Strong rule induction for syntax with bindings

Jan van Brügge (Heriot-Watt University)
James McKinna (Heriot-Watt University)
Andrei Popescu (University of Sheffield)
Dmitriy Traytel (University of Copenhagen)

Foreword: Mechanizing binders

Nameless (de-Bruijn)

- Easy to set up
- Needs dependent types for usability
- Bleeds into proofs
- Hard to define complex binders

Named (Nominal)

- Lot of work to set up
- Gets out of the way afterwards
- Complex binders do not need heavy encoding

The problem

$$\frac{}{(\lambda x.\ M)\ N \to M[N/x]}$$

$$\frac{M \to M'}{M\ N \to M'\ N}$$

$$\frac{N \to N'}{M\ N \to M\ N'}$$

$$\frac{M \to M'}{\lambda x.\ M \to \lambda x.\ M'}$$

Beta reduction is the **smallest** relation **closed** under these rule

fixpoint

Inductive definition as least fixpoints

Let **(L, ≤)** be a **complete lattice** and let **f : L → L** be an order-preserving (**monotonic**) function w.r.t. ≤. Then the set of fixed points of f in L forms a complete lattice under ≤.

Under the hood

$$
\begin{aligned}
&\text{step} = \text{lfp } (\lambda R\ t1\ t2. \\
&\quad (\exists x\ M\ N. \quad t1 = (\lambda x.\ M)\ N \land t2 = M[N/x]) \\
&\lor (\exists M\ M'\ N. \quad (R\ M\ M') \land t1 = M\ N \land t2 = M'\ N) \\
&\lor (\exists N\ N'\ M. \quad (R\ N\ N') \land t1 = M\ N \land t2 = M\ N') \\
&\lor (\exists x\ M\ M'. \quad (R\ M\ M') \land t1 = (\lambda x.\ M) \land t2 = (\lambda x.\ M')) \\
&)
\end{aligned}
$$

$$\frac{}{(\lambda x.\ M)\ N \to M[N/x]}$$

$$\frac{M \to M'}{M\ N \to M'\ N} \qquad \frac{N \to N'}{M\ N \to M\ N'} \qquad \frac{M \to M'}{\lambda x.\ M \to \lambda x.\ M'}$$

Throwing binders into the mix

$G = \lambda R\ B\ t1\ t2.$

$(\exists x\ M\ N.\ B = \{x\} \wedge t1 = (\lambda x.\ M)\ N \wedge t2 = M[N/x])$

$\vee (\exists M\ M'\ N.\ B = \{\} \wedge (R\ M\ M') \wedge t1 = M\ N \wedge t2 = M'\ N)$

$\vee (\exists N\ N'\ M.\ B = \{\} \wedge (R\ N\ N') \wedge t1 = M\ N \wedge t2 = M\ N')$

$\vee (\exists x\ M\ M'.\ B = \{x\} \wedge (R\ M\ M') \wedge t1 = (\lambda x.\ M) \wedge t2 = (\lambda x.\ M'))$

Obviously still monotonic

Equivariance & Refreshability

The relation is equivariant if:

$$G\ R\ B\ t1\ t2 \Longrightarrow$$

$$G\ (\lambda x1\ x2.\ R\ (\pi^{-1} \cdot x1)\ (\pi^{-1} \cdot x2))\ (\pi \cdot B)\ (\pi \cdot t1)\ (\pi \cdot t2)$$

The relation is refreshable if:

$$G\ R\ B\ t1\ t2 \Longrightarrow$$

$$\exists B'.\ B' \cap (\text{supp}\ t1 \cup \text{supp}\ t2) = \{\} \wedge G\ R\ B'\ t1\ t2$$

What are the advantages

- **Independent of the format of the rules!**
  - E.g. supports higher order relations, quantifiers etc
  - Works on other fixpoints
- **No extra (freshness) side conditions in the rules**
  - Freshness is the **output** of the strengthening
  - No need to prove equality of the relation with and without extra side conditions
- **Automation (somewhat WIP)**

Demo

Code at https://github.com/jvanbruegge/binder_datatypes

More in the paper

- **Generalizations for**
  - Using inductive information for refreshability
  - Infinite (co-)datatypes
  - Non-Equivariant relations
- **Case studies**
  - (Parallel-)Beta Reduction of Untyped Lambda Calculus
  - Transitivity of subtyping of System Fsub (POPLmark 1A)
  - Reduction in the Process Calculus
  - Mazza's Infinitary Lambda Calculus

Summary

- **Inductive definitions are least fixpoints**
- **If the rules defining the relation are monotonic, equivariant and refreshable we can derive a strong induction theorem**
- **It can be automated**