From Explicit Substitution Calculi to MELL Proof-Nets

Agenda for Today

1. Typing Systems for Explicit Substitutions
2. Strong Normalisation for Typed Terms with ES
3. From Typed Terms to MELL - Static Translation
4. From Typed Terms to MELL - Dynamic Translation

Agenda

Type Systems for Explicit Substitutions - Definitions

- Capital greek letters $\Gamma, \Delta, \Pi, \ldots$ are used to denote finite **contexts**, defined as partial functions from variables to types.
- Alternative notation for contexts: $x_1 : A_1, \ldots, x_n : A_n$.
- **Domain of contexts**: $\mathbf{dom}(\Gamma) = \{x \mid \Gamma(x) \text{ defined}\}$.
- **Union of contexts**:

$$(\Gamma \cup \Delta)(x) := \begin{cases} \Gamma(x) & \text{if } x \in \mathbf{dom}(\Gamma) \;\&\; x \notin \mathbf{dom}(\Delta) \\ \Delta(x) & \text{if } x \notin \mathbf{dom}(\Gamma) \;\&\; x \in \mathbf{dom}(\Delta) \\ \Delta(x) & \text{if } x \in \mathbf{dom}(\Gamma) \;\&\; x \in \mathbf{dom}(\Delta) \;\&\; \text{ and } \Gamma(x) = \Delta(x) \\ \text{undefined} & \text{otherwise} \end{cases}$$

- **Union of disjoint contexts**: $\Gamma, \Delta := \Gamma \cup \Delta$ when $\mathbf{dom}(\Gamma) \cap \mathbf{dom}(\Delta) = \emptyset$.

Simple Types for Explicit Substitutions - Addivite System

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i} \text{(axiom)}$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B} (\to \text{ intro}) \qquad \frac{\Gamma \vdash t : A \to B \quad \Gamma \vdash u : A}{\Gamma \vdash tu : B} (\to \text{ elim})$$

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

Notation We write $\Gamma \vdash_a t : A$ if $\Gamma \vdash t : A$ is derivable in this system.

Remark If $\Gamma \vdash_a t : A$ is derivable, then $\mathbf{fv}(t) \subseteq \mathbf{dom}(\Gamma)$.

Simple Types for Explicit Substitutions - Multiplicative System

$$\frac{}{x : A \vdash x : A}\ (\text{ax}) \qquad \frac{\Gamma \vdash t : A \to B \qquad \Delta \vdash u : A}{\Gamma \cup \Delta \vdash tu : B}\ (\to\ \text{e})$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\ (\to\ \text{i1}) \qquad \frac{\Gamma \vdash t : B \qquad x \notin \mathbf{fv}(t)}{\Gamma \vdash \lambda x.t : A \to B}\ (\to\ \text{i2})$$

$$\frac{\Gamma \vdash u : B \qquad \Delta, x : B \vdash t : A}{\Gamma \cup \Delta \vdash t[x\backslash u] : A}\ (\text{cut1})$$

$$\frac{\Gamma \vdash u : B \qquad \Delta \vdash t : A \qquad x \notin \mathbf{fv}(t)}{\Gamma \cup \Delta \vdash t[x\backslash u] : A}\ (\text{cut2})$$

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

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

Relating the Additive and the Multiplicative System

Lemma

If $\Gamma \vdash_{\mathtt{a}} t : A$, *then* $\Gamma|_{\mathbf{fv}(t)} \vdash_{\mathtt{m}} t : A$.

Lemma

If $\Gamma \vdash_{\mathtt{m}} t : A$, *then* $\Gamma \vdash_{\mathtt{a}} t : A$.

Exercise:
Give a typing derivation for the term $\lambda w.(xy)[x\backslash a][z\backslash b]$ in the additive as well as in the multiplicative system.

Reduction Preserves Types

Theorem (Subject Reduction)

- *If* $\Gamma \vdash_{\mathtt{a}} t : A$ *and* $t \to_{\lambda\mathtt{pn}} t'$*, then* $\Gamma \vdash_{\mathtt{a}} t' : A$*.*
- *If* $\Gamma \vdash_{\mathtt{m}} t : A$ *and* $t \to_{\lambda\mathtt{pn}} t'$*, then* $\Gamma' \vdash_{\mathtt{m}} t' : A$ *for some* $\Gamma' \subseteq \Gamma$*.*

Proof.

The proof of the first point is by induction on $t \to_{\lambda\mathtt{pn}} t'$. The proof of the second point uses the following property:

Lemma: Let $\Gamma \vdash_{\mathtt{m}} t : A$.

- If $t \equiv u$, then $\Gamma \vdash_{\mathtt{m}} u : A$.
- If $t \to_{\mathtt{dB,var,arg,dup}} u$, then $\Gamma \vdash_{\mathtt{m}} u : A$.
- If $t \to_{\mathtt{gc}} u$, then $\Gamma' \vdash_{\mathtt{m}} u : A$ for some $\Gamma' \subseteq \Gamma$

. □

Agenda

1. Typing Systems for Explicit Substitutions
2. Strong Normalisation for Typed Terms with ES
3. From Typed Terms to MELL - Static Translation
4. From Typed Terms to MELL - Dynamic Translation

Strong Normalisation

Lemma

- *If* $\Gamma, x : B \vdash_{\mathtt{a}} t : A$, $\Gamma \vdash_{\mathtt{a}} u : B$, *and* $t, u \in SN(\lambda\mathrm{pn})$, *then* $t\{x\backslash u\} \in SN(\lambda\mathrm{pn})$.
- *If* $\Gamma, x : B \vdash_{\mathtt{m}} t : A$, $\Delta \vdash_{\mathtt{m}} u : B$, *and* $t, u \in SN(\lambda\mathrm{pn})$, *then* $t\{x\backslash u\} \in SN(\lambda\mathrm{pn})$.

Proof.

Induction on the lexicographic 3-tuple $\langle B, \eta_{\lambda\mathrm{pn}}(t), t\rangle$, where $\eta_{\lambda\mathrm{pn}}(t)$ denotes the maximal length of an $\lambda\mathrm{pn}$-reduction sequence starting at t. The proof also uses subject reduction. □

Theorem (SN for λpn)

- *If* $\Gamma \vdash_{\mathtt{a}} t : A$*, then* $t \in SN(\lambda\mathrm{pn})$.
- *If* $\Gamma \vdash_{\mathtt{m}} t : A$*, then* $t \in SN(\lambda\mathrm{pn})$.

Proof 1: Induction on typing derivability and previous lemma using `ISN`.
Proof 2: Using Intersection Types.
Proof 3:

- Define a translation `T` from λpn to λ by unfolding substitutions.
- Show that t typable implies $\mathtt{T}(t)$ typable in λ-calculus.
- Apply SN Theorem of λ-calculus to get $\mathtt{T}(t) \in SN(\beta)$.
- Conclude $\mathtt{T}(t) \in SN(\lambda\mathrm{pn})$ by PSN.
- Since $\mathtt{T}(t) \rightarrow^*_{\lambda\mathrm{pn}} t$, then $t \in SN(\lambda\mathrm{pn})$.

Proof 4: In a few more slides

Agenda

(Call-by-Name) Translation of Formulae

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

Observation:

- $(?(A^-) \parr B^+)^\perp = !A^+ \otimes B^-$.
- $(?A^-)^\perp = !A^+$.

Translation of Derivations

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

Alternative Multiplicative Presentation of Simple Types for Explicit Substitutions

$$\frac{}{x : A \vdash x : A}\ (\text{ax}) \qquad \frac{\Gamma_t, \Gamma_{tu} \vdash t : A \to B \quad \Gamma_u, \Gamma_{tu} \vdash u : A \quad \Gamma_t \# \Gamma_u}{\Gamma_t, \Gamma_u, \Gamma_{tu} \vdash tu : B}\ (\to\ \text{e})$$

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}\ (\to\ \text{i1}) \qquad \frac{\Gamma \vdash t : B \quad x \notin \mathbf{fv}(t)}{\Gamma \vdash \lambda x.t : A \to B}\ (\to\ \text{i2})$$

$$\frac{\Gamma_u, \Gamma_{tu} \vdash u : B \quad \Gamma_t, \Gamma_{tu}, x : B \vdash t : A \quad \Gamma_t \# \Gamma_u}{\Gamma_t, \Gamma_u, \Gamma_{tu} \vdash t[x \backslash u] : A}\ (\text{cut1})$$

$$\frac{\Gamma_u, \Gamma_{tu} \vdash u : B \quad \Gamma_t, \Gamma_{tu} \vdash t : A \quad x \notin \mathbf{fv}(t) \quad \Gamma_t \# \Gamma_u}{\Gamma_t, \Gamma_u, \Gamma_{tu} \vdash t[x \backslash u] : A}\ (\text{cut2})$$

Translating (ax)

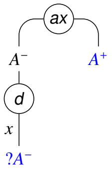
Original derivation:

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

Sequent Translation:

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

Proof-Net Translation:



Translating $(\to e)$

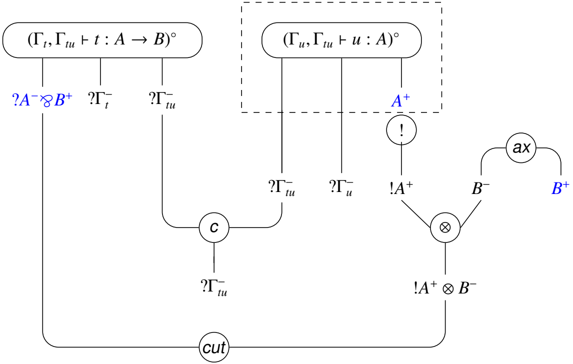
Original derivation:

$$\frac{\Gamma_t, \Gamma_{tu} \vdash t : A \to B \qquad \Gamma_u, \Gamma_{tu} \vdash u : A \qquad \Gamma_t \# \Gamma_u}{\Gamma_t, \Gamma_u, \Gamma_{tu} \vdash tu : B}\ (\to e)$$

Sequent Translation:

$$\frac{\dfrac{\dfrac{i.h.}{\vdash ?\Gamma_t^-, ?\Gamma_{tu}^-, ?A^- \parr B^+} \qquad \dfrac{\dfrac{\dfrac{i.h.}{\vdash ?\Gamma_u^-, ?\Gamma_{tu}^-, A^+}}{\vdash ?\Gamma_u^-, ?\Gamma_{tu}^-, !A^+} \qquad \dfrac{}{\vdash B^-, B^+}}{\vdash ?\Gamma_u^-, ?\Gamma_{tu}^-, !A^+ \otimes B^-, B^+}}{\vdash ?\Gamma_t^-, ?\Gamma_u^-, ?\Gamma_{tu}^-, ?\Gamma_{tu}^-, B^+}}{\vdash ?\Gamma_t^-, ?\Gamma_u^-, ?\Gamma_{tu}^-, B^+}$$

Proof-Net Translation:



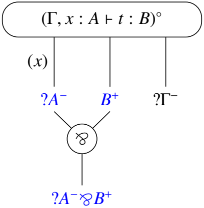
Original derivation:

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x.t : A \rightarrow B}\ (\rightarrow\ \text{i1})$$

Sequent Translation:

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

Proof-Net Translation:



Translating ($\rightarrow$ i2)

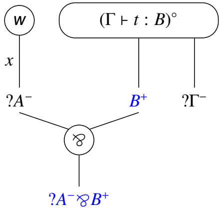
Original derivation:

$$\frac{\Gamma \vdash t : B \qquad x \notin \mathbf{fv}(t)}{\Gamma \vdash \lambda x.t : A \rightarrow B}\ (\rightarrow \text{ i2})$$

Sequent Translation:

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

Proof-Net Translation:



Translating (cut1)

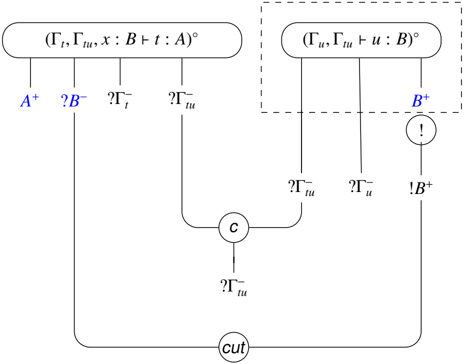
Original derivation:

$$\dfrac{\Gamma_u, \Gamma_{tu} \vdash u : B \qquad \Gamma_t, \Gamma_{tu}, x : B \vdash t : A \qquad \Gamma_t \# \Gamma_u}{\Gamma_u, \Gamma_t, \Gamma_{tu} \vdash t[x \backslash u] : A} \text{(cut1)}$$

Sequent Translation:

$$\dfrac{\dfrac{\dfrac{\dfrac{i.h.}{\vdash ?\Gamma_u^-, ?\Gamma_{tu}^-, B^+}}{\vdash ?\Gamma_u^-, ?\Gamma_{tu}^-, !B^+} \qquad \dfrac{i.h.}{\vdash ?\Gamma_t^-, ?\Gamma_{tu}^-, ?B^-, A^+}}{\vdash ?\Gamma_u^-, ?\Gamma_t^-, ?\Gamma_{tu}^-, ?\Gamma_{tu}^-, A^+}}{\vdash ?\Gamma_u^-, ?\Gamma_t^-, ?\Gamma_{tu}^-, A^+}$$

Proof-Net Translation:



Translating (cut2)

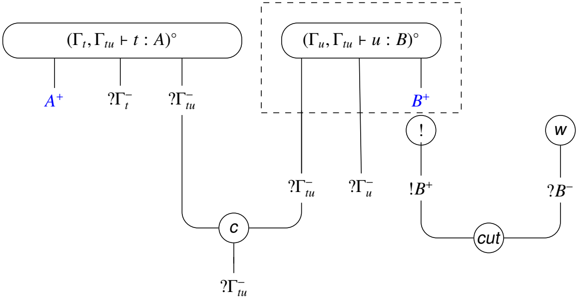
Original derivation:

$$\frac{\Gamma_u, \Gamma_{tu} \vdash u : B \qquad \Gamma_t, \Gamma_{tu} \vdash t : A \qquad \Gamma_t \# \Gamma_u \qquad x \notin \mathbf{fv}(t)}{\Gamma_u, \Gamma_t, \Gamma_{tu} \vdash t[x\backslash u] : A} \text{(cut2)}$$

Sequent Translation:

$$\dfrac{\dfrac{\dfrac{i.h.}{\vdash ?\Gamma_u^-, ?\Gamma_{tu}^-, B^+}}{\vdash ?\Gamma_u^-, ?\Gamma_{tu}^-, !B^+}(!) \qquad \dfrac{\dfrac{i.h.}{\vdash ?\Gamma_t^-, ?\Gamma_{tu}^-, A^+}}{\vdash ?\Gamma_t^-, ?\Gamma_{tu}^-, ?B^-, A^+}}{\dfrac{\vdash ?\Gamma_u^-, ?\Gamma_t^-, ?\Gamma_{tu}^-, ?\Gamma_{tu}^-, A^+}{\vdash ?\Gamma_u^-, ?\Gamma_t^-, ?\Gamma_{tu}^-, A^+}}$$

Proof-Net Translation:



Salient Features

- Arguments of applications are translated to boxes.
- Arguments of substitutions are translated to boxes.

Agenda

Translating $\lambda\mathrm{pn}$-Reduction to PN

Theorem

Let $\Gamma \vdash_{\mathtt{m}} t : A$ *and* $t \to_{\lambda\mathrm{pn}} u$*. Consider the derivation* $\Gamma' \vdash_{\mathtt{m}} u : A$ *(for some* $\Gamma' \subseteq \Gamma$*) giving by the subject reduction property. Then* $(\Gamma \vdash t : A)^{\circ} \to^{*}_{\mathcal{R}/\mathcal{E}} W[(\Gamma' \vdash_{\mathtt{m}} u : A)^{\circ}]$ *for some MELL-context* W *made only of weakenings.*

The proof uses the following property:

Lemma

Let $\Gamma \vdash_{\mathtt{m}} t : A$ *and* $t \to u$ *and* $\Gamma' \vdash_{\mathtt{m}} u : A$ *as in the statement of the theorem.*

- *If* $t \equiv u$*, then* $(\Gamma \vdash t : A)^{\circ} \simeq_{\mathcal{E}} (\Gamma \vdash u : A)^{\circ}$.
- *If* $t \to_{\mathtt{dB}} u$*, then* $(\Gamma \vdash t : A)^{\circ} \to^{+}{}_{\mathtt{C}(\parr,\otimes),\mathtt{C}(a)} (\Gamma \vdash u : A)^{\circ}$.
- *If* $t \to_{\mathtt{var}} u$*, then* $(\Gamma \vdash t : A)^{\circ} \to^{+}{}_{\mathtt{C}(d,b),\mathtt{C}(a)} (\Gamma \vdash u : A)^{\circ}$.
- *If* $t \to_{\mathtt{arg}} u$*, then* $(\Gamma \vdash t : A)^{\circ} \to^{+}{}_{\mathtt{C}(b,b)} (\Gamma \vdash u : A)^{\circ}$.
- *If* $t \to_{\mathtt{dup}} u$*, then* $(\Gamma \vdash t : A)^{\circ} \to^{+}{}_{\mathtt{C}(c,b)} (\Gamma \vdash u : A)^{\circ}$.
- *If* $t \to_{\mathtt{gc}} u$*, then* $(\Gamma \vdash t : A)^{\circ} \to^{+}{}_{\mathtt{C}(w,b),w\text{-}b,w\text{-}c} W[(\Gamma' \vdash u : A)^{\circ}]$.

Corollary of the Translation

Theorem (SN for $\lambda\mathtt{pn}$)

If $\Gamma \vdash_{\mathtt{m}} t : A$*, then* $t \in SN(\lambda\mathtt{pn})$*.*

Proof 4: using the previous theorem and SN of $\rightarrow_{\mathcal{R}/\mathcal{E}}$.

The notion of σ-equivalence in λ-calculus

$$
\begin{array}{llll}
(\lambda x.\lambda y.u)v & \equiv_{\sigma_1} & \lambda y.(\lambda x.u)v & \text{if } y \notin \mathbf{fv}(v) \\
(\lambda x.uv)t & \equiv_{\sigma_2} & ((\lambda x.u)t)v & \text{if } x \notin \mathbf{fv}(v)
\end{array}
$$

Lemma

If $t \equiv_\sigma t'$, then $t \equiv_\beta t'$.

Theorem (Regnier'90)

If $t \equiv_\sigma t'$, then $\eta_\beta(t) = \eta_\beta(t')$.

Theorem

If $t \equiv_\sigma t'$, then t and t' translate to the same MELL Proof-Net (modulo multiplicative cuts).

The notion of σ-equivalence for calculi with ES

Recall the relation $\equiv$ on $\lambda\mathrm{pn}$-terms:

$$
\begin{array}{lcll}
(\lambda y.u)[x\backslash v] & \equiv & \lambda y.u[x\backslash v] & \text{if } y \notin \mathbf{fv}(v) \\
(tu)[x\backslash v] & \equiv & t[x\backslash v]u & \text{if } x \notin \mathbf{fv}(u) \\
t[y\backslash u][x\backslash v] & \equiv & t[x\backslash v][y\backslash u] & \text{if } x \notin \mathbf{fv}(u), y \notin \mathbf{fv}(v)
\end{array}
$$

Theorem

If $t \equiv t'$, then $\eta_{\lambda\mathrm{pn}}(t) = \eta_{\lambda\mathrm{pn}}(t')$.

Theorem

If $t \equiv t'$, then t and t' translate to the same MELL Proof-Net (modulo multiplicative cuts).

Summary

The $\lambda\mathrm{pn}$-calculus is a computational interpretation of Natural Deduction plus cut enjoying the following properties:

- Full composition.
- Confluence on terms.
- Confluence on open terms.
- Simulation of one-step β-reduction.
- Preservation of β-strong normalization.
- Strong normalization of well-typed terms.
- Admits a (fine-grained) translation into MELL Proof-Nets.