Higher-Order Abstract Syntax in Isabelle/HOL

Douglas J. Howe

Carleton University

July 13, 2010

Problem: representing data with binding operations

Running example: expressions of the untyped λ-calculus.

First-order representation:

$$\ulcorner \lambda x.\, x\, x \urcorner \;=\; Lam\ \text{“}x\text{”}\ (App\ (Var\ \text{“}x\text{”})\ (Var\ \text{“}x\text{”}))$$

Higher-order representation:

$$\ulcorner \lambda x.\, x\, x \urcorner \;=\; Lam\,(\lambda x.\, App\ x\ x)$$

Recursive datatype view

First-order representation:

$$exp \;=\; Var\, string \mid App\, exp\, exp \mid Lam\, string\, exp$$

Higher-order representation:

$$exp \;=\; App\, exp\, exp \mid Lam\,(exp \rightarrow exp)$$

Negative occurrences: use *parametric* functions

$$exp \;=\; App\,exp\,exp \;\mid\; Lam\,(exp \rightarrow exp)$$

$$\Downarrow$$

$$exp \;=\; App\,exp\,exp \;\mid\; Lam\,(exp \hookrightarrow exp)$$

$A \hookrightarrow B$ is the type of functions in $A \rightarrow B$ that are *parametric*, i.e. they don't analyze, or discriminate on, their arguments.

- *Parametric:* $\lambda x.\; App\; x\; x$
- *Not:* $\lambda x.\; case\; x\; of\; App\,(u, u) \rightarrow True \;\mid\; Lam\,(f) \rightarrow False$

Representing open expressions (i.e. free variables)

Generally, we use a type of “contexts”:

$$\alpha \lhd \beta \ = \ \mathit{Closed}\ \ \beta \ \mid \mathit{Bind}\ \ \alpha \hookrightarrow \beta$$

Contexts are iterated abstractions, over α, of β objects.

Example:

$$\begin{aligned} \ulcorner \lambda x.\, xy \urcorner &\ :: \ \mathit{exp} \lhd \mathit{exp} \\ \ulcorner \lambda x.\, xy \urcorner &\ = \ \mathit{Bind}\,(\lambda y.\, \mathit{Closed}\,(\mathit{Lam}\,(\lambda x.\, \mathit{App}\,x\,y))) \end{aligned}$$

Lift *Lam*, *App* to $\mathit{exp} \lhd \mathit{exp}$: *Lamc*, *Appc*.

Well-founded ordering on contexts

Can define well-founded ordering $\prec$ on $\alpha \lhd \beta$ for general class of types β. For *exp*, have (for example):

$$\begin{array}{rcl} \mathit{Closed}\, u & \prec & \mathit{Closed}\,(\mathit{App}\, u\, v) \\ \mathit{Closed}\, u & \prec & \mathit{Closed}\,(\mathit{App}\, u\, v) \\ \mathit{Bind}\,(\lambda x.\, \mathit{Closed}\,(f\ x)) & \prec & \mathit{Closed}\,(\mathit{Lam}\, f) \end{array}$$

Ordering yields induction rule (quantifying over $\mathit{exp} \lhd \mathit{exp}$):

$$\begin{array}{l} \forall x.\ \mathit{is\_var}\, x \Rightarrow p\ x\ \ \& \\ \forall x\, y.\ p\ x \Rightarrow p\ y \Rightarrow p\ (\mathit{Appc}\, x\ y)\ \ \& \\ \forall x.\ p\ x \Rightarrow p\ (\mathit{Lamc}\, x) \\ \Rightarrow\ \ p\ x \end{array}$$

HOAS induction

Using context-induction, can prove the following conventional HOAS-style induction scheme (quantifiers are over *exp*).

$$\begin{array}{l} \forall x\, y.\ p\ x \Rightarrow p\ y \Rightarrow p\,(App\ x\ y)\ \ \& \\ \forall f.\ (\forall x.\ p\ x \Rightarrow p\ (f\ x)) \Rightarrow p\ (Lam\ f) \\ \Rightarrow p\ x \end{array}$$

Conclusion/Future

- Isabelle/HOL theory: general account of parametric functions and contexts; induction principles for *exp*.
- Main contribution: HOAS-style recursive type definitions in HOL, along with a way to represent open terms without explicit variable representation.
- Approach based on an extended set-theoretic semantics for HOL. Axiomatized for this project.
- Make the Isabelle/HOL theory definitional (axioms are evil (!)).
- Packagize?
- What's this representation good for?