Translating Extensionality in Polymorphic HOL

Thesis B Seminar

Vincent Jackson (z5060345)

School of Computer Science and Engineering, UNSW Sydney

November 6, 2019

Aim and Overview

Aim: Create a translation from extensional polymorphic Higher-Order Logic to intensional polymorphic Higher Order Logic.

Overview:

1. What is a translation?
2. Finding a consistent polymorphic HOL.
3. My progress so far.

What is a Translation?

Translations:

- Take formulas from one logic to a different logic
- The translation preserves validity
- The translation is non-trivial (if the translated formula is provable, everything in its image is provable)

In symbols:

$$(-)^\bullet : \mathrm{Fml}_S \Rrightarrow \mathrm{Fml}_T \qquad \text{(Translation)}$$

$$\Gamma \vdash_S s \Longrightarrow \Gamma^\bullet \vdash_T s^\bullet \qquad \text{(Preserves Validity)}$$

$$\Gamma^\bullet \vdash_T s^\bullet \Longrightarrow \Gamma \vdash_S s \qquad \text{(Not Trivial)}$$

Finding a Consistent Polymorphic HOL

Problem: Girard's Paradox means that naïvely adding type polymorphism renders the logic inconsistent [Geu07].

Solution: Use HOL2P [Völ07], a consistent higher-order logic with type polymorphism, based on HOL Light [Har09].

HOL Light

Types: o (propositions), ι (individuals), $\tau_1 \Rightarrow \tau_2$ (functions), α (type variables)

Terms: x (variables), $\lambda x.\, t$ (abstraction), $f\, s$ (application), and constants (e.g. $=$ and ε).

Rules:

refl, trans, mk-comb, abs, beta,
assm, eq-mp, deduct-antisym-rule,
inst, ty-inst,
eta-ax, select-ax, infinity-ax

HOL Light

$$\frac{}{\Gamma \vdash s = s}\ \text{refl} \qquad \frac{\Gamma \vdash s = t \quad \Delta \vdash t = u}{\Gamma, \Delta \vdash s = u}\ \text{trans}$$

$$\frac{\Gamma \vdash s = t \quad \Delta \vdash u = v}{\Gamma, \Delta \vdash s\ u = t\ v}\ \text{mk-comb}$$

$$(x \text{ not in } \Gamma)\ \frac{\Gamma \vdash s = t}{\Gamma \vdash \lambda x.\, s = \lambda x.\, t}\ \text{abs} \qquad \frac{}{\vdash (\lambda x.\, s)\ y = s[y/x]}\ \text{beta}$$

$$\frac{}{p \vdash p}\ \text{assm}$$

$$\frac{\Gamma \vdash s =_o t \quad \Delta \vdash s}{\Gamma, \Delta \vdash t}\ \text{eq-mp} \qquad \frac{\Gamma \vdash p \quad \Delta \vdash q}{\Gamma - q, \Delta - p \vdash p =_o q}\ \text{deduct-antisym-rule}$$

$$\frac{\Gamma \vdash s}{\Gamma[t_1, \ldots, t_n/x_1, \ldots, x_n] \vdash s[t_1, \ldots, t_n/x_1, \ldots, x_n]}\ \text{inst}$$

$$\frac{\Gamma \vdash s}{\Gamma[\tau_1, \ldots, \tau_n/\alpha_1, \ldots, \alpha_n] \vdash s[\tau_1, \ldots, \tau_n/\alpha_1, \ldots, \alpha_n]}\ \text{ty-inst}$$

$$\frac{}{\Gamma \vdash (\lambda x.\, t\ x) = t}\ \text{eta-ax} \qquad \frac{}{\vdash p\ x \longrightarrow p\ (\varepsilon x.\ p\ x)}\ \text{select-ax}$$

$$\frac{}{\vdash \exists (f : \iota \Rightarrow \iota).\, \mathrm{inj}\ f \wedge \neg \mathrm{onto}\ f}\ \text{infinity-ax}$$

HOL Light

$$\frac{}{\Gamma \vdash s = s}\ \text{refl} \qquad \frac{\Gamma \vdash s = t \quad \Delta \vdash t = u}{\Gamma, \Delta \vdash s = u}\ \text{trans}$$

$$\frac{\Gamma \vdash s = t \quad \Delta \vdash u = v}{\Gamma, \Delta \vdash s\ u = t\ v}\ \text{mk-comb}$$

$$(x \text{ not in } \Gamma)\ \frac{\Gamma \vdash s = t}{\Gamma \vdash \lambda x.\, s = \lambda x.\, t}\ \text{abs} \qquad \frac{}{\vdash (\lambda x.\, s)\ y = s[y/x]}\ \text{beta}$$

$$\frac{}{p \vdash p}\ \text{assm}$$

$$\frac{\Gamma \vdash s =_o t \quad \Delta \vdash s}{\Gamma, \Delta \vdash t}\ \text{eq-mp} \qquad \frac{\Gamma \vdash p \quad \Delta \vdash q}{\Gamma - q, \Delta - p \vdash p =_o q}\ \text{deduct-antisym-rule}$$

$$\frac{\Gamma \vdash s}{\Gamma[t_1, \ldots, t_n/x_1, \ldots, x_n] \vdash s[t_1, \ldots, t_n/x_1, \ldots, x_n]}\ \text{inst}$$

$$\frac{\Gamma \vdash s}{\Gamma[\tau_1, \ldots, \tau_n/\alpha_1, \ldots, \alpha_n] \vdash s[\tau_1, \ldots, \tau_n/\alpha_1, \ldots, \alpha_n]}\ \text{ty-inst}$$

$$\frac{}{\Gamma \vdash (\lambda x.\, t\ x) = t}\ \text{eta-ax} \qquad \frac{}{\vdash p\ x \longrightarrow p\ (\varepsilon x.\ p\ x)}\ \text{select-ax}$$

$$\frac{}{\vdash \exists (f : \iota \Rightarrow \iota).\, \mathrm{inj}\ f \wedge \neg \mathrm{onto}\ f}\ \text{infinity-ax}$$

Functional Extensionality

Functional extensionality: abs (ξ) and eta-ex (η) [BBK04]

$$(x \text{ not in } \Gamma)\ \frac{\Gamma \vdash s = t}{\Gamma \vdash \lambda x.\, s = \lambda x t}\ \text{abs} \qquad \frac{}{\vdash \lambda x.\, s\, x = s}\ \text{eta-ax}$$

$$\dfrac{\dfrac{\dfrac{}{\vdash (=) = (=)}\ \text{refl} \quad \dfrac{}{\vdash (\lambda x.\, f\, x) = f}\ \text{eta-ax} \quad \dfrac{}{\vdash (\lambda x.\, g\, x) = g}\ \text{eta-ax}}{\vdash ((\lambda x.\, f\, x) = (\lambda x.\, g\, x)) =_o (f = g)}\ \text{mk-comb*} \quad \dfrac{\Gamma \vdash f\, x = g\, x}{\Gamma \vdash (\lambda x.\, f\, x) = (\lambda x.\, g\, x)}\ \text{abs}}{\Gamma \vdash f = g}\ \text{eq-mp}$$

HOL2P

Extra Types: $\Pi\alpha.\, \tau$ (type polymorphism)

Extra Terms: $\Lambda\alpha.\, s$ (type abstraction), $s\, [:\tau:]$ (type application)

Extra Rules:

tyapp, tyabs, tybeta

$$\frac{\Gamma \vdash s = t}{\Gamma s\, [:\tau:] = t\, [:\tau:]}\ \text{tyapp}$$

$$\frac{\Gamma \vdash s = t}{\Gamma \vdash (\Lambda\alpha.\, s) = (\Lambda\alpha.\, t)}\ \text{tyabs} \qquad \frac{}{\vdash (\Lambda\alpha.\, s)\ \tau = s[\tau/\alpha]}\ \text{tybeta}$$

How HOL2P deals with impredicativity: ranks.

In HOL2P, types have ranks: $\tau :_{\mathfrak{r}} r$.

Two ranks: large ($\mathfrak{l}$) and small ($\mathfrak{s}$)

Type polymorphism quantifies over a small type, and produces a large type:

$$\big(\Pi(\alpha :_{\mathfrak{r}} \mathfrak{s}).\, \tau\big) :_{\mathfrak{r}} \mathfrak{l}$$

This means we have a well founded induction principle for type polymorphism.

Translation of Terms

The translation of terms:

$$
\begin{aligned}
(-)^\bullet &: \mathrm{Fml}_S \Rrightarrow \mathrm{Fml}_T \\
x^\bullet &\triangleq x \\
(\lambda x.\, s)^\bullet &\triangleq \lambda x.\, s^\bullet \\
(s\ t)^\bullet &\triangleq s^\bullet\ t^\bullet \\
(\Lambda\alpha.\, s)^\bullet &\triangleq \Lambda\alpha.\, s^\bullet \\
(s\ [\!:\tau:\!])^\bullet &\triangleq s^\bullet\ [\!:\tau:\!] \\
(s = t)^\bullet &\triangleq s^\bullet \doteq t^\bullet
\end{aligned}
$$

Pseudo-Extensional Equality

Pseudo-extensional equality is defined by recursion on β:

$$
\begin{aligned}
(\doteq_\beta) &: \beta \Rightarrow \beta \Rightarrow o \\
s \doteq_o t &\triangleq s =_o t \\
s \doteq_\iota t &\triangleq s =_\iota t \\
f \doteq_{\tau_1 \Rightarrow \tau_2} g &\triangleq \forall x\, y.\, x \doteq_{\tau_1} y \longrightarrow f\, x \doteq_{\tau_2} g\, y \\
s \doteq_{\Pi\alpha.\,\tau} t &\triangleq \underset{\mathsf{ty}}{\forall} \alpha.\, \mathcal{E}(\alpha) \longrightarrow s\,[:\alpha:] \doteq_\tau t\,[:\alpha:] \\
s \doteq_\alpha t & \qquad\qquad\qquad\qquad\qquad\qquad \text{(stuck)}
\end{aligned}
$$

Where $\underset{\mathsf{ty}}{\forall} \alpha.\, p$ is $(\Lambda\alpha.\, p) = (\Lambda\alpha.\, \mathrm{True})$.

Translation of Type Variables

$\mathcal{E}(\alpha)$ is the assumptions:

$$\forall(s, t, u : \alpha).\, s \doteq t \longrightarrow t \doteq u \longrightarrow s \doteq u$$

and

$$\forall(s : \alpha).\, s \doteq s$$

and

$$\forall(s : \alpha).\, \big((\lambda x.\, s\, x) \doteq s\big)$$

Overloaded to terms: $\mathcal{E}(s)$ is the union of all $\mathcal{E}(\alpha)$ for free type variables α in s.

Translation

$$\Gamma \vdash s$$

is translated to

$$\mathcal{E}(\Gamma), \mathcal{E}(s), \mathcal{R}(\Gamma), \mathcal{R}(s), \Gamma^\bullet \vdash s^\bullet$$

$\mathcal{R}(s)$ is a set of formulas $x \doteq x$ for each free-var in s.

Progress

refl, trans, mk-comb, abs, beta,
assm, eq-mp, deduct-antisym-rule,
inst, ty-inst
tyapp, tyabs, tybeta
eta-ax, select-ax, infinity-ax

Progress

Rules which don't contain equality:

refl, trans, mk-comb, abs, beta,
~~assm~~, eq-mp, deduct-antisym-rule,
inst, ty-inst,
tyapp, tyabs, tybeta
eta-ax, select-ax, infinity-ax

Progress

Rules which only use propositional equality:

refl, trans, mk-comb, abs, beta,
~~assm~~, ~~eq-mp~~, ~~deduct-antisym-rule~~,
inst, ty-inst,
tyapp, tyabs, tybeta
eta-ax, select-ax, infinity-ax

Progress

I have proofs* of correct translation for:

~~refl~~, ~~trans~~, ~~mk-comb~~, ~~abs~~, ~~beta~~,
~~assm~~, ~~eq-mp~~, ~~deduct-antisym-rule~~,
inst, ty-inst,
tyapp, tyabs, tybeta,
~~eta-ax~~, select-ax, infinity-ax

* caveat: assuming some lemmas hold

Progress

Lemmas that I assume hold (yet to be proven):

If an object is 'extensionally equatable', it is in the extensional domain.

$$\frac{}{x \doteq_\tau y \vdash x \doteq_\tau x} \qquad \frac{}{x \doteq_\tau y \vdash y \doteq_\tau y}$$

(similar to a lemma from [Riz09])

$$(x \text{ not in } \Gamma \text{ or } p)\ \frac{\Gamma, \mathcal{R}(x) \vdash p}{\Gamma \vdash p}$$

(also similar to a lemma from [Riz09])

Substitution of extensional variables under extensional equality:

$$\frac{}{u \doteq_{\tau_1} v \vdash t[u/x] \doteq_{\tau_2} s[v/x]}$$

Plan

Thesis B
- Extend the translation to the polymorphic HOL
- Give the Thesis B Presentation
- Write the Thesis B Report

Thesis C
- Finish the proofs of the extended translation
- Give the Thesis Presentation
- Write the Thesis Report
- Extension: Formalise the translation in a theorem prover

Christoph Benzmüller, Chad E. Brown, and Michael Kohlhase.
Higher-order semantics and extensionality.
Journal of Symbolic Logic, 69(4):1027–1088, 2004.

Herman Geuvers.
(In)consistency of extensions of higher order logic and type theory.
In Thorsten Altenkirch and Conor McBride, editors, *Types for Proofs and Programs*, pages 140–159, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

John Harrison.
Hol light: An overview.
In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *Theorem Proving in Higher Order Logics*, pages 60–66, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

Christine Rizkallah.
Proof representations for higher order logic.
Master's thesis, Universität des Saarlandes, Saarbrücken, Germany, 2009.

Norbert Völker.
Hol2p - a system of classical higher order logic with second order polymorphism.
In Klaus Schneider and Jens Brandt, editors, *Theorem Proving in Higher Order Logics*, pages 334–351, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.