Proof theory for programming-language problems

Gabriel Scherer

Parsifal, INRIA Saclay

February 12, 2020

Question

When are two program fragments t, u contextually equivalent?

$$\forall C, \qquad C[t] \approx C[u]$$

Specifics depend on the programming language: input/output, non-termination, just values?

Untyped λ-calculus: undecidable.
Simple type system $\Lambda C(\alpha, \rightarrow)$: decidable.
Polymorphism $\Lambda C(\alpha, \rightarrow, \forall)$, dependent types $\Lambda C(\alpha, \rightarrow, \Pi)$: undecidable.

What's in the middle? Simple types, but richer datatypes?

History

Decidability of equivalence:

- $\Lambda C(\alpha, \to)$: Tait, 1967 or earlier.
- $\Lambda C(\alpha, \to, \times)$: essentially the same proof.
- $\Lambda C(\alpha, \to, \times, 1)$: essentially the same proof.
- $\Lambda C(\alpha, \to, \times, 1, +)$: Ghani [1995]; Altenkirch, Dybjer, Hofmann, and Scott [2001]; Balat, Di Cosmo, and Fiore [2004].
- $\Lambda C(\alpha, \to, \times, 1, +, 0)$: 2017. The topic of this course (if time permits).

Why are $(+, 0)$ so hard?

Simply-typed lambda-calculus

$$
\begin{array}{llll}
A, B, C & ::= & & \\
 & | & \alpha, \beta, \gamma & \text{variable/atomic type} \\
 & | & A \to B & \text{function type} \\
 & | & A \times B & \text{pair type} \\
 & | & A + B & \text{sum type} \\
 & | & 1 & \text{unit type} \\
 & | & 0 & \text{empty type}
\end{array}
$$

Simply-typed lambda-calculus

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

$$\frac{(\Gamma \vdash t_i : A_i)^{i \in \{1,2\}}}{\Gamma \vdash (t_1, t_2) : A_1 \times A_2} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \pi_i\ t : A_i}$$

Simply-typed lambda-calculus

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

$$\frac{(\Gamma \vdash t_i : A_i)^{i \in \{1,2\}}}{\Gamma \vdash (t_1, t_2) : A_1 \times A_2} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \pi_i\ t : A_i}$$

$$\frac{\Gamma \vdash t : A_i}{\Gamma \vdash \sigma_i\ t : A_1 + A_2} \qquad \frac{\Gamma \vdash t : A_1 + A_2 \qquad (\Gamma, x_i : A_i \vdash u_i : C)^{i \in \{1,2\}}}{\Gamma \vdash \texttt{match}\ t\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x_1 \to u_1 \\ \sigma_2\ x_2 \to u_2 \end{array} \right. : C}$$

Simply-typed lambda-calculus

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

$$\frac{(\Gamma \vdash t_i : A_i)^{i \in \{1,2\}}}{\Gamma \vdash (t_1, t_2) : A_1 \times A_2} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \pi_i\ t : A_i}$$

$$\frac{\Gamma \vdash t : A_i}{\Gamma \vdash \sigma_i\ t : A_1 + A_2} \qquad \frac{\Gamma \vdash t : A_1 + A_2 \qquad (\Gamma, x_i : A_i \vdash u_i : C)^{i \in \{1,2\}}}{\Gamma \vdash \texttt{match}\ t\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x_1 \to u_1 \\ \sigma_2\ x_2 \to u_2 \end{array} \right. : C}$$

$$\frac{}{\Gamma \vdash () : 1}$$

Simply-typed lambda-calculus

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

$$\frac{(\Gamma \vdash t_i : A_i)^{i \in \{1,2\}}}{\Gamma \vdash (t_1, t_2) : A_1 \times A_2} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \pi_i\ t : A_i}$$

$$\frac{\Gamma \vdash t : A_i}{\Gamma \vdash \sigma_i\ t : A_1 + A_2} \qquad \frac{\Gamma \vdash t : A_1 + A_2 \qquad (\Gamma, x_i : A_i \vdash u_i : C)^{i \in \{1,2\}}}{\Gamma \vdash \texttt{match}\ t\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x_1 \to u_1 \\ \sigma_2\ x_2 \to u_2 \end{array} \right. : C}$$

$$\frac{}{\Gamma \vdash () : 1} \qquad \frac{\Gamma \vdash t : 0}{\Gamma \vdash \texttt{absurd}(t) : A}$$

Simply-typed $\beta\eta$-equivalence?

$$(\lambda x.\, t)\ u \ \rhd_\beta \ t[u/x] \qquad\qquad \pi_i\ (t_1, t_2) \ \rhd_\beta \ t_i$$

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \left|\ \begin{array}{l} \sigma_1\ x_1 \to u_1 \\ \sigma_2\ x_2 \to u_2 \end{array}\right. \ \rhd_\beta \ u_i[t/x_i]$$

$$\frac{\Gamma \vdash t : A \to B}{t \rhd_\eta \lambda x.\,(t\ x)} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{t \rhd_\eta (\pi_1\ t, \pi_2\ t)} \qquad \frac{\Gamma \vdash t : 1}{t \rhd_\eta ()}$$

Simply-typed $\beta\eta$-equivalence?

$$(\lambda x.\, t)\ u \ \triangleright_\beta\ t[u/x] \qquad\qquad \pi_i\ (t_1, t_2)\ \triangleright_\beta\ t_i$$

$$\texttt{match}\ \sigma_i\ t\ \texttt{with}\ \left|\ \begin{array}{l} \sigma_1\ x_1 \to u_1 \\ \sigma_2\ x_2 \to u_2 \end{array}\right. \ \triangleright_\beta\ u_i[t/x_i]$$

$$\frac{\Gamma \vdash t : A \to B}{t \triangleright_\eta \lambda x.\,(t\ x)} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{t \triangleright_\eta (\pi_1\ t, \pi_2\ t)} \qquad \frac{\Gamma \vdash t : 1}{t \triangleright_\eta ()} \qquad \frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta\ ?}$$

$$\frac{\Gamma \vdash t : A_1 \times A_2}{t \rhd_\eta (\pi_1\ t, \pi_2\ t)}$$

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \rhd_\eta \texttt{match}\ t\ \texttt{with} \left|\begin{array}{l} \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \sigma_2\ x_2 \to \sigma_2\ x_2 \end{array}\right.}$$

?

$$\frac{\Gamma \vdash t : A_1 \times A_2}{t \rhd_\eta (\pi_1\ t, \pi_2\ t)} \qquad \frac{\Gamma \vdash t : A_1 + A_2}{t \rhd_\eta \texttt{match}\ t\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \sigma_2\ x_2 \to \sigma_2\ x_2 \end{array} \right.}$$

?

But:

$$(t, t') \approx_? \texttt{match}\ t\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x_1 \to (\sigma_1\ x_1, t') \\ \sigma_2\ x_2 \to (\sigma_2\ x_2, t') \end{array} \right.$$

$$\frac{\Gamma \vdash t : A_1 \times A_2}{t \rhd_\eta (\pi_1\ t, \pi_2\ t)}$$

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \rhd_\eta \texttt{match}\ t\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \sigma_2\ x_2 \to \sigma_2\ x_2 \end{array} \right.}$$

?

But:

$$(t, t') \approx_? \texttt{match}\ t\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x_1 \to (\sigma_1\ x_1, t') \\ \sigma_2\ x_2 \to (\sigma_2\ x_2, t') \end{array} \right.$$

General rule:

$$\frac{\Gamma \vdash t : A_1 + A_2 \qquad \Gamma, y : A_1 + A_2 \vdash u : C}{u[t/y] \rhd_\eta \texttt{match}\ t\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x_1 \to u[\sigma_1\ x_1/y] \\ \sigma_2\ x_2 \to u[\sigma_2\ x_2/y] \end{array} \right.}$$

(In the example, $u \overset{\text{def}}{=} (y, t')$)

Simply-typed $\beta\eta$-equivalence; full

$$(\lambda x.\, t)\; u \;\triangleright_\beta\; t[u/x] \qquad\qquad \pi_i\;(t_1, t_2) \;\triangleright_\beta\; t_i$$

$$\texttt{match}\; \sigma_i\; t\; \texttt{with} \;\left|\; \begin{array}{l} \sigma_1\; x_1 \to u_1 \\ \sigma_2\; x_2 \to u_2 \end{array}\right. \;\triangleright_\beta\; u_i[t/x_i]$$

$$\frac{\Gamma \vdash t : A \to B}{t \triangleright_\eta \lambda x.\,(t\; x)} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{t \triangleright_\eta (\pi_1\; t, \pi_2\; t)} \qquad \frac{\Gamma \vdash t : 1}{t \triangleright_\eta ()}$$

Simply-typed $\beta\eta$-equivalence; full

$$(\lambda x.\, t)\; u \;\triangleright_\beta\; t[u/x] \qquad\qquad \pi_i\; (t_1, t_2) \;\triangleright_\beta\; t_i$$

$$\texttt{match}\; \sigma_i\; t\; \texttt{with} \;\left|\; \begin{array}{l} \sigma_1\; x_1 \to u_1 \\ \sigma_2\; x_2 \to u_2 \end{array}\right. \;\triangleright_\beta\; u_i[t/x_i]$$

$$\frac{\Gamma \vdash t : A \to B}{t \triangleright_\eta \lambda x.\, (t\; x)} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{t \triangleright_\eta (\pi_1\; t, \pi_2\; t)} \qquad \frac{\Gamma \vdash t : 1}{t \triangleright_\eta ()}$$

$$\frac{\Gamma \vdash t : A_1 + A_2 \qquad \Gamma, y : A_1 + A_2 \vdash u : C}{u[t/y] \triangleright_\eta \begin{array}{l} \texttt{match}\; t\; \texttt{with} \\ \left|\; \begin{array}{l} \sigma_1\; x_1 \to u[\sigma_1\; x_1/y] \\ \sigma_2\; x_2 \to u[\sigma_2\; x_2/y] \end{array}\right. \end{array}}$$

Simply-typed $\beta\eta$-equivalence; full

$$(\lambda x.\, t)\ u \ \triangleright_\beta\ t[u/x] \qquad\qquad \pi_i\ (t_1, t_2)\ \triangleright_\beta\ t_i$$

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \left|\begin{array}{l} \sigma_1\ x_1 \to u_1 \\ \sigma_2\ x_2 \to u_2 \end{array}\right. \ \triangleright_\beta\ u_i[t/x_i]$$

$$\frac{\Gamma \vdash t : A \to B}{t \triangleright_\eta \lambda x.\,(t\ x)} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{t \triangleright_\eta (\pi_1\ t, \pi_2\ t)} \qquad \frac{\Gamma \vdash t : 1}{t \triangleright_\eta ()}$$

$$\frac{\Gamma \vdash t : A_1 + A_2 \qquad \Gamma, y : A_1 + A_2 \vdash u : C}{u[t/y] \triangleright_\eta \begin{array}{l}\texttt{match}\ t\ \texttt{with} \\ \left|\begin{array}{l} \sigma_1\ x_1 \to u[\sigma_1\ x_1/y] \\ \sigma_2\ x_2 \to u[\sigma_2\ x_2/y] \end{array}\right.\end{array}} \qquad \frac{\Gamma \vdash t : 0 \qquad \Gamma, y : 0 \vdash u : C}{u[t/y] \triangleright_\eta \texttt{absurd}(t)}$$

Simply-typed $\beta\eta$-equivalence; full

$$(\lambda x.\, t)\ u \ \rhd_\beta\ t[u/x] \qquad\qquad \pi_i\ (t_1, t_2)\ \rhd_\beta\ t_i$$

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \left|\begin{array}{l} \sigma_1\ x_1 \to u_1 \\ \sigma_2\ x_2 \to u_2 \end{array}\right. \ \rhd_\beta\ u_i[t/x_i]$$

$$\frac{\Gamma \vdash t : A \to B}{t \rhd_\eta \lambda x.\,(t\ x)} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{t \rhd_\eta (\pi_1\ t, \pi_2\ t)} \qquad \frac{\Gamma \vdash t : 1}{t \rhd_\eta ()}$$

$$\frac{\Gamma \vdash t : A_1 + A_2 \qquad \Gamma, y : A_1 + A_2 \vdash u : C}{u[t/y] \rhd_\eta \begin{array}{l}\texttt{match}\ t\ \texttt{with} \\ \left|\begin{array}{l} \sigma_1\ x_1 \to u[\sigma_1\ x_1/y] \\ \sigma_2\ x_2 \to u[\sigma_2\ x_2/y] \end{array}\right.\end{array}} \qquad \frac{\Gamma \vdash t : 0 \qquad \Gamma, y : 0 \vdash u : C}{u[t/y] \rhd_\eta \texttt{absurd}(t)}$$

Derived rules :

$$\frac{}{\Gamma \vdash t_1 \approx_\eta t_2 : 1} \qquad \frac{\Gamma \vdash t : 0 \qquad \Gamma \vdash u_1, u_2 : A}{\Gamma \vdash u_1 \approx_\eta u_2 : A}$$

β-normal forms (negative)

β-short normal forms:

$$\pi_1\ (t, u) = t$$

$$\begin{array}{lcl} v, w & ::= & \lambda x.\, v \mid (v, w) \mid n \\ n, m & ::= & \pi_i\ n \mid n\ v \mid x \end{array}$$

β-normal forms (negative)

β-short normal forms:

$$\pi_1\ (t, u) = t$$

$$\begin{array}{rcl} v, w & ::= & \lambda x.\, v \mid (v, w) \mid n \\ n, m & ::= & \pi_i\ n \mid n\ v \mid x \end{array}$$

β-short η-long:

$$(y : \alpha \to \beta) = \lambda x : \alpha.\, (y\ x : \beta)$$

β-normal forms (negative)

β-short normal forms:

$$\pi_1\ (t, u) = t$$

$$\begin{aligned} v, w &::= \lambda x.\, v \mid (v, w) \mid n \\ n, m &::= \pi_i\ n \mid n\ v \mid x \end{aligned}$$

β-short η-long:

$$(y : \alpha \to \beta) = \lambda x : \alpha.\, (y\ x : \beta)$$

$$\begin{aligned} v, w &::= \lambda x.\, v \mid (v, w) \mid (n : \alpha) \\ n, m &::= \pi_i\ n \mid n\ v \mid x \end{aligned}$$

What about sums?

$$
\begin{aligned}
v, w &::= \lambda x.\, v \mid (v, w) \mid \sigma_i\ v \mid (n : \alpha) \\
n, m &::= \pi_i\ n \mid n\ v \mid \left(\texttt{match}\ n\ \texttt{with} \left| \begin{array}{l} \sigma_1\ y_1 \to v_1 \\ \sigma_2\ y_2 \to v_2 \end{array} \right.\right) \mid x
\end{aligned}
$$

Does not work:

$$
\left(\begin{array}{l} \texttt{match}\ n\ \texttt{with} \\ \left| \begin{array}{l} \sigma_1\ y_1 \to \lambda z.\, v_1 \\ \sigma_2\ y_2 \to \lambda z.\, v_2 \end{array} \right. \end{array} \right) v
\qquad\qquad
\begin{array}{l} \texttt{match}\ n\ \texttt{with} \\ \left| \begin{array}{l} \sigma_1\ x \to \sigma_2\ x \\ \sigma_2\ x \to \sigma_1\ x \end{array} \right. \end{array}
$$

A last teaser

Define Bool $\stackrel{\text{def}}{=} 1 + 1$.
Suppose f : Bool $\to$ Bool.
Then

A last teaser

Define Bool $\stackrel{\text{def}}{=} 1 + 1$.
Suppose $f : \text{Bool} \to \text{Bool}$.
Then $f \approx f^3$.

A last teaser

Define Bool $\stackrel{\text{def}}{=} 1 + 1$.
Suppose f : Bool $\to$ Bool.
Then $f \approx f^3$.

$$f : \text{Bool} \to \text{Bool}, x : \text{Bool} \vdash f\ x \approx_{\beta\eta} f\ (f\ (f\ x)) : \text{Bool}$$

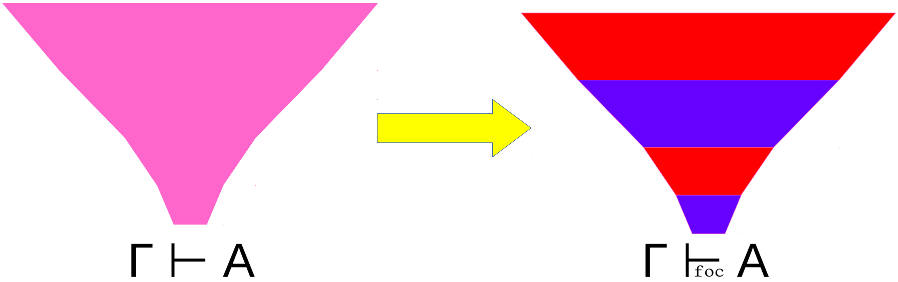
Section 2

Focusing

Focusing

Focusing is a technique from proof theory [Andreoli, 1992].

It studies **invertibility** of connectives
to structure the search space.



$$\frac{\Gamma \vdash \underline{A} \qquad \Gamma, \underline{B} \vdash C}{\Gamma, \underline{A \to B} \vdash C}\, - \qquad\qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

$$\frac{\Gamma, \underline{A_i} \vdash C}{\Gamma, \underline{A_1 \times A_2} \vdash C}\, - \qquad\qquad \frac{\Gamma \vdash A_1 \qquad \Gamma \vdash A_2}{\Gamma \vdash A_1 \times A_2}$$

$$\frac{\Gamma, A_1 \vdash C \qquad \Gamma, A_2 \vdash C}{\Gamma, A_1 + A_2 \vdash C} \qquad\qquad \frac{\Gamma \vdash \underline{A_i}}{\Gamma \vdash \underline{A_1 + A_2}}\, +$$

$$\frac{}{\Gamma, 0 \vdash C}\, + \qquad\qquad \frac{}{\Gamma \vdash 1}\, -$$

Invertible vs. non-invertible rules. Positives vs. negatives.

$$\frac{\Gamma \vdash \underline{A} \qquad \Gamma, \underline{B} \vdash C}{\Gamma, \underline{A \to B} \vdash C}\,- \qquad\qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

$$\frac{\Gamma, \underline{A_i} \vdash C}{\Gamma, \underline{A_1 \times A_2} \vdash C}\,- \qquad\qquad \frac{\Gamma \vdash A_1 \qquad \Gamma \vdash A_2}{\Gamma \vdash A_1 \times A_2}$$

$$\frac{\Gamma, A_1 \vdash C \qquad \Gamma, A_2 \vdash C}{\Gamma, A_1 + A_2 \vdash C} \qquad\qquad \frac{\Gamma \vdash \underline{A_i}}{\Gamma \vdash \underline{A_1 + A_2}}\,+$$

$$\frac{}{\Gamma, 0 \vdash C}\,+ \qquad\qquad \frac{}{\Gamma \vdash 1}\,-$$

Invertible vs. non-invertible rules. Positives vs. negatives.

$$N, M ::= A \to B \mid A \times B \mid 1 \qquad\qquad P, Q ::= A + B \mid 0$$

$$A, B ::= P \mid N \mid \alpha \qquad P_a, Q_a ::= P \mid \alpha \qquad N_a, M_a ::= N \mid \alpha$$

Invertible phase

$$\dfrac{\dfrac{?}{\alpha + \beta \vdash \alpha}}{\alpha + \beta \vdash \beta + \alpha}$$

If applied too early, non-invertible rules can ruin your proof.

Focusing restriction 1: invertible phases

Invertible rules must be applied as soon and as long as possible
– and their order does not matter.

Invertible phase

$$\dfrac{\dfrac{?}{\alpha + \beta \vdash \alpha}}{\alpha + \beta \vdash \beta + \alpha}$$

If applied too early, non-invertible rules can ruin your proof.

Focusing restriction 1: invertible phases

Invertible rules must be applied as soon and as long as possible
– and their order does not matter.

Imposing this restriction gives a single proof of $(\alpha \to \beta) \to (\alpha \to \beta)$ instead of two ($\lambda f.\, f$ and $\lambda f.\, \lambda x.\, f\ x$).

After all invertible rules, negative context Γ_{na}, positive goal P_{a}.

Non-invertible phases

After all invertible rules, negative context, positive goal.

Only step forward: select a formula, apply some non-invertible rule on it.

Non-invertible phases

After all invertible rules, negative context, positive goal.

Only step forward: select a formula, apply some non-invertible rule on it.

Focusing restriction 2: non-invertible phase

When a principal formula is selected for non-invertible rule, they should be applied as long as possible – until its polarity changes.

Non-invertible phases

After all invertible rules, negative context, positive goal.

Only step forward: select a formula, apply some non-invertible rule on it.

Focusing restriction 2: non-invertible phase

When a principal formula is selected for non-invertible rule, they should be applied as long as possible – until its polarity changes.

Completeness: this restriction preserves provability. **Non-trivial !**
Example of removed redundancy:

$$\dfrac{\dfrac{\dfrac{\alpha_2, \qquad \beta_1 \vdash A}{\alpha_2 \times \alpha_3, \qquad \beta_1 \vdash A}}{\alpha_2 \times \alpha_3, \quad \beta_1 \times \beta_2 \vdash A}}{\alpha_1 \times \alpha_2 \times \alpha_3, \beta_1 \times \beta_2 \vdash A}$$

This was focusing:

- invertible as long as a rule matches, until $\Gamma_{\mathsf{na}} \vdash P_{\mathsf{a}}$
- then pick a formula
- then non-invertible as long as a rule matches, until polarity change

Focused inference rules

$$N, M ::= A \to B \mid A \times B \mid 1 \qquad P, Q ::= A + B \mid 0$$

$$A, B ::= P \mid N \mid \alpha \qquad P_{\mathsf{a}}, Q_{\mathsf{a}} ::= P \mid \alpha \qquad N_{\mathsf{a}}, M_{\mathsf{a}} ::= N \mid \alpha$$

$$\Gamma_{\mathsf{na}} ::= \emptyset \mid \Gamma_{\mathsf{na}}, N_{\mathsf{a}}$$

$\Gamma_{\mathsf{na}}; \Delta \vdash_{\mathsf{inv}} A$ invertible phase (decomposes Δ, A)

$\Gamma_{\mathsf{na}} \vdash_{\mathsf{foc}} P_{\mathsf{a}}$ choice of focus

$\Gamma_{\mathsf{na}}, [N] \vdash_{\mathsf{foc.l}} M_{\mathsf{a}}$ non-invertible negative rules

$\Gamma_{\mathsf{na}} \vdash_{\mathsf{foc.r}} [P]$ non-invertible positive rules

Focused sequent calculus

$$\frac{\Gamma_{na}; \Delta, A \vdash_{inv} B}{\Gamma_{na}; \Delta \vdash_{inv} A \to B} \qquad \frac{(\Gamma_{na}; \Delta \vdash_{inv} C_i)^{i \in \{1,2\}}}{\Gamma_{na}; \Delta \vdash_{inv} C_1 \times C_2}$$

$$\frac{(\Gamma_{na}; \Delta, A_i \vdash_{inv} C)^{i \in \{1,2\}}}{\Gamma_{na}; \Delta, A_1 + A_2 \vdash_{inv} C} \qquad \frac{}{\Gamma_{na}; \Delta, 0 \vdash_{inv} C} \qquad \frac{}{\Gamma_{na}; \Delta \vdash_{inv} 1}$$

$$\frac{\Gamma_{na}, \Gamma'_{na} \vdash_{foc} P_a}{\Gamma_{na}; \Gamma'_{na} \vdash_{inv} P_a} \qquad \frac{\Gamma_{na} \vdash_{foc.r} [P]}{\Gamma_{na} \vdash_{foc} P} \qquad \frac{\Gamma_{na}, N, [N] \vdash_{foc.l} P_a}{\Gamma_{na}, N \vdash_{foc} P_a}$$

$$\frac{\Gamma_{na} \vdash_{foc.r} [A_i]}{\Gamma_{na} \vdash_{foc.r} [A_1 + A_2]} \qquad \frac{\Gamma_{na}, [A_i] \vdash_{foc.l} C}{\Gamma_{na}, [A_1 \times A_2] \vdash_{foc.l} C} \qquad \frac{\Gamma_{na} \vdash_{foc.r} [B] \qquad \Gamma_{na}, [A] \vdash_{foc.l} C}{\Gamma_{na}, [B \to A] \vdash_{foc.l} C}$$

$$\frac{}{\Gamma_{na}, [\alpha^-] \vdash_{foc.l} \alpha^-} \qquad \frac{\Gamma_{na}; P \vdash_{inv} C}{\Gamma_{na}, [P] \vdash_{foc.l} C} \qquad \frac{\Gamma_{na}; \emptyset \vdash_{inv} N_a}{\Gamma_{na} \vdash_{foc.r} [N_a]}$$

Focused natural deduction

$$\frac{\Gamma_{\mathsf{na}}; \Delta, A \vdash_{\mathsf{inv}} B}{\Gamma_{\mathsf{na}}; \Delta \vdash_{\mathsf{inv}} A \to B} \qquad \frac{(\Gamma_{\mathsf{na}}; \Delta \vdash_{\mathsf{inv}} A_i)^{i \in \{1,2\}}}{\Gamma_{\mathsf{na}}; \Delta \vdash_{\mathsf{inv}} A_1 \times A_2} \qquad \frac{(\Gamma_{\mathsf{na}}; \Delta, A_i \vdash_{\mathsf{inv}} C)^{i \in \{1,2\}}}{\Gamma_{\mathsf{na}}; \Delta, A_1 + A_2 \vdash_{\mathsf{inv}} C}$$

$$\frac{}{\Gamma_{\mathsf{na}}; \Delta, 0 \vdash_{\mathsf{inv}} C} \qquad \frac{}{\Gamma_{\mathsf{na}}; \Delta \vdash_{\mathsf{inv}} 1} \qquad \frac{\Gamma_{\mathsf{na}}, \Gamma'_{\mathsf{na}} \vdash_{\mathsf{foc}} P_{\mathsf{a}}}{\Gamma_{\mathsf{na}}; \Gamma'_{\mathsf{na}} \vdash_{\mathsf{inv}} P_{\mathsf{a}}}$$

$$\frac{\Gamma_{\mathsf{na}} \Uparrow P}{\Gamma_{\mathsf{na}} \vdash_{\mathsf{foc}} P} \qquad \frac{\Gamma_{\mathsf{na}} \Downarrow \alpha^-}{\Gamma_{\mathsf{na}} \vdash_{\mathsf{foc}} \alpha^-} \qquad \frac{\Gamma_{\mathsf{na}} \Downarrow P \qquad \Gamma_{\mathsf{na}}; P \vdash_{\mathsf{inv}} Q_{\mathsf{a}}}{\Gamma_{\mathsf{na}} \vdash_{\mathsf{foc}} Q_{\mathsf{a}}}$$

$$\frac{\Gamma_{\mathsf{na}} \Uparrow A_i}{\Gamma_{\mathsf{na}} \Uparrow A_1 + A_2} \qquad \frac{\Gamma_{\mathsf{na}} \Downarrow A_1 \times A_2}{\Gamma_{\mathsf{na}} \Downarrow A_i} \qquad \frac{\Gamma_{\mathsf{na}} \Downarrow A \to B \qquad \Gamma_{\mathsf{na}} \Uparrow A}{\Gamma_{\mathsf{na}} \Downarrow B}$$

$$\frac{\Gamma_{\mathsf{na}}; \emptyset \vdash_{\mathsf{inv}} N}{\Gamma_{\mathsf{na}} \Uparrow N} \qquad \frac{}{\Gamma_{\mathsf{na}}, N \Downarrow N}$$

Comparing the two

Just a list reversal.

Example: $P \to (A \times P') \in \Gamma_{\mathsf{na}}$

$$\dfrac{\dfrac{\Gamma_{\mathsf{na}} \vdash_{\mathsf{foc.r}} [P] \qquad \dfrac{\dfrac{\Gamma_{\mathsf{na}} ; P' \vdash_{\mathsf{inv}} Q_{\mathsf{a}}}{\Gamma_{\mathsf{na}}, [P'] \vdash_{\mathsf{foc.l}} Q_{\mathsf{a}}}}{\Gamma_{\mathsf{na}}, [A \times P'] \vdash_{\mathsf{foc.l}} Q_{\mathsf{a}}}}{\Gamma_{\mathsf{na}}, [P \to (A \times P')] \vdash_{\mathsf{foc.l}} Q_{\mathsf{a}}}}{\Gamma_{\mathsf{na}} \vdash_{\mathsf{foc}} Q_{\mathsf{a}}}$$

$$\dfrac{\dfrac{\dfrac{\Gamma_{\mathsf{na}} \Downarrow P \to (A \times P') \qquad \Gamma_{\mathsf{na}} \Uparrow P}{\Gamma_{\mathsf{na}} \Downarrow A \times P'}}{\Gamma_{\mathsf{na}} \Downarrow P'} \qquad \Gamma_{\mathsf{na}} ; P' \vdash_{\mathsf{inv}} Q_{\mathsf{a}}}{\Gamma_{\mathsf{na}} \vdash_{\mathsf{foc}} Q_{\mathsf{a}}}$$

Completeness

$$\Gamma \vdash A \qquad \Longrightarrow \qquad \emptyset; \Gamma \vdash_{\mathsf{inv}} A$$

(Possible proof: by translation to linear logic, being careful about exponential placement.)

Section 3

Focused λ-calculus

Reminder: β-normal forms (negative)

β-short normal forms:

$$\pi_1\ (t, u) = t$$

$$\begin{array}{rcl} v, w & ::= & \lambda x.\, v \mid (v, w) \mid n \\ n, m & ::= & \pi_i\ n \mid n\ v \mid x \end{array}$$

Reminder: β-normal forms (negative)

β-short normal forms:

$$\pi_1\ (t, u) = t$$

$$\begin{array}{rcl} v, w & ::= & \lambda x.\, v \mid (v, w) \mid n \\ n, m & ::= & \pi_i\ n \mid n\ v \mid x \end{array}$$

β-short η-long:

$$(y : \alpha \to \beta) = \lambda x : \alpha.\, (y\ x : \beta)$$

Reminder: β-normal forms (negative)

β-short normal forms:

$$\pi_1\ (t, u) = t$$

$$\begin{array}{rcl} v, w & ::= & \lambda x.\, v \mid (v, w) \mid n \\ n, m & ::= & \pi_i\ n \mid n\ v \mid x \end{array}$$

β-short η-long:

$$(y : \alpha \to \beta) = \lambda x : \alpha.\, (y\ x : \beta)$$

$$\begin{array}{rcl} v, w & ::= & \lambda x.\, v \mid (v, w) \mid (n : \alpha) \\ n, m & ::= & \pi_i\ n \mid n\ v \mid x \end{array}$$

Reminder: What about sums?

$$
\begin{aligned}
v, w &::= \lambda x.\, v \mid (v, w) \mid \sigma_i\ v \mid (n : \alpha) \\
n, m &::= \pi_i\ n \mid n\ v \mid \left(\texttt{match}\ n\ \texttt{with} \left|\ \begin{array}{l} \sigma_1\ y_1 \to v_1 \\ \sigma_2\ y_2 \to v_2 \end{array}\right.\right) \mid x
\end{aligned}
$$

Does not work:

$$
\left(\begin{array}{l}\texttt{match}\ n\ \texttt{with} \\ \left|\ \begin{array}{l}\sigma_1\ y_1 \to \lambda z.\, v_1 \\ \sigma_2\ y_2 \to \lambda z.\, v_2\end{array}\right.\end{array}\right) v
\qquad\qquad
\begin{array}{l}\texttt{match}\ n\ \texttt{with} \\ \left|\ \begin{array}{l}\sigma_1\ x \to \sigma_2\ x \\ \sigma_2\ x \to \sigma_1\ x\end{array}\right.\end{array}
$$

Focusing to the rescue

$$\begin{array}{rcl} v, w & ::= & \lambda x.\, v \mid (v, w) \mid (n : \alpha) \\ n, m & ::= & \pi_i\ n \mid n\ v \mid x \end{array}$$

$$\Downarrow$$

$$\begin{array}{rcl} v, w & ::= & \lambda x.\, v \mid (v, w) \mid () \\ & & \mid \mathtt{absurd}(x) \mid \left(\mathtt{match}\ x\ \mathtt{with} \,\middle|\, \begin{array}{l} \sigma_1\ y_1 \to v_1 \\ \sigma_2\ y_2 \to v_2 \end{array}\right) \\ & & \mid (\Gamma_{\mathsf{na}} \vdash f : P_{\mathsf{a}}) \\ \\ n, m & ::= & \pi_i\ n \mid n\ p \mid x \\ p, q & ::= & \sigma_i\ p \mid (v : N_{\mathsf{a}}) \\ \\ f & ::= & (n : \alpha) \mid (p : P) \mid \mathtt{let}\ x = (n : P)\ \mathtt{in}\ v \end{array}$$

(See also Munch-Maccagnoni [2013])

Focused λ-calculus

$$\frac{\Gamma_{na}; \Delta, x : A \vdash_{inv} t : B}{\Gamma_{na}; \Delta \vdash_{inv} \lambda x.\, t : A \to B} \qquad \frac{(\Gamma_{na}; \Delta \vdash_{inv} t_i : A_i)^{i \in \{1,2\}}}{\Gamma_{na}; \Delta \vdash_{inv} (t_1, t_2) : A_1 \times A_2} \qquad \frac{}{\Gamma_{na}; \Delta \vdash_{inv} () : 1}$$

$$\frac{(\Gamma_{na}; \Delta, x : A_i \vdash_{inv} t_i : C\ |)^{i \in \{1,2\}}}{\Gamma_{na}; \Delta, x : A_1 + A_2 \vdash_{inv} \begin{array}{l} \texttt{match}\ x\ \texttt{with} \\ \mid \sigma_1\ x \to t_1 \\ \mid \sigma_2\ x \to t_2 \end{array} : C} \qquad \frac{}{\Gamma_{na}; x : \Delta, 0 \vdash_{inv} \texttt{absurd}(x) : C}$$

$$\frac{\Gamma_{na}, \Gamma'_{na} \vdash_{foc} f : Q_a}{\Gamma_{na}; \Gamma'_{na} \vdash_{inv} f : Q_a} \qquad \frac{\Gamma_{na} \vdash n \Downarrow \alpha^-}{\Gamma_{na} \vdash_{foc} n : \alpha^-} \qquad \frac{\Gamma_{na} \vdash n \Downarrow P \qquad \Gamma_{na}; x : P \vdash_{inv} t : Q_a}{\Gamma_{na} \vdash_{foc} \texttt{let}\ x = n\ \texttt{in}\ t : Q_a}$$

$$\frac{}{\Gamma_{na}, x : N \vdash x \Downarrow N} \qquad \frac{\Gamma_{na}; \emptyset \vdash_{inv} t : N}{\Gamma_{na} \vdash t \Uparrow N} \qquad \frac{\Gamma_{na} \vdash p \Uparrow P}{\Gamma_{na} \vdash_{foc} p : P} \qquad \frac{\Gamma_{na} \vdash n \Downarrow A_1 \times A_2}{\Gamma_{na} \vdash \pi_i\ n \Downarrow A_i}$$

$$\frac{\Gamma_{na} \vdash n \Downarrow A \to B \qquad \Gamma_{na} \vdash p \Uparrow A}{\Gamma_{na} \vdash n\ p \Downarrow B} \qquad \frac{\Gamma_{na} \vdash p \Uparrow A_i}{\Gamma_{na} \vdash \sigma_i\ p \Uparrow A_1 + A_2}$$

Completeness of focusing

Logic:

$$\Gamma \vdash A \qquad \Longrightarrow \qquad \Gamma \vdash_{\texttt{foc}} A$$

Completeness of focusing

Logic:

$$\Gamma \vdash A \qquad \Longrightarrow \qquad \Gamma \vdash_{\texttt{foc}} A$$

Programming:

$$\Gamma \vdash t : A \qquad \Longrightarrow \qquad \exists v, \begin{array}{c} \Gamma \vdash_{\texttt{foc}} v : A \\ v \approx_{\beta\eta} t \end{array}$$

Canonicity

Focused normal forms are canonical for the impure λ-calculus.

Proof in Zeilberger [2009], using ideas from Girard's ludics.

Canonicity

Focused normal forms are canonical for the impure λ-calculus.

Proof in Zeilberger [2009], using ideas from Girard's ludics.

Not canonical for the **pure** calculus.

$$\texttt{let}\ x = n\ \texttt{in}\ C\left[\texttt{let}\ x' = n'\ \texttt{in}\ v\right]$$

$$\texttt{let}\ x' = n'\ \texttt{in}\ C\left[\texttt{let}\ x = n\ \texttt{in}\ v\right]$$

Section 4

Maximal multi-focusing, saturation

Multi-focusing

Idea: have several **foci** in parallel in each non-invertible phase.

$$\frac{\Gamma_{\mathsf{na}}, \Gamma'_{\mathsf{na}}, [\Gamma'_{\mathsf{na}}] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [\Sigma'_{\mathsf{na}}]}{\Gamma_{\mathsf{na}}, \Gamma'_{\mathsf{na}} \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, \Sigma'_{\mathsf{na}}}$$

$$\frac{\Gamma_{\mathsf{na}}, [\Delta] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [A_i]}{\Gamma_{\mathsf{na}}, [\Delta] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [A_1 + A_2]} \qquad \frac{\Gamma_{\mathsf{na}}, [\Delta, A_i] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [\Delta']}{\Gamma_{\mathsf{na}}, [\Delta, A_1 \times A_2] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [\Delta']}$$

$$\frac{\Gamma_{\mathsf{na}}, [\Delta] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [B] \qquad \Gamma_{\mathsf{na}}, [A] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [\Delta']}{\Gamma_{\mathsf{na}}, [\Delta, B \to A] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [\Delta']}$$

$$\frac{}{\Gamma_{\mathsf{na}}, [\alpha^-] \vdash_{\mathsf{foc}} \alpha^-, [\emptyset]} \qquad \frac{\Gamma_{\mathsf{na}}; \Sigma_{\mathsf{p}} \vdash_{\mathsf{inv}} \Sigma_{\mathsf{pa}}, \Gamma'_{\mathsf{na}}}{\Gamma_{\mathsf{na}}, [\Sigma_{\mathsf{p}}] \vdash_{\mathsf{foc}} \Sigma_{\mathsf{pa}}, [\Gamma'_{\mathsf{na}}]}$$

Maximal multi-focusing

Maximal parallelism among permutation-equivalent proofs.

Good: Canonical for linear, intuitionistic, classical logic without units.

$$\texttt{let}\ x = n\ \texttt{in}\ C\left[\texttt{let}\ x' = n'\ \texttt{in}\ v\right]$$

$$\texttt{let}\ x' = n'\ \texttt{in}\ C\left[\texttt{let}\ x = n\ \texttt{in}\ v\right]$$

$$\Longrightarrow$$

$$\texttt{let}\ x, x' = n, n'\ \texttt{in}\ C\left[v\right]$$

Bad: no goal-directed structure.

Saturation

$$\texttt{let}\ x = n\ \texttt{in}\ C\left[\texttt{let}\ x' = n'\ \texttt{in}\ v\right]$$

$$\texttt{let}\ x' = n'\ \texttt{in}\ C\left[\texttt{let}\ x = n\ \texttt{in}\ v\right]$$

$$\Longrightarrow$$

$$\texttt{let}\ x, x' = n, n'\ \texttt{in}\ C\left[v\right]$$

We want the $\texttt{let}\ x = n$ to be "as early as possible" – maximal multi-focusing. "Split neutrals early".

Idea: split on **all** possible neutrals.

$$
\begin{array}{lll}
v, w & ::= & \lambda x.\, v \mid (v, w) \mid () \\
 & & \mid () \mid \texttt{absurd}(x) \mid \left(\texttt{match}\ x\ \texttt{with} \;\middle|\; \begin{array}{l} \sigma_1\, y_1 \rightarrow v_1 \\ \sigma_2\, y_2 \rightarrow v_2 \end{array} \right) \\
 & & \mid (\Gamma_{\mathrm{na}} \vdash f : P_{\mathrm{a}}) \\
n, m & ::= & \pi_i\, n \mid n\, p \mid x \\
p, q & ::= & \sigma_i\, p \mid (v : N_{\mathrm{a}}) \\
f & ::= & \texttt{let}\ \bar{x} = \bar{n}\ \texttt{in}\ v \mid (n : \alpha) \mid (p : P)
\end{array}
$$

Plus side-condition on the $\texttt{let}\ \bar{x} = \bar{n}$:

- they are a set (no duplicates)
- **freshness**: must use a variable of the preceding invertible phase v
- **saturation**: $n \mid p$ can only be chosen if no fresh variable

Saturation rules

$$\frac{\Gamma_{na}; \Gamma'_{na} \vdash_{sat} f : P_a}{\Gamma_{na}; \Gamma'_{na} \vdash_{sinv} f : P_a \mid} \qquad \frac{\Gamma_{na} \vdash_s p \Uparrow P}{\Gamma_{na}; \emptyset \vdash_{sat} p : P} \qquad \frac{\Gamma_{na} \vdash_s n \Downarrow \alpha^-}{\Gamma_{na}; \emptyset \vdash_{sat} n : \alpha^-}$$

$$\frac{(\bar{n}, \bar{P}) \stackrel{\text{def}}{=} \Phi(\Gamma_{na}, \Gamma'_{na})\left\{(n, P) \mid \begin{array}{l} (\Gamma_{na}, \Gamma'_{na} \vdash_s n \Downarrow P) \\ \wedge\ \exists x \in \Gamma'_{na}, x \in n \end{array}\right\} \qquad \Gamma_{na}, \Gamma'_{na}; \bar{x} : \bar{P} \vdash_{sinv} t : \emptyset \mid Q_a}{\Gamma_{na}; \Gamma'_{na} \vdash_{sat} \texttt{let } \bar{x} = \bar{n} \texttt{ in } t : Q_a}$$

$\Phi(\Gamma_{na})(E)$ is a **horizon** parameter for the type system, returning a finite set of neutrals to split.

Local completeness

$$(\Gamma \vdash_{\texttt{foc}} v : A) \quad \Longrightarrow \quad \exists \Phi, v', \begin{array}{c} \Gamma \vdash_{\texttt{sat}:\Phi} v' : A \\ v \approx_{\beta\eta} v' \end{array}$$

Empty type?

$$f : 1 \to \beta, g : \beta \to 0, x : \alpha, y : \alpha \vdash ? : \alpha$$

x, y would be bad saturated terms.

Empty type?

$$f : 1 \to \beta, g : \beta \to 0, x : \alpha, y : \alpha \vdash ? : \alpha$$

x, y would be bad saturated terms.

Additional condition on Φ:

$$(\exists n,\ \Gamma \vdash_{\mathtt{foc}} n : P) \qquad \Longrightarrow \qquad (\exists n \in \Phi(\Gamma),\ \Gamma \vdash_{\mathtt{foc}} n : P)$$

Idea: set of those P is finite – subformula property.
Idea: complete for provability.

Canonicity

$\Gamma \vdash_{\mathtt{sat}:\Phi} v, w : A$
$v \not\approx_{\alpha} w$

$\Longrightarrow$

$v \not\approx_{\mathtt{ctx}} w$

(The hard part.)

Canonicity

$$\begin{array}{l}\Gamma \vdash_{\mathtt{sat:\Phi}} v, w : A \\ v \not\approx_{\alpha} w\end{array} \implies v \not\approx_{\mathtt{ctx}} w$$

(The hard part.)

Corollary: $(\approx_{\beta\eta}) = (\approx_{\mathtt{ctx}})$

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\ (\sigma_1\ ()),\ n\ (\sigma_2\ \ (n\ \sigma_1\ ())\ ) : \alpha$$

Saturated forms:

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\ (\sigma_1\ ()),\ n\ (\sigma_2\ \ (n\ \sigma_1\ ())\ ) : \alpha$$

Saturated forms:

$$n : (1 + \alpha) \to \alpha \vdash \qquad \not\approx_{\texttt{stx}} \qquad : \alpha$$

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ (n\,\sigma_1\,())\,) : \alpha$$

Saturated forms:

$$n : (1 + \alpha) \to \alpha \vdash \qquad \not\approx_{\mathtt{stx}} \qquad : \alpha$$

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\,(\sigma_1\,()),\; n\,(\sigma_2\,(n\,\sigma_1\,()))) : \alpha$$

Saturated forms:

$$\texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in}$$

$$n : (1 + \alpha) \to \alpha \vdash \qquad \not\approx_{\texttt{stx}} \qquad : \alpha$$

$$\texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in}$$

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\; n\,(\sigma_2\;\,(n\,\sigma_1\,())\;) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \end{array} : \alpha$$

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ (n\,\sigma_1\,()))\ : \alpha$$

Saturated forms:

$$\texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in}$$

$$n : (1 + \alpha) \to \alpha \vdash \qquad \not\approx_{\texttt{stx}} \qquad : \alpha$$

$$\texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in}$$

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ (n\,\sigma_1\,()))\ : \alpha$$

Saturated forms:

$$n : (1 + \alpha) \to \alpha \vdash \begin{array}{l} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in} \\ \qquad\qquad \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in} \end{array} : \alpha$$

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\,(\sigma_1\,()),\; n\,(\sigma_2\;(n\,\sigma_1\,())\;) : \alpha$$

Saturated forms:

$$n : (1 + \alpha) \to \alpha \vdash \begin{array}{l} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in} \\ \qquad \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in} \end{array} : \alpha$$

Shared context.

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ (n\,\sigma_1\,())\ ) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{l} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ \square \\ \qquad\qquad \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ \square \end{array} : \alpha$$

Shared context. Source of inequality:

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ (n\,\sigma_1\,())\ ) : \alpha$$

Saturated forms:

$$n : (1 + \alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\;()),\; n\,(\sigma_2\;\;(n\,\sigma_1\;())\;) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\;())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\;())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\,(\sigma_1\,()),\; n\,(\sigma_2\;\,(n\,\sigma_1\,())\;) : \alpha$$

Saturated forms:

$$n : (1 + \alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.
Type variables:

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ (n\,\sigma_1\,())\ ) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{l} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \qquad\qquad \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.
Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ \ (n\,\sigma_1\,())\ ) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.
Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.

Here,

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\; n\,(\sigma_2\;(n\,\sigma_1\,())\;) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{l} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \qquad\qquad \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.
Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.
Here, $\hat{\alpha} \stackrel{\text{def}}{=} 1 + 1$, $\hat{z} \stackrel{\text{def}}{=} \sigma_1\,()$ and $\hat{o} \stackrel{\text{def}}{=} \sigma_2\,()$.

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ (n\,\sigma_1\,())\ ) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.

Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.

Here, $\hat{\alpha} \stackrel{\text{def}}{=} 1 + 1$, $\hat{z} \stackrel{\text{def}}{=} \sigma_1\,()$ and $\hat{o} \stackrel{\text{def}}{=} \sigma_2\,()$.

Separating context: $C\,[\square] \stackrel{\text{def}}{=} (\lambda n.\,\square)\ \hat{n}$

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ \ (n\,\sigma_1\,())\ ) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.
Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.
Here, $\hat{\alpha} \stackrel{\text{def}}{=} 1 + 1$, $\hat{z} \stackrel{\text{def}}{=} \sigma_1\,()$ and $\hat{o} \stackrel{\text{def}}{=} \sigma_2\,()$.
Separating context: $C\,[\square] \stackrel{\text{def}}{=} (\lambda n.\,\square)\ \hat{n}$

$$\hat{n} \stackrel{\text{def}}{=} \left\{ \right.$$

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\; n\,(\sigma_2\;(n\,\sigma_1\,())\,) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.

Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.

Here, $\hat{\alpha} \overset{\text{def}}{=} 1 + 1$, $\hat{z} \overset{\text{def}}{=} \sigma_1\,()$ and $\hat{o} \overset{\text{def}}{=} \sigma_2\,()$.

Separating context: $C\,[\square] \overset{\text{def}}{=} (\lambda n.\,\square)\;\hat{n}$

$$\hat{n} \overset{\text{def}}{=} \Bigg\{$$

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ \ (n\,\sigma_1\,())\ ) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.
Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.
Here, $\hat{\alpha} \overset{\text{def}}{=} 1 + 1$, $\hat{z} \overset{\text{def}}{=} \sigma_1\,()$ and $\hat{o} \overset{\text{def}}{=} \sigma_2\,()$.
Separating context: $C\,[\Box] \overset{\text{def}}{=} (\lambda n.\,\Box)\ \hat{n}$

$$\hat{n} \overset{\text{def}}{=} \begin{cases} \sigma_1\,() & \mapsto \\ & \end{cases}$$

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\ (\sigma_1\ ()),\ n\ (\sigma_2\ \ (n\ \sigma_1\ ())\ ) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\ (\sigma_1\ ())\ \texttt{in} \\ \quad \texttt{let}\ o = n\ (\sigma_2\ z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\ (\sigma_1\ ())\ \texttt{in} \\ \quad \texttt{let}\ o = n\ (\sigma_2\ z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.

Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.

Here, $\hat{\alpha} \stackrel{\text{def}}{=} 1+1$, $\hat{z} \stackrel{\text{def}}{=} \sigma_1\ ()$ and $\hat{o} \stackrel{\text{def}}{=} \sigma_2\ ()$.

Separating context: $C\,[\Box] \stackrel{\text{def}}{=} (\lambda n.\,\Box)\ \hat{n}$

$$\hat{n} \stackrel{\text{def}}{=} \begin{cases} \sigma_1\ () & \mapsto \\ & \end{cases}$$

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\; n\,(\sigma_2\;(n\,\sigma_1\,())\,) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\; z = n\,(\sigma_1\,())\;\texttt{in} \\ \quad \texttt{let}\; o = n\,(\sigma_2\,z)\;\texttt{in}\; z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\; z = n\,(\sigma_1\,())\;\texttt{in} \\ \quad \texttt{let}\; o = n\,(\sigma_2\,z)\;\texttt{in}\; o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.
Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.
Here, $\hat{\alpha} \stackrel{\text{def}}{=} 1+1$, $\hat{z} \stackrel{\text{def}}{=} \sigma_1\,()$ and $\hat{o} \stackrel{\text{def}}{=} \sigma_2\,()$.
Separating context: $C\,[\Box] \stackrel{\text{def}}{=} (\lambda n.\,\Box)\;\hat{n}$

$$\hat{n} \stackrel{\text{def}}{=} \begin{cases} \sigma_1\,() & \mapsto \quad \hat{z} \\ & \end{cases}$$

Canonicity: example

$$n : (1 + \alpha) \to \alpha \vdash n\,(\sigma_1\,()),\ n\,(\sigma_2\ (n\,\sigma_1\,())\ ) : \alpha$$

Saturated forms:

$$n : (1 + \alpha) \to \alpha \vdash \begin{array}{l} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\ z)\ \texttt{in}\ z \\ \qquad \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\ z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.

Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.

Here, $\hat{\alpha} \stackrel{\text{def}}{=} 1 + 1$, $\hat{z} \stackrel{\text{def}}{=} \sigma_1\,()$ and $\hat{o} \stackrel{\text{def}}{=} \sigma_2\,()$.

Separating context: $C\,[\square] \stackrel{\text{def}}{=} (\lambda n.\,\square)\ \hat{n}$

$$\hat{n} \stackrel{\text{def}}{=} \begin{cases} \sigma_1\,() & \mapsto \quad \hat{z} \\ & \end{cases}$$

Canonicity: example

$$n : (1+\alpha) \to \alpha \vdash n\,(\sigma_1\,()),\; n\,(\sigma_2\;\;(n\,\sigma_1\,())\;) : \alpha$$

Saturated forms:

$$n : (1+\alpha) \to \alpha \vdash \begin{array}{c} \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ z \\ \not\approx_{\texttt{stx}} \\ \texttt{let}\ z = n\,(\sigma_1\,())\ \texttt{in} \\ \quad \texttt{let}\ o = n\,(\sigma_2\,z)\ \texttt{in}\ o \end{array} : \alpha$$

Shared context. Source of inequality: $z \not\approx_{\texttt{stx}} o$.
Type variables: pick a finite type of codes of the form $1 + (1 + \ldots)$.
Here, $\hat{\alpha} \overset{\text{def}}{=} 1+1$, $\hat{z} \overset{\text{def}}{=} \sigma_1\,()$ and $\hat{o} \overset{\text{def}}{=} \sigma_2\,()$.
Separating context: $C\,[\Box] \overset{\text{def}}{=} (\lambda n.\,\Box)\;\hat{n}$

$$\hat{n} \overset{\text{def}}{=} \begin{cases} \sigma_1\,() & \mapsto \quad \hat{z} \\ \sigma_2\,\hat{z} & \mapsto \quad \hat{o} \end{cases}$$

Looking back: applications of proof theory

A clean way to extend our understanding to positives ($+$, 0).

- evaluation order in presence of effects
- which types have a unique inhabitant?
- decidability of equivalence
- Böhm separation results: contextual and ($\beta\eta$) coincide
- λ-definability?

Thanks. Questions?

Thorsten Altenkirch, Peter Dybjer, Martin Hofmann, and Philip J. Scott. Normalization by evaluation for typed lambda calculus with coproducts. In **LICS**, 2001.

Jean-Marc Andreoli. Logic Programming with Focusing Proof in Linear Logic. **Journal of Logic and Computation**, 2(3), 1992.

Vincent Balat, Roberto Di Cosmo, and Marcelo P. Fiore. Extensional normalisation and type-directed partial evaluation for typed lambda calculus with sums. In **POPL**, 2004.

Neil Ghani. Beta-Eta Equality for Coproducts. In **TLCA**, 1995.

Guillaume Munch-Maccagnoni. **Syntax and Models of a non-Associative Composition of Programs and Proofs**. PhD thesis, Univ. Paris Diderot, 2013.

Noam Zeilberger. **The Logical Basis of Evaluation Order and Pattern-Matching**. PhD thesis, Carnegie Mellon University, 2009.