Canonical typed representations – and synthesis?

Gabriel Scherer

Northeastern University, Boston

Talk to the Computer Assisted Programming Group, MIT
March 1, 2017

A question

"Which types have a unique inhabitant (modulo program equivalence)?"

Why?

Understand fundamental properties of **non-ambiguous** code **inference** mechanisms:

- static overloading
- type classes
- implicit parameters (maybe)

Idea: **predictability** corresponds to finding a program with a **unique** solution.
(Program synthesis?)

In practice: I understand simple type systems and equivalence better, more work to extend to realistic languages (first-class polymorphism).

How?

Naive idea: enumerate programs, stop at two.

Problem: what about redundancies?
Redundancy: two (syntactically) distinct terms that are equivalent.

Goal: Enumerate programs **without duplicates**.

Define a **representation** of programs without redundancy – canonical.

A side-result

Simply-typed λ-calculus without polymorphism.

$$A, B \quad ::= \quad A \to B \mid A \times B \mid A + B \mid 1 \mid 0$$

If you have **canonical** representations, it's easy to decide equivalence.

Side-result: decision procedure for equivalence in this type system.
(Was an open problem because of 0.)

What about synthesis?

My poor understanding of program synthesis:

- searching a large space until you find a solution
- try to reduce the space by inverting the shape of the solution

Canonical forms reduce the search space by eliminating redundancies. Obvious win?

Yes: many of Peter Michael-Osera and Steve Zdancewic's 2015 heuristics are instances of focusing simplifications.

No: sometimes canonical forms require bookkeeping, or their normal forms are too large, and it can hurt performance. (Same as proof search.)

In any case, understanding the relations between my theory and your practice would be nice.

Section 2

Background

Simply-typed lambda-calculus

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

$$\frac{\Gamma \vdash t_1 : A_1 \quad \Gamma \vdash t_2 : A_2}{\Gamma \vdash (t_1, t_2) : A_1 \times A_2} \qquad \frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \pi_i\ t : A_i}$$

$$\frac{\Gamma \vdash t : A_i}{\Gamma \vdash \sigma_i\ t : A_1 + A_2} \qquad \frac{\Gamma \vdash t : A_1 + A_2 \qquad \begin{array}{c}\Gamma, x_1 : A_1 \vdash u_1 : C \\ \Gamma, x_2 : A_2 \vdash u_2 : C\end{array}}{\Gamma \vdash \texttt{match}\ t\ \texttt{with} \mid \begin{array}{l}\sigma_1\ x_1 \to u_1 \\ \sigma_2\ x_2 \to u_2\end{array} : C}$$

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

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta \begin{array}{l} \texttt{match}\ t\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \mid\ \sigma_2\ x_2 \to \sigma_2\ x_2 \end{array}}$$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \mid \sigma_1\ x_1 \to u_1 \mid \sigma_2\ x_2 \to u_2 \ \triangleright_\beta\ u_i[t/x_i]$$

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

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta \begin{array}{l}\texttt{match}\ t\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \mid\ \sigma_2\ x_2 \to \sigma_2\ x_2\end{array}}$$

$$(t_1, t_2) \overset{?}{\approx}_\eta \begin{array}{l}\texttt{match}\ t_1\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to (\sigma_1\ x_1, t_2) \\ \mid\ \sigma_2\ x_2 \to (\sigma_2\ x_2, t_2)\end{array}$$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \mid \sigma_1\ x_1 \to u_1 \mid \sigma_2\ x_2 \to u_2 \ \triangleright_\beta\ u_i[t/x_i]$$

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

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta \begin{array}{l}\texttt{match}\ t\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \mid\ \sigma_2\ x_2 \to \sigma_2\ x_2\end{array}}$$

$$(t_1, t_2) \overset{?}{\approx}_\eta \begin{array}{l}\texttt{match}\ t_1\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to (\sigma_1\ x_1, t_2) \\ \mid\ \sigma_2\ x_2 \to (\sigma_2\ x_2, t_2)\end{array}$$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \mid \sigma_1\ x_1 \to u_1 \mid \sigma_2\ x_2 \to u_2 \ \triangleright_\beta\ u_i[t/x_i]$$

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

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta \begin{array}{l}\texttt{match}\ t\ \texttt{with} \\ \mid \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \mid \sigma_2\ x_2 \to \sigma_2\ x_2\end{array}}$$

$$(t_1, t_2) \overset{?}{\approx}_\eta \begin{array}{l}\texttt{match}\ t_1\ \texttt{with} \\ \mid \sigma_1\ x_1 \to (\sigma_1\ x_1, t_2) \\ \mid \sigma_2\ x_2 \to (\sigma_2\ x_2, t_2)\end{array}$$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \mid \sigma_1\ x_1 \to u_1 \mid \sigma_2\ x_2 \to u_2 \ \triangleright_\beta\ u_i[t/x_i]$$

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

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta \ \texttt{match}\ t\ \texttt{with} \mid \sigma_1\ x_1 \to \sigma_1\ x_1 \mid \sigma_2\ x_2 \to \sigma_2\ x_2}$$

$$(t_1, t_2) \overset{?}{\approx}_\eta \ \texttt{match}\ t_1\ \texttt{with} \mid \sigma_1\ x_1 \to (\sigma_1\ x_1, t_2) \mid \sigma_2\ x_2 \to (\sigma_2\ x_2, t_2)$$

$$u[t_1/y] \quad \text{with} \quad u \overset{\text{def}}{=} (y, t_2)$$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \mid \sigma_1\ x_1 \to u_1 \mid \sigma_2\ x_2 \to u_2 \ \triangleright_\beta\ u_i[t/x_i]$$

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

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta \begin{array}{l}\texttt{match}\ t\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \mid\ \sigma_2\ x_2 \to \sigma_2\ x_2\end{array}}$$

$$u[t_1/y] \overset{?}{\approx}_\eta \begin{array}{l}\texttt{match}\ t_1\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to (\sigma_1\ x_1, t_2) \\ \mid\ \sigma_2\ x_2 \to (\sigma_2\ x_2, t_2)\end{array}$$

$u[t_1/y]$ with $u \overset{\text{def}}{=} (y, t_2)$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

$$\texttt{match}\ \sigma_i\ t\ \texttt{with} \mid \sigma_1\ x_1 \to u_1 \mid \sigma_2\ x_2 \to u_2 \ \triangleright_\beta\ u_i[t/x_i]$$

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

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta \begin{array}{l}\texttt{match}\ t\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \mid\ \sigma_2\ x_2 \to \sigma_2\ x_2\end{array}}$$

$$u[t_1/y] \overset{?}{\approx}_\eta \begin{array}{l}\texttt{match}\ t_1\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to u[\sigma_1\ x_1/y] \\ \mid\ \sigma_2\ x_2 \to u[\sigma_2\ x_2/y]\end{array}$$

$u[t_1/y]$ with $u \overset{\text{def}}{=} (y, t_2)$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

$$\frac{\Gamma \vdash t : A_1 + A_2}{t \triangleright_\eta \begin{array}{l} \texttt{match}\ t\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to \sigma_1\ x_1 \\ \mid\ \sigma_2\ x_2 \to \sigma_2\ x_2 \end{array}}$$

$$u[t_1/y] \approx_\eta \begin{array}{l} \texttt{match}\ t_1\ \texttt{with} \\ \mid\ \sigma_1\ x_1 \to u[\sigma_1\ x_1/y] \\ \mid\ \sigma_2\ x_2 \to u[\sigma_2\ x_2/y] \end{array}$$

$$u[t_1/y] \quad \text{with} \quad u \stackrel{\text{def}}{=} (y, t_2)$$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

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

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

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

$$\frac{\Gamma \vdash t : 0 \qquad \Gamma,\ y : 0 \vdash u : C}{u[t/y] \triangleright_\eta \texttt{absurd}(t)}$$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

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

Derived rules :

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

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

$$\frac{\Gamma \vdash t : 0 \qquad \Gamma,\ y : 0 \vdash u : C}{u[t/y] \triangleright_\eta \texttt{absurd}(t)}$$

Derived rules : $$\frac{}{\Gamma \vdash t_1 \approx_\eta t_2 : 1}$$

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

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

$$\frac{\Gamma \vdash t : 0 \qquad \Gamma, y : 0 \vdash u : C}{u[t/y] \triangleright_\eta \texttt{absurd}(t)}$$

Derived rules :

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

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

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

$$\frac{\Gamma \vdash t : 0 \qquad \Gamma, y : 0 \vdash u : C}{u[t/y] \triangleright_\eta \texttt{absurd}(t)}$$

Derived rules :

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

Simply-typed $\beta\eta$-equivalence; Why is it difficult?

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

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

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

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

$$\frac{\Gamma \vdash t : 0 \qquad \Gamma, y : 0 \vdash u : C}{u[t/y] \triangleright_\eta \texttt{absurd}(t)}$$

Derived rules :

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

Section 3

High-level view

Question

What is a **canonical form** for equivalence of simply-typed terms?

Redundancy: two (syntactically) distinct terms that are equivalent.

Canonical representation: a syntax of programs with no redundancy:

$$(\not\approx_{\mathtt{stx}}) \implies (\not\approx_{\beta\eta})$$

Question

What is a **canonical form** for equivalence of simply-typed terms?

Redundancy: two (syntactically) distinct terms that are equivalent.

Canonical representation: a syntax of programs with no redundancy:

$$(\not\approx_{\mathtt{stx}}) \implies (\not\approx_{\beta\eta})$$

With only functions and pairs, there is a reasonable notion of β-short η-long normal form.

Question

What is a **canonical form** for equivalence of simply-typed terms?

Redundancy: two (syntactically) distinct terms that are equivalent.

Canonical representation: a syntax of programs with no redundancy:

$$(\not\approx_{\mathtt{stx}}) \implies (\not\approx_{\beta\eta})$$

With only functions and pairs, there is a reasonable notion of β-short η-long normal form. It does not scale to sums.

Question

What is a **canonical form** for equivalence of simply-typed terms?

Redundancy: two (syntactically) distinct terms that are equivalent.

Canonical representation: a syntax of programs with no redundancy:

$$(\not\approx_{\mathtt{stx}}) \implies (\not\approx_{\beta\eta})$$

With only functions and pairs, there is a reasonable notion of β-short η-long normal form. It does not scale to sums.

Normal form (for reduction) $\neq$ Canonical form (for equivalence)

(see also Watkins, Cervesato, Pfenning, Walker, 2002)

Idea

Curry-Howard, again: programs as proofs.

The structure of

canonical forms

corresponds to the structure of

proof **search**

Restricting the search space restricts expression redundancy.

Proof search: Focusing

(existing work)

Gives a term representation ($\vdash_{\mathtt{foc}}$).

Canonical for **effectful** programs.
(Noam Zeilberger's thesis, 2009)
Not canonical for pure programs (stronger equivalences).

Complete: any term can be focused.

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

Proof search: Focusing

(existing work)

Gives a term representation ($\vdash_{\mathtt{foc}}$).

Canonical for **effectful** programs.
(Noam Zeilberger's thesis, 2009)
Not canonical for pure programs (stronger equivalences).

Complete: any term can be focused.

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

$$\Gamma \vdash t : A \qquad \Longrightarrow \qquad \exists v \approx_{\beta\eta} t, \quad \Gamma \vdash_{\mathtt{foc}} v : A$$

Proof search: Saturation

(my contribution)

Family of representations ($\vdash_{\mathtt{sat}:\Phi}$).

Canonical for **pure** programs.

Locally complete: for any finite set of terms,
there is a Φ such that ($\vdash_{\mathtt{sat}:\Phi}$) is complete.


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


Kaustuv Chaudhuri, Dale Miller, and Alexis Saurin. "Canonical Sequent Proofs via Multi-Focusing". IFIP TCS, 2008.


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


Peter-Michael Osera and Steve Zdancewic. "Type-and-Example-Directed Program Synthesis". PLDI, 2015.


Gabriel Scherer and Didier Rémy. "Which simple types have a unique inhabitant?" ICFP, 2015.


Jonathan Frankle, Peter-Michael Osera, David Walker, and Steve Zdancewic. "Example-directed synthesis: a type-theoretic interpretation". POPL, 2016.


Nadia Polikarpova, Ivan Kuraj, and Armando Solar-Lezama. "Program synthesis from polymorphic refinement types". PLDI, 2016.


Gabriel Scherer. "Which types have a unique inhabitant? Focusing on pure program equivalence." PhD thesis. Université Paris-Diderot, 2016.


Gabriel Scherer and Amal Ahmed. "Search for Program Structure". SNAPL, 2017.

Section 5

Focusing

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

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

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

$$\frac{\Gamma \vdash A_1 \qquad \Gamma \vdash A_2}{\Gamma \vdash A_1 \times A_2}$$

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

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

$$\frac{}{\Gamma, 0 \vdash C}\ +$$

$$\frac{}{\Gamma \vdash 1}\ -$$

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

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

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

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

$$\frac{\Gamma \vdash A_1 \qquad \Gamma \vdash A_2}{\Gamma \vdash A_1 \times A_2}$$

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

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

$$\frac{}{\Gamma, 0 \vdash C}\ +$$

$$\frac{}{\Gamma \vdash 1}\ -$$

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

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

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

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:

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

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

Completeness:

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

Section 6

Focused λ-calculus

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-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:

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

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

β-short η-long:

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

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

What about sums?

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

Does not work:

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

$$
\begin{array}{l} \texttt{match}\ n\ \texttt{with} \\ \mid\ \sigma_1\ x \to \sigma_2\ x \\ \mid\ \sigma_2\ x \to \sigma_1\ x \end{array}
$$

Focusing to the rescue

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

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

Focusing to the rescue

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

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

Remark: "broken neutrals" are gone

$$
\pi_1 \left(\mathtt{match}\ x\ \mathtt{with} \;\middle|\; \begin{array}{l} \sigma_1\ y \to n_1 \\ \sigma_2\ y \to n_2 \end{array} \right)
$$

Completeness of focusing

Logic:

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

Completeness of focusing

Logic:

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

Programming:

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

Canonicity

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

Proof in Noam Zeilberger's thesis (2009), using ideas from ludics.

Section 7

Saturation

Nope !

Focusing is still not canonical – for pure languages.

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

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

Nope !

Focusing is still not canonical – for pure languages.

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

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

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

Nope !

Focusing is still not canonical – for pure languages.

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

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

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

Is $v \approx_{\beta\eta} w$? Pull the let-bindings to the roots and compare. Works for sums.

PhD topic strikes back

I wanted to **enumerate** the canonical inhabitants at a given type.

No existing term to start with.

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

Saturated focused λ-terms

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

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

- 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

Selection

Which $\bar{n}$ to split in a given context Γ?

"All of them" $\implies$ infinite set $(x : \mathbb{N} \to P \vdash \dots)$

Parameter: a **selection function** $\Phi(\Gamma)$ returning the (finite) $\bar{n}$.

For unicity: "at most two at each type"

Local completeness:

Selection

Which $\bar{n}$ to split in a given context Γ?

"All of them" $\implies$ infinite set $(x : \mathbb{N} \to P \vdash \dots)$

Parameter: a **selection function** $\Phi(\Gamma)$ returning the (finite) $\bar{n}$.

For unicity: "at most two at each type"

Local completeness:

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

Idea: $\Phi(\Gamma) \supseteq \{\Gamma \vdash_{\mathtt{foc}} n : P \mid n \in v\}$ suffices.
$\Phi \cup \Phi' \implies$ complete for finite sets of terms.

Canonicity

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

Thanks

Questions?