Higher-order Logic: Foundations
Motivation

- Higher-order logic (HOL) is an expressive foundation for mathematics: analysis, algebra, . . .
  computer science: program correctness, hardware verification, . . .
- Reasoning in HOL is classical.
- Still important: modeling of problems (now in HOL).
- Still important: deriving relevant reasoning principles.
Motivation (2)

• HOL offers **safety through strength:**
  - small kernel of constants and axioms;
  - Safety via conservative (definitional) extensions.

• **Contrast with**
  - weak logics (e.g., propositional logic): can’t define much;
  - axiomatic extensions: can lead to inconsistency

Bertrand Russell once likened the advantages of postulation over definition to the advantages of theft over honest toil!
Alternatives to Isabelle/HOL

• We will use and focus on Isabelle/HOL.
• Could forgo the use of a meta-logic and employ alternatives, e.g., HOL system or PVS. Or use constructive alternatives such as Coq or Nuprl.
• Choice depends on culture and application.
Which Foundation?

- **Set theory** is often seen as the basis for mathematics.
  - Zermelo-Fraenkel, Bernays-Gödel, . . .
  - Set theories (both) distinguish between sets and classes.
  - Consistency maintained as some collections are “too big” to be sets, e.g., class of all sets is not a set. A class cannot belong to another class (let alone a set)!

- **HOL** as an alternative (Church 1940, Henkin 1950).
  - **Rationale:** one usually works with typed entities.
  - Isabelle/HOL also supports like polymorphism and type classes. HOL is weaker than ZF set theory, but for most applications this does not matter. If you prefer ML to Lisp, you will probably prefer HOL to ZF.
    —Larry Paulson

- Another alternative: category theory (Eilenberg, Mac Lane)
Meaning of “Higher Order”

1st-order: quantification over individuals (0th-order objects).

\[ \forall x, y. R(x, y) \rightarrow R(y, x) \]

2nd-order: quantification over predicates and functions.

\[ \text{false} \equiv \forall P. P \]
\[ P \land Q \equiv \forall R. (P \rightarrow Q \rightarrow R) \rightarrow R \]

3rd-order: quantify over variables whose arguments are predicates.

“higher order” \[\iff\] union of all finite orders
Basic HOL Syntax (1)

- **Types:**
  \[ \tau ::= \text{bool} \mid \text{ind} \mid \tau \Rightarrow \tau \]
  - \text{bool} and \text{ind} are also called \text{o} and \text{i} in literature [Chu40, And86]
  - Isabelle allows definitions of new type constructors, e.g., \text{list}(\text{bool})
  - Isabelle supports polymorphic type definitions, e.g., \text{list}(\alpha)

- **Terms:** (\mathcal{V} \text{ set of variables and } \mathcal{C} \text{ set of constants})
  \[ \mathcal{T} ::= \mathcal{V} \mid \mathcal{C} \mid (\mathcal{T}\mathcal{T}) \mid \lambda \mathcal{V}. \mathcal{T} \]
  - Terms are simply-typed.
  - Terms of type \text{bool} are called (well-formed) formulae.
Basic HOL Syntax (2)

• Constants are always supplied with types and include:

  $\text{True, False : bool}$

  $\_ = \_ : \tau \Rightarrow \tau \Rightarrow \text{bool}$ (for all types $\tau$)

  $\_ \rightarrow \_ : \text{bool} \Rightarrow \text{bool} \Rightarrow \text{bool}$

  $\nu_\_ : (\tau \Rightarrow \text{bool}) \Rightarrow \tau$ (for all types $\tau$)

• Note that the description operator $\nu f$ yields the unique element $x$ for which $f \, x$ is True, provided it exists. Otherwise, it yields an arbitrary value.

• Note that in Isabelle, the provisos “for all types $\tau$” can be expressed by using polymorphic type variables $\alpha$. 
HOL Semantics

- Intuitively an extension of many-sorted semantics with functions
  - FOL: structure is domain and functions/relations
    \[ \langle D, (f_i)_{i \in F}, (r_i)_{i \in R} \rangle \]
  - Many-sorted FOL: domains are sort-indexed
    \[ \langle (D_i)_{i \in S}, (f_i)_{i \in F}, (r_i)_{i \in R} \rangle \]
  - HOL extends idea: domain \( D \) is indexed by (infinitely many) types

- Our presentation ignores polymorphism on the object-logical level, it is treated on the meta-level, though (a version covering object-level parametric polymorphism is [GM93]).
Model Based on Universe of Sets $\mathcal{U}$

Definition 1 (Universe):
$\mathcal{U}$ is a collection of sets, fulfilling closure conditions:

**Inhab:** Each $X \in \mathcal{U}$ is a nonempty set

**Sub:** If $X \in \mathcal{U}$ and $Y \neq \emptyset \subseteq X$, then $Y \in \mathcal{U}$

**Prod:** If $X, Y \in \mathcal{U}$ then $X \times Y \in \mathcal{U}$.

$X \times Y$ is Cartesian product, $\{\{x\}, \{x, y\}\}$ encodes $(x, y)$

**Pow:** If $X \in \mathcal{U}$ then $\mathcal{P}(X) = \{Y : Y \subseteq X\} \in \mathcal{U}$

**Infty:** $\mathcal{U}$ contains a distinguished infinite set $I$
Universe of Sets $\mathcal{U}$ (cont.)

- **Function space:**
  $X \Rightarrow Y$ is the set of (graphs of all total) functions from $X$ to $Y$
  - For $X$ and $Y$ nonempty, $X \Rightarrow Y$ is a nonempty subset of $\mathcal{P}(X \times Y)$
  - From closure conditions: $X, Y \in \mathcal{U}$ then so is $X \Rightarrow Y$.

- **Distinguished sets:**
  from **Infty** and **Sub** there is (at least one) set
  - **Unit:** A distinguished 1 element set $\{1\}$
  - **Bool:** A distinguished 2 element set $\{T, F\}$. 

Definition 2 (Frame):

A frame is a collection \((D_\alpha)_{\alpha \in \tau}\) with \(D_\alpha \in \mathcal{U}\), for \(\alpha \in \tau\) and

- \(D_{\text{bool}} = \{T, F\}\)
- \(D_{\text{ind}} = X\) where \(X\) is some infinite set of individuals
- \(D_{\alpha \Rightarrow \beta} \subseteq D_\alpha \Rightarrow D_\beta\), i.e., some collection of functions from \(D_\alpha\) to \(D_\beta\)

Example: \(D_{\text{bool} \Rightarrow \text{bool}}\) is some nonempty subset of functions from \(\{T, F\}\) to \(\{T, F\}\). Some of these subsets contain, e.g., the identity function, others do not.
Definition 3 (Interpretation):
An interpretation $\langle (D_\alpha)_{\alpha \in \tau}, J \rangle$ consists of a frame $(D_\alpha)_{\alpha \in \tau}$ and a denotation function $J$ mapping each constant of type $\alpha$ to an element of $D_\alpha$ where:

- $J(\text{True}) = T$ and $J(\text{False}) = F$
- $J(\equiv_{\alpha \Rightarrow \alpha \Rightarrow \text{bool}})$ is the identity on $D_\alpha$
- $J(\rightarrow)$ denotes the implication function over $D_{\text{bool}}$, i.e.,
  $$b \rightarrow b' = \begin{cases} 
  F & \text{if } b = T \text{ and } b' = F \\
  T & \text{otherwise}
  \end{cases}$$
- $J(\nu(\alpha \Rightarrow \text{bool}) \Rightarrow \alpha) \in (D_\alpha \Rightarrow D_{\text{bool}}) \Rightarrow D_\alpha$ denotes the function
  $$\text{the}(f) = \begin{cases} 
  a & \text{if } f = (\lambda x.x = a) \\
  y & \text{otherwise } (y \in D_\alpha \text{ is arbitrary})
  \end{cases}$$
Definition 4 (Generalized Models):
An interpretation $m = \langle (D_\alpha)_{\alpha \in \tau}, J \rangle$ is a (general) model for HOL iff there is a binary function $\mathcal{V}^m$ such that

- for all type-indexed families of substitutions $\sigma = (\sigma_\alpha)_{\alpha \in \tau}$ and terms $t$ of type $\alpha$, $\mathcal{V}^m(\sigma, t) \in D_\alpha$, and

- for all type-indexed families of substitutions $\sigma = (\sigma_\alpha)_{\alpha \in \tau}$,
  (a) $\mathcal{V}^m(\sigma, x_\alpha) = \sigma_\alpha(x_\alpha)$
  (b) $\mathcal{V}^m(\sigma, c) = J(c)$, for $c$ a (primitive) constant
  (c) $\mathcal{V}^m(\sigma, s_\alpha \Rightarrow_\beta t_\alpha) = \mathcal{V}^m(\sigma, s) \mathcal{V}^m(\sigma, t)$
    i.e., the value of the function $\mathcal{V}^m(\sigma, s)$ at the argument $\mathcal{V}^m(\sigma, t)$
  (d) $\mathcal{V}^m(\lambda x_\alpha. t_\beta) = \text{“the function from } D_\alpha \text{ into } D_\beta \text{ whose value for each } z \in D_\alpha \text{ is } \mathcal{V}^m(\sigma[x \leftarrow z], t)\text{”}
Generalized Models - Facts (1)

• If $\mathcal{M}$ is a general model and $\sigma$ a substitution, then $V^\mathcal{M}(\sigma, t)$ is uniquely determined, for every term $t$. $V^\mathcal{M}(\sigma, t)$ is value of $t$ in $\mathcal{M}$ w.r.t. $\sigma$.

• Gives rise to the standard notion of satisfiability/validity:
  ○ We write $V^\mathcal{M}, \sigma \models \phi$ for $V^\mathcal{M}(\sigma, \phi) = T$.
  ○ $\phi$ is satisfiable in $\mathcal{M}$ if $V^\mathcal{M}, \sigma \models \phi$, for some substitution $\sigma$.
  ○ $\phi$ is valid in $\mathcal{M}$ if $V^\mathcal{M}, \sigma \models \phi$, for every substitution $\sigma$.
  ○ $\phi$ is valid (in the general sense) if $\phi$ is valid in every general model $\mathcal{M}$. 

(rev. 12275)
Generalized Models - Facts (2)

- Not all interpretations are general models.
- Closure conditions guarantee every well-formed formula has a value under every assignment, e.g.,

  closure under functions: identity function from $D_\alpha$ to $D_\alpha$
  must belong to $D_{\alpha\Rightarrow\alpha}$ so that $\mathcal{V}^m(\sigma, \lambda x_\alpha. x)$ is defined.

  closure under application:
  
  - if $D_N$ is set of natural numbers and
  - $D_{N\Rightarrow N\Rightarrow N}$ contains addition function $p$ where $p x y = x + y$
  - then $D_{N\Rightarrow N}$ must contain $k x = 2x + 5$
    since $k = \mathcal{V}^m(\sigma, \lambda x. f(f x x) y)$ where $\sigma(f) = p$ and $\sigma(y) = 5$. 
Standard Models

Definition 5 (Standard Models):
A general model is a standard model iff for all $\alpha, \beta \in \tau$, $\mathcal{D}_{\alpha \Rightarrow \beta}$ is the set of all functions from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$.

- A standard model is a general model, but not necessary vice versa.
- Analogous definitions for satisfiability and validity w.r.t. standard models.
Standard Models

Definition 5 (Standard Models):
A general model is a standard model iff for all $\alpha, \beta \in \tau$, $\mathcal{D}_{\alpha \Rightarrow \beta}$ is the set of all functions from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$.

- A standard model is a general model, but not necessary vice versa.
- Analogous definitions for satisfiability and validity w.r.t. standard models.
- We can now re-introduce HOL in Isabelle’s meta-logic.
### Isabelle/HOL

The syntax of the core-language is introduced by:

<table>
<thead>
<tr>
<th>const</th>
<th>type definition</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Not</td>
<td><code>bool ⇒ bool</code></td>
<td>(&quot;¬ &quot; [40] 40)</td>
</tr>
<tr>
<td>True</td>
<td><code>bool</code></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td><code>bool</code></td>
<td></td>
</tr>
<tr>
<td>If</td>
<td><code>[bool, 'a, 'a] ⇒ 'a</code></td>
<td>(&quot;( if _ then _ else _)&quot;&quot;)</td>
</tr>
<tr>
<td>The</td>
<td><code>( 'a ⇒ bool) ⇒ 'a</code></td>
<td>(binder &quot;THE &quot; 10)</td>
</tr>
<tr>
<td>All</td>
<td><code>( 'a ⇒ bool) ⇒ bool</code></td>
<td>(binder &quot;∀ &quot; 10)</td>
</tr>
<tr>
<td>Ex</td>
<td><code>( 'a ⇒ bool) ⇒ bool</code></td>
<td>(binder &quot;∃ &quot; 10)</td>
</tr>
<tr>
<td>=</td>
<td><code>[ 'a, 'a] ⇒ bool</code></td>
<td>(infixl 50)</td>
</tr>
<tr>
<td>∧</td>
<td><code>[bool, bool] ⇒ bool</code></td>
<td>(infixr 35)</td>
</tr>
<tr>
<td>∨</td>
<td><code>[bool, bool] ⇒ bool</code></td>
<td>(infixr 30)</td>
</tr>
<tr>
<td>−→</td>
<td><code>[bool, bool] ⇒ bool</code></td>
<td>(infixr 25)</td>
</tr>
</tbody>
</table>
The Axioms of HOL (1)

axioms

refl : "t = t"

subst: "[ s = t; P(s) ] \implies P(t)"

ext : "(\forall x. f x = g x) \implies (\lambda x. f x) = (\lambda x. g x)"

impl: "(P \implies Q) \implies P \implies Q"

mp: "[ P \implies Q; P ] \implies Q"

iff : "(P \implies Q) \implies (Q \implies P) \implies (P = Q)"

True_or_False : "(P = \text{True}) \lor (P = \text{False})"

the_eq_trivial : "(\text{THE } x. x = a) = (a :: 'a)"
The Axioms of HOL (2)

Additionally, there is:

- universal $\alpha$, $\beta$, and $\eta$ congruence on terms (implicitly),
- the axiom of infinity, and
- the axiom of choice (Hilbert operator).

- This is the entire basis!
### Core Definitions of HOL

**defs**

<table>
<thead>
<tr>
<th>Definition</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>True_def:</td>
<td>True $\equiv (\lambda x :: \text{bool}. \ x = (\lambda x. \ x))$</td>
</tr>
<tr>
<td>All_def:</td>
<td>All $(P)$ $\equiv (P = (\lambda x. \ True))$</td>
</tr>
<tr>
<td>Ex_def:</td>
<td>Ex $(P)$ $\equiv \forall Q. \ (orall x. P \ x \rightarrow Q) \rightarrow Q$</td>
</tr>
<tr>
<td>False_def:</td>
<td>False $\equiv (\forall P. \ P)$</td>
</tr>
<tr>
<td>not_def:</td>
<td>$\neg P$ $\equiv P \rightarrow \text{False}$</td>
</tr>
<tr>
<td>and_def:</td>
<td>$P \land Q$ $\equiv \forall R. \ (P \rightarrow Q \rightarrow R) \rightarrow R$</td>
</tr>
<tr>
<td>or_def:</td>
<td>$P \lor Q$ $\equiv \forall R. \ (P \rightarrow R) \rightarrow (Q \rightarrow R) \rightarrow R$</td>
</tr>
<tr>
<td>if_def:</td>
<td>If $P \ x \ y$ $\equiv \text{THE} \ z :: \alpha. \ (P=\text{True} \rightarrow z=x) \land (P=\text{False} \rightarrow z=y)$</td>
</tr>
</tbody>
</table>
Meta-theoretic Properties of HOL

Theorem 1 (Soundness of HOL, [And86]):
HOL is sound w.r.t. to general models.

\( \vdash_{HOL} \phi \) implies \( \phi \) is valid

Theorem 2 (Completeness of HOL, [And86]):

- HOL is complete w.r.t. to general models.

\( \phi \) is valid implies \( \vdash_{HOL} \phi \)

- HOL is complete w.r.t. to standard models.

Theorem 3 (HOL with infinity, [And86]):

- HOL + infinity is complete w.r.t. general models.

- HOL + infinity is incomplete w.r.t. standard models.
Conclusions

• HOL generalizes semantics of FOL
  ◦ bool serves as type of propositions
  ◦ Syntax/semantics allows for higher-order functions

• Logic is rather minimal: 8 rules, more-or-less obvious

• Logic is very powerful in terms of what we can represent/derive.
  ◦ Other “logical” syntax
  ◦ Rich theories via conservative extensions
    (topic for next few weeks!)
Bibliography


References


[PM68] Dag Prawitz and Per-Erik Malmnäs. A survey of some connections between classical, intuitionistic and minimal logic. In A. Schmidt and H. Schütte, ed-


