then $\mathbf{fv}(t) = \mathbf{dom}(\Gamma)$.

# 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:

$$t[x/u][y/v] \equiv t[y/v][x/u] \text{ if } y \notin \mathbf{fv}(u) \mathbin{\&} x \notin \mathbf{fv}(v)$$

Contraction and substitution have the same status:

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

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

$$(B) \qquad (\lambda x.t)\ u \rightarrow \ t[x/u]$$

# SubSystem $\mathtt{x}$

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

# SubSystem t

| | | | | |
|---|---|---|---|---|
| $(WAbs)$ | $\lambda x.W_y(t)$ | $\rightarrow$ | $W_y(\lambda x.t)$ | $x \neq y$ |
| $(WApp1)$ | $W_y(u)\ v$ | $\rightarrow$ | $W_y(u\ v)$ | |
| $(WApp2)$ | $u\ W_y(v)$ | $\rightarrow$ | $W_y(u\ v)$ | |
| $(WSubs)$ | $t[x/W_y(u)]$ | $\rightarrow$ | $W_y(t[x/u])$ | |
| $(Merge)$ | $C_w^{y,z}(W_y(t))$ | $\rightarrow$ | $R_w^z(t)$ | |
| $(Cross)$ | $C_w^{y,z}(W_x(t))$ | $\rightarrow$ | $W_x(C_w^{y,z}(t))$ | $x \neq y, \quad x \neq z$ |
| $(CAbs)$ | $C_w^{y,z}(\lambda x.t)$ | $\rightarrow$ | $\lambda x.C_w^{y,z}(t)$ | |
| $(CApp1)$ | $C_w^{y,z}(t\ u)$ | $\rightarrow$ | $C_w^{y,z}(t)\ u$ | $y, z \in \mathbf{fv}(t)$ |
| $(CApp2)$ | $C_w^{y,z}(t\ u)$ | $\rightarrow$ | $t\ C_w^{y,z}(u)$ | $y, z \in \mathbf{fv}(u)$ |
| $(CSubs)$ | $C_w^{y,z}(t[x/u])$ | $\rightarrow$ | $t[x/C_w^{y,z}(u)]$ | $y, z \in \mathbf{fv}(u)$ |

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

The reduction relation is generated by the previous rewriting rules and congruence axioms (which are closed by all contexts):

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

# Example

$$
\begin{array}{ll}
(\lambda x.W_u(C_x^{y,z}(y\ z)))\ w & \to \\
W_u(C_x^{y,z}(y\ z))[x/w] & \to \\
W_u(C_x^{y,z}(y\ z)[x/w]) & \to \\
W_u(C_w^{w_1,w_2}((y\ z)[y/w_1][z/w_2])) & \to \\
W_u(C_w^{w_1,w_2}((y[y/w_1]\ z)[z/w_2])) & \to \\
W_u(C_w^{w_1,w_2}(y[y/w_1]\ z[z/w_2])) & \to \\
W_u(C_w^{w_1,w_2}(w_1\ z[z/w_2])) & \to \\
W_u(C_w^{w_1,w_2}(w_1\ w_2)) &
\end{array}
$$

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

1. (Full composition) $t[x/v] \to^* \; t\{x = v\}$
   for an *appropiate* notion of meta-substitution and even when $t$ contains non-evaluated substitutions
2. (Free variables are preserved) If $t \to_{\lambda\mathtt{lxr}} \; t'$, then $\mathbf{fv}(t) = \mathbf{fv}(t')$
3. (Subject reduction) If $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$ et $t \to_{\lambda\mathtt{lxr}} \; t'$, then $\Gamma \vdash_{\lambda\mathtt{lxr}} t' : A$.
4. (Convergence) $\mathtt{xt} = \mathtt{x} \cup \mathtt{t}$ is convergent (terminating and confluent).
   Which is the form of a term in $\mathtt{xt}$-normal form?

# Connexion with $\lambda$-calculus

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

$$\lambda\texttt{lxr} \quad \begin{array}{c} \xrightarrow{\mathcal{B}()} \\ \xleftarrow{\mathcal{A}()} \end{array} \quad \lambda$$

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

# From $\lambda$lxr to $\lambda$

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

# Relating $\lambda\texttt{lxr}$ and $\lambda$

### Lemma

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* $M \to_{\texttt{xt}} N$*, then* $\mathcal{B}(M) = \mathcal{B}(N)$*.*

**Proposition [Projecting $\lambda\texttt{lxr}$-reductions]** $M \to_{\lambda\texttt{lxr}} N$, then $\mathcal{B}(M) \to^*_\beta \mathcal{B}(N)$.

# From $\lambda$ to $\lambda$lxr

$$
\begin{array}{lcl}
\mathcal{A}(x) & := & x \\
\mathcal{A}(\lambda x.t) & := & \begin{cases} \lambda x.\mathcal{A}(t) & \text{if } x \in \mathbf{fv}(t) \\ \lambda x.W_x(\mathcal{A}(t)) & \text{if } x \notin \mathbf{fv}(t) \end{cases} \\
\mathcal{A}(tu) & := & \begin{cases} C_{\Phi}^{\Delta,\Pi}(R_{\Delta}^{\Phi}(\mathcal{A}(t))\ R_{\Pi}^{\Phi}(\mathcal{A}(u))) & \text{if } \Phi := \mathbf{fv}(t) \cap \mathbf{fv}(u) \neq \emptyset \ (\Delta, \Pi \text{ are fresh}) \\ \mathcal{A}(t)\mathcal{A}(u) & \text{if } \mathbf{fv}(t) \cap \mathbf{fv}(u) = \emptyset \end{cases}
\end{array}
$$

**Example** $\mathcal{A}(\lambda x.y\ y) = \lambda x.W_x(C_y^{z,z'}(z\ z'))$

### Lemma

*For all $\lambda$-terms $t$ and $u$ such that $x \in \mathbf{fv}(t)$, we have*

$$C^{\Delta,\Pi}_{\Phi}(R^{\Phi}_{\Delta}(\mathcal{A}(t))[x/R^{\Phi}_{\Pi}(\mathcal{A}(u))]) \rightarrow^{*}_{\mathtt{xt}} \mathcal{A}(t\{x\backslash u\})$$

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

**Proposition [Simulating $\beta$-reductions]**

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

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

# Connexion with $\lambda$-calculus (Summary)

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

$$t \to^*_{\mathtt{xt}} W_\Pi(\mathcal{A}(\mathcal{B}(t))) \qquad \lambda\mathtt{lxr} \quad \underset{\overset{\longleftarrow}{\mathcal{A}()}}{\overset{\mathcal{B}()}{\longrightarrow}} \quad \lambda \qquad t = \mathcal{B}(\mathcal{A}(t))$$

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

**Example** $t = (\lambda x.W_x(y))W_z(z') \to^*_{\mathtt{xt}} W_z((\lambda x.W_x(y))z') = W_z(\mathcal{A}(\mathcal{B}(t)))$.

### Lemma

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

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

### Theorem (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)
   1. If $\Gamma \vdash_{\mathtt{a}} t : A$ then $\Gamma \vdash_{\lambda\mathtt{lxr}} W_{\mathbf{dom}(\Gamma)\setminus\mathbf{fv}(t)}(\mathcal{A}(t)) : A$
   2. If $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$ then $\Gamma \vdash_{\mathtt{a}} \mathcal{B}(t) : A$

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

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

**(Call-by-Name) Translation of Formulae**

$$
\begin{array}{lcl}
\iota^+ & := & \iota \\
(A \to B)^+ & := & ?(A^-) \parr B^+ \\
A^- & := & (A^+)^\perp
\end{array}
$$

**Translation of Derivations**

Let $\Gamma = x_1 : B_1, \ldots, x_n : B_n$. Then $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$ translates to a MELL Proof-Net written $(\Gamma \vdash t : A)^\circ$ with interface $?\Gamma^-, A^+$, where $?\Gamma^-$ means $?B_1^-, \ldots, ?B_n^-$

# Translating (ax)

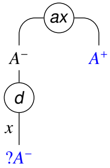
**Original derivation**:

$$\dfrac{}{x : A \vdash x : A}\ (\text{ax})$$

**Sequent Translation:**

$$\dfrac{\dfrac{}{\vdash A^-, A^+}}{\vdash ?A^-, A^+}$$

**Proof-Net Translation:**



# Translating (→ e)

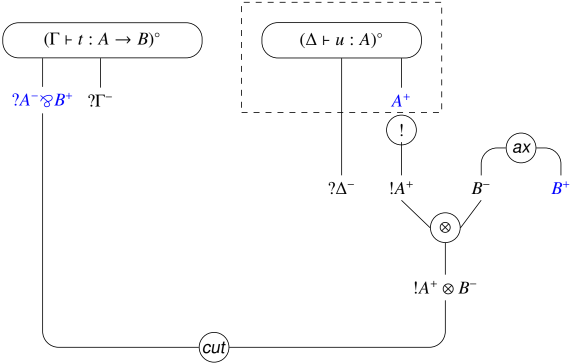
**Original derivation**:

$$\dfrac{\Gamma \vdash t : A \to B \qquad \Delta \vdash u : A}{\Gamma, \Delta \vdash tu : B}\ (\to\ \mathrm{e})$$

**Sequent Translation:**

$$\dfrac{\dfrac{i.h.}{\vdash ?\Gamma^-, ?A^- \parr B^+} \qquad \dfrac{\dfrac{\dfrac{i.h.}{\vdash ?\Delta^-, A^+}}{\vdash ?\Delta^-, !A^+} \qquad \dfrac{}{\vdash B^-, B^+}}{\vdash ?\Delta^-, !A^+ \otimes B^-, A^+}}{\vdash ?\Gamma^-, ?\Delta^-, B^+}$$

# Proof-Net Translation:



# Translating ($\rightarrow$ i)

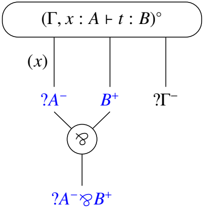
**Original derivation**:

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \rightarrow B}\ (\rightarrow \text{i})$$

**Sequent Translation:**

$$\dfrac{\dfrac{i.h.}{\vdash ?\Gamma^-, ?A^-, B^+}}{\vdash ?\Gamma^-, ?A^- \invamp B^+}$$

**Proof-Net Translation:**



# Translating (cut)

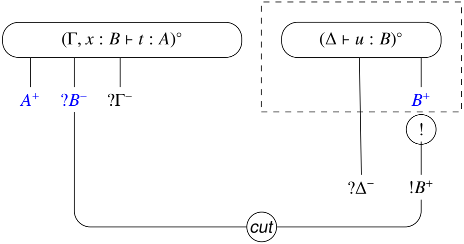
**Original derivation**:

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

**Sequent Translation:**

$$\dfrac{\dfrac{\dfrac{i.h.}{\vdash ?\Delta^-, B^+}}{\vdash ?\Delta^-, !B^+} \qquad \dfrac{i.h.}{\vdash ?\Gamma^-, ?B^-, A^+}}{\vdash ?\Delta^-, ?\Gamma^-, A^+}$$

**Proof-Net Translation:**



# Translating (c)

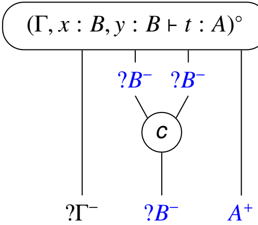
**Original derivation**:

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

**Sequent Translation:**

$$\frac{\vdash ?\Gamma^-, ?B^-, ?B^-, A^+}{\vdash ?\Gamma^-, ?B^-, A^+}$$

**Proof-Net Translation:**



# Translating (w)

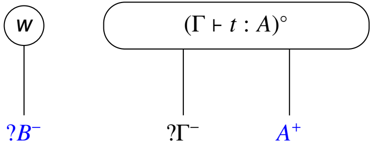
**Original derivation**:

$$\frac{\Gamma \vdash t : A}{\Gamma, x : B \vdash W_x(t) : A}\ (\mathrm{w})$$

**Sequent Translation:**

$$\frac{\vdash ?\Gamma^-, A^+}{\vdash ?\Gamma^-, ?B^-, A^+}$$

**Proof-Net Translation:**



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

## Theorem (Soundness)

*$\lambda\mathtt{lxr}$ is sound w.r.t proof-nets:*
*If $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$, then $t \rightarrow_{\lambda\mathtt{lxr}} u$ implies $(\Gamma \vdash_{\lambda\mathtt{lxr}} t : A)^{\circ} \rightarrow^{*}_{\mathcal{R}/\mathcal{E}} (\Gamma \vdash_{\lambda\mathtt{lxr}} u : A)^{\circ}$.*

**The proof uses the following property:**

## Lemma

*Let $t, u$ be a $\lambda\mathtt{lxr}$-typed terms s.t. $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$ and $\Gamma \vdash_{\lambda\mathtt{lxr}} u : A$.*

- *If $t \equiv u$, then $(\Gamma \vdash t : A)^{\circ} \simeq_{\mathcal{E}} (\Gamma \vdash u : A)^{\circ}$.*
- *If $t \rightarrow_{B} u$, then $(\Gamma \vdash t : A)^{\circ} \rightarrow^{+}{}_{\mathcal{R}/\mathcal{E}} (\Gamma \vdash u : A)^{\circ}$.*
- *If $t \rightarrow_{\mathtt{xt}} u$, then $(\Gamma \vdash t : A)^{\circ} \rightarrow^{*}_{\mathcal{R}/\mathcal{E}} (\Gamma \vdash u : A)^{\circ}$.*

# Corollary of the Translation

### Theorem (Strong Normalisation)

*The relation $\rightarrow_{\lambda\mathtt{lxr}}$ is strongly normalising on well-typed $\lambda\mathtt{lxr}$-terms:*
*if $\Gamma \vdash_{\lambda\mathtt{lxr}} t : A$, then $t \in SN(\lambda\mathtt{lxr})$.*

### Proof.

Using the previous lemma, the termination property of the relation $\mathtt{xt}$ and SN of $\rightarrow_{\mathcal{R}/\mathcal{E}}$. □

# Towards completeness

- Define a congruence $\approx$ for proof-nets.
- Define a congruence $\cong$ for $\lambda\mathtt{lxr}$-terms.
- Show that $(\Gamma \vdash t_1 : A)^\circ \approx (\Gamma' \vdash t_2 : A')^\circ$ implies $t_1 \cong t_2$.

# Summary

The $\lambda$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 $\beta$-reduction.
- Preservation of $\beta$-strong normalization.
- Strong normalization of well-typed terms.
- Full and safe composition.
- Sound and complete with respect to proof-nets.
- Explicit operators for implementation issues.