universität
innsbruck

Translation-respecting Semantics for Dependently-Typed Higher-Order Logic

Daniel Ranalter

Overview

- Motivation
- Higher-Order Logic
- Henkin Semantics
- *Dependently-Typed* Higher-Order Logic
- DHOL Semantics
- Conclusion

Motivation

Simple Type Theory/Higher-Order Logic (HOL)

- theoretic: Type Theory is used as mathematical foundation, created in response to the foundational crisis
- practical: also used as a model of computation (Functional Programming)

Motivation

Simple Type Theory/Higher-Order Logic (HOL)

- theoretic: Type Theory is used as mathematical foundation, created in response to the foundational crisis
- practical: also used as a model of computation (Functional Programming)

Dependent Type Theory/Dependently-Typed Higher-Order Logic (DHOL)

- theoretic: allows to express mathematical concepts like finite, fixed-size sets
- practical: allows to incorporate guards into the level of types (eg. unfailing head function)

Syntax

HOL Syntax

- This is only one presentation of HOL
- Simple Type Theory a la Church with a base-type for booleans, implication and equality

$$
\begin{array}{lcll}
T & ::= & \circ \mid T, a\ tp \mid T, c : A \mid T, F & \text{theory} \\
\Gamma & ::= & \bullet \mid \Gamma, x : A \mid \Gamma, F & \text{context} \\
A, B & ::= & a \mid o \mid A \to B & \text{types} \\
t, u, v & ::= & x \mid \lambda x : A.t \mid tu \mid t \Rightarrow u \mid t =_A u \mid \bot & \text{terms}
\end{array}
$$

- Con- and Disjunction, Quantification, etc. can be encoded
- $\forall f : nat \to nat \to nat.((\lambda n : nat.f\ 0\ n) =_{nat \to nat} f\ 0)$

Judgements

What can we do with it?

- $\forall f : nat \to nat \to nat.((\lambda n : nat.f\ 0\ n) =_{nat \to nat} f\ 0)$?
- How to reason about statements?
- Judgements:

$\Gamma \vdash t$	Well-formed boolean term t is provable
$\Gamma \vdash t : A$	Term t is of (well-formed) type A
$\Gamma \vdash A \equiv B$	Well-formed types A and B are equal
$\Gamma \vdash A\ tp$	Type A is well-formed

Judgements

What can we do with it?

- $\forall f : nat \to nat \to nat.((\lambda n : nat.f\ 0\ n) =_{nat \to nat} f\ 0)$?
- Syntax has no meaning
- We give meaning by Judgements:

$\Gamma \vdash t$	Well-formed boolean term t is provable
$\Gamma \vdash t : A$	Term t is of (well-formed) type A
$\Gamma \vdash A \equiv B$	Well-formed types A and B are equal
$\Gamma \vdash A\ tp$	Type A is well-formed

The missing piece

But how do we arrive at a judgement?

Rules

Some Natural Deduction Rules

$$\frac{\Gamma \vdash s : o \quad \Gamma \vdash t : o}{\Gamma \vdash (s \Rightarrow t) : o} \Rightarrow\text{Type} \qquad \frac{\Gamma \vdash s : o \quad \Gamma, s \vdash t}{\Gamma \vdash s \Rightarrow t} \Rightarrow$$

$$\frac{\Gamma \vdash A \equiv A' \quad \Gamma \vdash B \equiv B'}{\Gamma \vdash A \to B \equiv A' \to B'} \to\text{Cong} \qquad \frac{\Gamma \vdash A\ tp}{\Gamma \vdash A \equiv A} \text{tpRefl}$$

$$\frac{a\ tp \ \in\ T}{\Gamma \vdash a\ tp} \text{tp}$$

Example

Natural Numbers - Theory

types	constants/functions	axioms
$nat\ tp$	$0 : nat$	$\forall n, m : nat.(plus\ (suc\ m)\ n =_{nat} plus\ m\ (suc\ n))$
	$suc : nat \to nat$	$\forall n : nat.(plus\ 0\ n =_{nat} n)$
	$plus : nat \to nat \to nat$	

Natural Numbers - Judgements

- $\Gamma \vdash \forall i, j, k : nat.(plus\ i\ (plus\ j\ k) =_{nat} plus\ (plus\ i\ j) k)$
- $\Gamma \vdash suc\ (plus\ 0\ (suc\ 0)) : nat$

Semantics

Standard Models - informal

A Standard Model is a tuple $(D, [\![\bullet]\!])$ where the class $\{D_\alpha\}$ consists of

- D_ι — a set of arbitrary elements for each base type
- $D_o = \{T, F\}$
- $D_{\beta\gamma} = \{f \mid f : D_\beta \mapsto D_\gamma\}$ for all β, γ

Semantics

Standard Models - informal

A Standard Model is a tuple $(D, [\![\bullet]\!])$ where the class $\{D_\alpha\}$ consists of

- D_ι — a set of arbitrary elements for each base type
- $D_o = \{T, F\}$
- $D_{\beta\gamma} = \{f \mid f : D_\beta \mapsto D_\gamma\}$ for all β, γ

and the interpretation function $[\![\bullet]\!]$ maps

- contexts to sets of variable assignments, s.t. any axioms evaluate to $\top$,
- terms of a type to elements of the corresponding set,
- boolean connectives to their standard interpretation,
- lambda abstractions to functions, and
- applications to function calls on their arguments.

Semantics

Model for our formulation of Natural Numbers

Continuing our previous example of the natural numbers a possible model would be

- $D_{nat} = \{0_{\mathbb{N}}, 1_{\mathbb{N}}, 2_{\mathbb{N}}, \ldots\}$
- $[\![0]\!] = 0_{\mathbb{N}}$
- $[\![suc]\!] = 1_{\mathbb{N}} +$
- $[\![plus]\!] = +$

It is easy to see that $+$ satisfies the definitional axioms of *plus*, making this a valid model of our theory.

Semantics

However...

This would now allow us to model arithmetic — due to Gödel it has to be incomplete!

Semantics

However...

This would now allow us to model arithmetic — due to Gödel it has to be incomplete!

Henkin Models/General Models

To get sound *and* complete models, we follow Henkin. In order to regain completeness, we restrict the domain of functions:

- D_ι — a set of arbitrary elements for each base type
- $D_o = \{T, F\}$
- $D_{\beta\gamma} = \{f \mid f : D_\beta \mapsto D_\gamma\}$ for all β, γ

Semantics

However...

This would now allow us to model arithmetic — due to Gödel it has to be incomplete!

Henkin Models/General Models

To get sound *and* complete models, we follow Henkin. In order to regain completeness, we restrict the domain of functions:

- D_ι — a set of arbitrary elements for each type α
- $D_o = \{T, F\}$
- $D_{\beta\gamma} \subseteq \{f \mid f : D_\beta \mapsto D_\gamma\}$ for all β, γ

Extensions

DHOL Syntax

Now we can extend HOL to dependent types by replacing every occurrence of type-formation...

$$
\begin{array}{llll}
T & ::= & \circ \mid T, a\ tp \mid T, x : A \mid T, F & \text{theory} \\
\Gamma & ::= & \bullet \mid \Gamma, x : A \mid \Gamma, F & \text{context} \\
A, B & ::= & a \mid o \mid A \to B & \text{types} \\
t, u, v & ::= & x \mid \lambda x : A.t \mid tu \mid t \Rightarrow u \mid t =_A u \mid \bot & \text{terms}
\end{array}
$$

Extensions

DHOL Syntax

Now we can extend HOL to dependent types by replacing every occurrence of type-formation...

$$
\begin{array}{llll}
T & ::= & \circ \mid T, a : (\Pi x : A.)^* \, tp \mid T, x : A \mid T, F & \text{theory} \\
\Gamma & ::= & \bullet \mid \Gamma, x : A \mid \Gamma, F & \text{context} \\
A, B & ::= & a t_1 \ldots t_n \mid o \mid \Pi x : A.B & \text{types} \\
t, u, v & ::= & x \mid \lambda x : A.t \mid tu \mid t \Rightarrow u \mid t =_A u \mid \bot & \text{terms}
\end{array}
$$

... with the more general, dependent variant

Rules

HOL-ND to DHOL-ND

$$\frac{\Gamma \vdash s\colon o \quad \Gamma \vdash t\colon o}{\Gamma \vdash (s \Rightarrow t)\colon o} \Rightarrow\text{Type} \qquad \frac{\Gamma \vdash s\colon o \quad \Gamma, s \vdash t}{\Gamma \vdash s \Rightarrow t} \Rightarrow$$

$$\frac{\Gamma \vdash A \equiv A' \quad \Gamma \vdash B \equiv B'}{\Gamma \vdash A \to B \equiv A' \to B'} \to\text{Cong} \qquad \frac{\Gamma \vdash A\ tp}{\Gamma \vdash A \equiv A}\text{tpRefl}$$

Rules

HOL-ND to DHOL-ND

$$\frac{\Gamma \vdash s \colon o \quad \Gamma, s \vdash t \colon o}{\Gamma \vdash (s \Rightarrow t) \colon o} \Rightarrow\text{Type} \qquad \frac{\Gamma \vdash s \colon o \quad \Gamma, s \vdash t}{\Gamma \vdash s \Rightarrow t} \Rightarrow$$

$$\frac{\Gamma \vdash A \equiv A' \quad \Gamma \vdash B \equiv B'}{\Gamma \vdash A \rightarrow B \equiv A' \rightarrow B'} \rightarrow\text{Cong} \qquad \frac{\Gamma \vdash A\ tp}{\Gamma \vdash A \equiv A} \text{tpRefl}$$

Rules

HOL-ND to DHOL-ND

$$\frac{\Gamma \vdash s : o \quad \Gamma, s \vdash t : o}{\Gamma \vdash (s \Rightarrow t) : o} \Rightarrow \mathrm{Type} \qquad \frac{\Gamma \vdash s : o \quad \Gamma, s \vdash t}{\Gamma \vdash s \Rightarrow t} \Rightarrow$$

$$\frac{\Gamma \vdash A \equiv A' \quad \Gamma, x : A \vdash B \equiv B'}{\Gamma \vdash \Pi x : A.B \equiv \Pi x' : A'.B'} \Pi\mathrm{Cong} \qquad \frac{\Gamma \vdash A\ tp}{\Gamma \vdash A \equiv A} \mathrm{tpRefl}$$

Rules

HOL-ND to DHOL-ND

$$\frac{\Gamma \vdash s : o \quad \Gamma, s \vdash t : o}{\Gamma \vdash (s \Rightarrow t) : o} \Rightarrow\text{Type} \qquad \frac{\Gamma \vdash s : o \quad \Gamma, s \vdash t}{\Gamma \vdash s \Rightarrow t} \Rightarrow$$

$$\frac{\Gamma \vdash A \equiv A' \quad \Gamma, x : A \vdash B \equiv B'}{\Gamma \vdash \Pi x : A.B \equiv \Pi x' : A'.B'} \Pi\text{Cong}$$

$$\frac{a : (\Pi x_1 : A_1,\ \ldots,\ \Pi x_n : A_n) \in \Gamma \quad \Gamma \vdash s_1 =_{A_1} t_1 \quad \ldots \quad \Gamma \vdash s_n =_{A_n[x_1/s_1,\ldots,x_{n-1}/s_{n-1}]} t_n}{\Gamma \vdash as_1 \ldots s_n \equiv at_1 \ldots t_n}$$

tpRefl

Example

Fixed Length Lists of Natural Numbers - Theory

types	constants/functions
$lst : \Pi n : nat\ tp$	$nil : lst\ 0$
	$cons : \Pi n : nat.nat \to lst\ n \to lst\ (suc\ n)$
	$app : \Pi n, m : nat.lst\ n \to lst\ m \to lst\ (plus\ n\ m)$

Fixed Length Lists of Natural Numbers - Judgements

- $\Gamma \vdash \forall n : nat.\forall x : lst\ n.(app\ 0\ n\ nil\ x =_{lst\ n} x)$

Erasure

Simplifying things by making them more complicated

- DHOL is currently barely supported
- To increase usability, an erasure from DHOL to HOL exists
- Basic idea: Capture information lost during erasure in a Partial Equivalence Relation (PER)

Erasure, abridged

$\overline{a : \Pi x_1 : A_1, \ldots, \Pi x_n : A_n\ tp} =$

- $a\ tp$
- $a^* : \overline{A_1} \to \ldots \to \overline{A_n} \to a \to a \to o$
- Set of Axioms establishing PER properties for a^*

$\overline{x : A} =$

- $x : \overline{A}$
- $A^*\, x\, x$

Erasure Example

Erasure, abridged

$\overline{a : \Pi x_1 : A_1, \ldots, \Pi x_n : A_n \, tp} =$

- $a \, tp$
- $a^* : \overline{A_1} \to \ldots \to \overline{A_n} \to a \to a \to o$
- Set of Axioms establishing PER properties for a^*

$\overline{x : A} =$

- $x : \overline{A}$
- $A^* \, x \, x$

Erasing the Fixed Length List of Natural Numbers

$\overline{lst : \Pi n : nat \, tp} =$

- $lst \, tp$
- $lst^* : nat \to lst \to lst \to o$
- $+$ axioms

$\overline{nil : lst \, 0} =$

- $nil : lst$
- $lst^* \, 0 \, nil \, nil$

Erasure Example

Erasure, abridged

$\overline{a : \Pi x_1 : A_1, \ldots, \Pi x_n : A_n \ tp} =$

- $a \ tp$
- $a^* : \overline{A_1} \to \ldots \to \overline{A_n} \to a \to a \to o$
- Set of Axioms establishing PER properties for a^*

$\overline{x : A} =$

- $x : \overline{A}$
- $A^* \, x \, x$

$\overline{\forall \, x : A.t} =$
$\forall x : \overline{A}.$
$A^* \, x \, x \Rightarrow \overline{t}$

Erasing the Fixed Length List of Natural Numbers

$\overline{lst : \Pi n : nat \ tp} =$

- $lst \ tp$
- $lst^* : nat \to lst \to lst \to o$
- $+$ axioms

$\overline{nil : lst \, 0} =$

- $nil : lst$
- $lst^* \, 0 \, nil \, nil$

$\overline{\forall x : lst \, 0.t} =$
$\forall x : lst.$
$lst^* \, 0 \, x \, x \Rightarrow \overline{t}$

Motivation

Current Situation

- DHOL's semantics currently only defined in terms of inference rules
- It would be desireable to have a model theory
- Depending on the goals, different models lend themselves to consideration

Motivation

Current Situation

- DHOL’s semantics currently only defined in terms of inference rules
- It would be desireable to have a model theory
- Depending on the goals, different models lend themselves to consideration

Our Goals

- DHOL is implemented in the automated theorem prover Lash
- Unclear whether it is sound to use erased and non-erased terms “interchangeably”
- We suspect it is!

Notation

Family of sets

There are a lot of different ways to express indexed families of sets.
We will write a family of sets A_i with indices in the set I as $\langle A_i \rangle_{i:I}$. Accessing the subsets of A will then be written as $(A)_i$.

Semantics of DHOL

DHOL General Models

Models are defined as previously.
The interesting case for the interpretation function is that of dependent types in the theory and their realisation:

$$[\![T, a : \Pi x_1 : A_1 \ldots \Pi x_n : A_n\ tp]\!] = [\![T]\!] \cup (\langle \ldots (\langle x_{a_n} \rangle_{a_n : A_n}) \ldots \rangle_{a_1})_{a_1 : A_1}$$
$$[\![a\ t_1\ \ldots\ t_n]\!] = (\ldots(([\![a]\!])_{[\![t_n]\!]}) \ldots)_{[\![t_1]\!]}$$

i.e. the set resulting of instantiating the index family $\langle a \rangle$ with $t_1, \ldots, t_n$

Open Challenges

What remains to be done?

- Soundness proofs seem to be straight-forward.
- Translation-preservation (i.e. “For every model M, iff $[\![\Gamma]\!] \models^{DHOL}_{[\![T]\!]} [\![F]\!]$ and $\Gamma \vdash^{DHOL}_{T} F$ then $[\![\overline{\Gamma}]\!] \models^{HOL}_{[\![\overline{T}]\!]} [\![\overline{F}]\!]$”) is some work but I am optimistic.
- However, conversations with colleagues suggest completeness proof might be a problem.

Conclusion

- Henkin semantics/General models are an established interpretation of HOL
- We want a HOL-compatible interpretation of DHOL so we can mix reasoning steps
- General DHOL models are our suggestion to achieve that
- Several open questions remain