# Focusing for code inference, a tutorial

**Gabriel Scherer**, Didier Rémy

Northeastern University

May 31, 2016

# Context

A formal look at **code inference** (program synthesis).

$$\Gamma \vdash ? : A$$

We are searching among well-typed programs for a fixed $(\Gamma, A)$.

# Context

A formal look at **code inference** (program synthesis).

$$\Gamma \vdash ? : A$$

We are searching among well-typed programs for a fixed $(\Gamma, A)$.

Many of these programs are equivalent; considering duplicates is useless.

# Context

A formal look at **code inference** (program synthesis).

$$\Gamma \vdash ? : A$$

We are searching among well-typed programs for a fixed $(\Gamma, A)$.

Many of these programs are equivalent; considering duplicates is useless.

Question: is there a restricted judgment $\Gamma \vdash_{\mathsf{foc}} t : A$ that is

# Context

A formal look at **code inference** (program synthesis).

$$\Gamma \vdash ? : A$$

We are searching among well-typed programs for a fixed $(\Gamma, A)$.

Many of these programs are equivalent; considering duplicates is useless.

Question: is there a restricted judgment $\Gamma \vdash_{\text{foc}} t : A$ that is

- **complete**: For any $\Gamma \vdash t : A$, there is an equivalent $\Gamma \vdash_{\text{foc}} t' : A$.

# Context

A formal look at **code inference** (program synthesis).

$$\Gamma \vdash ? : A$$

We are searching among well-typed programs for a fixed $(\Gamma, A)$.

Many of these programs are equivalent; considering duplicates is useless.

Question: is there a restricted judgment $\Gamma \vdash_{\text{foc}} t : A$ that is

- **complete**: For any $\Gamma \vdash t : A$, there is an equivalent $\Gamma \vdash_{\text{foc}} t' : A$.
- **canonical**: If $\Gamma \vdash_{\text{foc}} t : A$ and $\Gamma \vdash_{\text{foc}} u : A$ and $t \neq u$, then $t$ and $u$ are not equivalent.

# Context

A formal look at **code inference** (program synthesis).

$$\Gamma \vdash ? : A$$

We are searching among well-typed programs for a fixed $(\Gamma, A)$.

Many of these programs are equivalent; considering duplicates is useless.

Question: is there a restricted judgment $\Gamma \vdash_{\text{foc}} t : A$ that is

- **complete**: For any $\Gamma \vdash t : A$, there is an equivalent $\Gamma \vdash_{\text{foc}} t' : A$.
- **canonical**: If $\Gamma \vdash_{\text{foc}} t : A$ and $\Gamma \vdash_{\text{foc}} u : A$ and $t \neq u$, then $t$ and $u$ are not equivalent.

?

# Context

A formal look at **code inference** (program synthesis).

$$\Gamma \vdash ? : A$$

We are searching among well-typed programs for a fixed $(\Gamma, A)$.

Many of these programs are equivalent; considering duplicates is useless.

Question: is there a restricted judgment $\Gamma \vdash_{\text{foc}} t : A$ that is

- **complete**: For any $\Gamma \vdash t : A$, there is an equivalent $\Gamma \vdash_{\text{foc}} t' : A$.
- **canonical**: If $\Gamma \vdash_{\text{foc}} t : A$ and $\Gamma \vdash_{\text{foc}} u : A$ and $t \neq u$, then $t$ and $u$ are not equivalent.

?

**Focusing**: not the complete answer (not canonical), but a good step forward.

# Simply-typed lambda-calculus

$$\frac{}{\Gamma, x : A \vdash x : A}$$

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

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

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

(plus units 0 and 1)

# $\lambda \Longrightarrow$ sequents

$$\frac{\Gamma \vdash t : A_1 \times A_2}{\Gamma \vdash \pi_1\ t : A_1} \quad \Rightarrow \quad \frac{\Gamma \vdash A_1 \times A_2}{\Gamma \vdash A_1} \quad \Rightarrow \quad \frac{\Gamma, A_1 \vdash C}{\Gamma, A_1 \times A_2 \vdash C}$$

$(,)$ is **non**-disjoint union

# Sequent calculus

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

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

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

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

# Invertible phase

$$\dfrac{\dfrac{?}{X + Y \vdash X}}{X + Y \vdash Y + X}$$

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

$$\dfrac{\dfrac{?}{X + Y \vdash X}}{X + Y \vdash Y + X}$$

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 $(X \to Y) \to (X \to Y)$ instead of two ($\lambda f.\, f$ and $\lambda f.\, \lambda x.\, f\ x$).

After all invertible rules, negative context, positive goal.

# Non-invertible phases

After all invertible rules, negative context, positive goal.

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

# Non-invertible phases

After all invertible rules, negative context, positive goal.

Only step forward: select a formula, apply some non-invertible rules 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 rules 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:

$$\dfrac{\dfrac{\dfrac{X_2, \qquad Y_1 \vdash A}{X_2 \times X_3, \qquad Y_1 \vdash A}}{X_2 \times X_3, \quad Y_1 \times Y_2 \vdash A}}{X_1 \times X_2 \times X_3, Y_1 \times Y_2 \vdash A}$$

# Demo Time

$$\frac{}{\vdash (1 \to X \to (Y + Z)) \to X \to (Y \to W) \to (Z + W)}$$

invertible rules

# Demo Time

$$\frac{}{(1 \to X \to (Y + Z)) \vdash X \to (Y \to W) \to (Z + W)}$$

invertible rules

# Demo Time

$$\frac{}{(1 \to X \to (Y + Z)), \quad X \vdash (Y \to W) \to (Z + W)}$$

invertible rules

# Demo Time

$$\frac{}{(1 \to X \to (Y + Z)), \quad X, \quad Y \to W \vdash Z + W}$$

invertible rules

# Demo Time

$$\frac{}{(1 \to X \to (Y + Z)), \quad X, \quad Y \to W \vdash Z + W}$$

choice of focus

# Demo Time

$$\frac{}{(1 \to X \to (Y + Z)), \quad X, \quad Y \to W \vdash Z + W}$$

choice of focus

# Demo Time

$$\frac{}{(1 \to X \to (Y + Z)), \quad X, \quad Y \to W \vdash Z + W}$$

non-invertible rules

# Demo Time

$$\frac{}{(1 \to X \to (Y + Z)),\quad X,\quad Y \to W \vdash Z + W}$$

non-invertible rules

# Demo Time

$$\frac{}{(1 \to X \to (Y + Z)), \quad X, \quad Y \to W \vdash Z + W}$$

invertible rules

# Demo Time

$$\frac{Y, Y \to W \vdash Z + W \qquad Z \vdash Z + W}{(1 \to X \to (Y + Z)), \quad X, \quad Y \to W \vdash Z + W}$$

invertible rules

# Demo Time

$$\frac{Y, Y \to W \vdash Z + W \qquad Z \vdash Z + W}{(1 \to X \to (Y + Z)),\quad X,\quad Y \to W \vdash Z + W}$$

choice of focus

# Demo Time

$$\frac{Y, Y \to W \vdash Z + W \qquad Z \vdash Z + W}{(1 \to X \to (Y + Z)),\quad X,\quad Y \to W \vdash Z + W}$$

conclusion

# This was focusing

Focused proofs are structured in alternating phases,
invertible (boring) and non-invertible (focus).

Phases are forced to be as long as possible – to eliminate duplicate proofs.

The idea is independent from the proof system.
Applies to sequent calculus or natural deduction;
intuitionistic, classical, linear, you-name-it logic.

# Focused normal forms for $\lambda$-calculus

(Grammar with type annotations)

$$
\begin{aligned}
v ::=\ & \text{values} \\
\mid\ & \lambda x.\, v \\
\mid\ & (v_1, v_2) \\
\mid\ & \texttt{match}\ x\ \texttt{with} \left| \begin{array}{l} \sigma_1\ x \to v_1 \\ \sigma_2\ x \to v_2 \end{array} \right. \\
\mid\ & (f : P \mid X)
\end{aligned}
$$

$$
\begin{aligned}
f ::=\ & \text{focused forms} \\
\mid\ & \texttt{let}\ (x : P) = n\ \texttt{in}\ v \\
\mid\ & (n : X^-) \\
\mid\ & (p : P)
\end{aligned}
$$

$$
\begin{aligned}
n ::=\ & \text{negative neutrals} \\
\mid\ & (x : N) \\
\mid\ & \pi_i\ n \\
\mid\ & n\ p
\end{aligned}
$$

$$
\begin{aligned}
p ::=\ & \text{positive neutrals} \\
\mid\ & (x : X^+) \\
\mid\ & \sigma_i\ p \\
\mid\ & (v : N)
\end{aligned}
$$