Computer Supported Modeling and Reasoning

David Basin, Achim D. Brucker, Jan-Georg Smaus, and Burkhart Wolff

April 2005

http://www.infsec.ethz.ch/education/permanent/csmr/

Higer-Order Logic: Foundations

David Basin

Motivation

Motivation (1)

Higher-Order Logic (HOL) is:

Motivation (1)

Higher-Order Logic (HOL) is:

- an expressive foundation for

  mathematics: analysis, algebra, ...

  computer science: program correctness, hardware verification, ...

Motivation (1)

Higher-Order Logic (HOL) is:

- an expressive foundation for

  mathematics: analysis, algebra, ...

  computer science: program correctness, hardware verification, ...

- a logical framework for embedding languages/deductive systems.

  In contrast, there is no meta/object distinction then.

  Everything is defined within HOL. Reasoning is classical.

Motivation (2)

- Still modeling of problems is important (now in HOL)
  So is deriving relevant reasoning principles

Motivation (2)

- Still modeling of problems is important (now in HOL)
  So is deriving relevant reasoning principles
- We will use Isabelle/HOL

Motivation (2)

- Still modeling of problems is important (now in HOL)
  So is deriving relevant reasoning principles
- We will use Isabelle/HOL
  - Could forgo the use of a metalogic and employ alternatives, e.g., HOL system or PVS. Or use constructive alternatives such as Coq or Nuprl.

Motivation (2)

- Still modeling of problems is important (now in HOL)
  So is deriving relevant reasoning principles
- We will use Isabelle/HOL
  - Could forgo the use of a metalogic and employ alternatives, e.g., HOL system or PVS. Or use constructive alternatives such as Coq or Nuprl.
  - Choice depends on culture and application

Motivation (2)

- Still modeling of problems is important (now in HOL)
  So is deriving relevant reasoning principles
- We will use Isabelle/HOL
  - Could forgo the use of a metalogic and employ alternatives, e.g., HOL system or PVS. Or use constructive alternatives such as Coq or Nuprl.
  - Choice depends on culture and application

Motivation (3)

- HOL offers safety through strength

Motivation (3)

- HOL offers safety through strength

  Safety via conservative (definitional) extensions: functions, relations, inductive definitions, ...

Motivation (3)

- HOL offers safety through strength

  Safety via conservative (definitional) extensions: functions, relations, inductive definitions, ...
  - Extend theory with new constants and types defined by existing ones
  - Derive properties

Motivation (3)

- HOL offers safety through strength

  Safety via conservative (definitional) extensions: functions, relations, inductive definitions, ...

  - Extend theory with new constants and types defined by existing ones
  - Derive properties

- Contrast with

Motivation (3)

- HOL offers safety through strength

  Safety via conservative (definitional) extensions: functions, relations, inductive definitions, ...
  - Extend theory with new constants and types defined by existing ones
  - Derive properties
- Contrast with
  - Use of weak logics (e.g., propositional logic): can't define much
  - Axiomatic extensions: can lead to inconsistency

Motivation (3)

- HOL offers safety through strength

  Safety via conservative (definitional) extensions: functions, relations, inductive definitions, ...
  - Extend theory with new constants and types defined by existing ones
  - Derive properties
- Contrast with
  - Use of weak logics (e.g., propositional logic): can't define much
  - Axiomatic extensions: can lead to inconsistency

Bertrand Russel once likened the advantages of postulation over definition to the advantages of theft over honest toil!

Which Foundation?

- Set theory: the choice as basis for modern mathematics
  - ZFC (in Isabelle): impressive applications!
  - Bernays-Gödel: used for resolution since finitely axiomatizable

Which Foundation?

- Set theory: the choice as basis for modern mathematics
  - ZFC (in Isabelle): impressive applications!
  - Bernays-Gödel: used for resolution since finitely axiomatizable
- 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 V is not a set.
  - A class cannot belong to another class (let alone a set)!

Which Foundation?

- Set theory: the choice as basis for modern mathematics
  - ZFC (in Isabelle): impressive applications!
  - Bernays-Gödel: used for resolution since finitely axiomatizable
- 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 V is not a set.
  - A class cannot belong to another class (let alone a set)!
- HOL as alternative (Church 1940, Henkin 1950)
  - **Rationale:** one usually works with typed entities
  - Reasoning is then easier with support for types
  - 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. (Paulson)

HOL — Why Higher-Order? (1)

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

$$\forall x, y.\, R(x, y) \longrightarrow R(y, x)$$

2nd-order: quantification over predicates/functions

$$\begin{aligned} false &\equiv \forall P.\, P \\ P \wedge Q &\equiv \forall R.\, (P \longrightarrow Q \longrightarrow R) \longrightarrow R \end{aligned}$$

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

- Instead of defining:

$$subrel(R, S) \equiv \forall x.\, R(x) \longrightarrow S(x)$$

- Abstract and use:

$$\forall X.\, (X(R, S) \Leftrightarrow \forall x.\, R(x) \longrightarrow S(x)) \longrightarrow \ldots X(R', S') \ldots$$

HOL — Why Higher-Order? (2)

- Hierarchy: 4th, 5th, ... order logic

HOL — Why Higher-Order? (2)

- Hierarchy: 4th, 5th, ... order logic
- Omega-order logic: includes logics of all finite orders.

  Also called finite-type theory or higher-order logic

HOL — Why Higher-Order? (2)

- Hierarchy: 4th, 5th, ... order logic
- Omega-order logic: includes logics of all finite orders.
  Also called finite-type theory or higher-order logic
- Contrast: quantification over propositions versus functions
  - in LF we quantified over functions at all types

$$\forall f, g : N \to N. \forall x : n. f(x) =_N g(x) \Rightarrow f =_{N \to N} g$$

  - Result is a proposition and we cannot quantify over these!

HOL — Why Higher-Order? (2)

- Hierarchy: 4th, 5th, ... order logic
- Omega-order logic: includes logics of all finite orders.
  Also called finite-type theory or higher-order logic
- Contrast: quantification over propositions versus functions
  - in LF we quantified over functions at all types

$$\forall f, g : N \to N. \forall x : n. f(x) =_N g(x) \Rightarrow f =_{N \to N} g$$

  - Result is a proposition and we cannot quantify over these!

(LF sometimes called first-order! Better is minimal predicate logic with quantification over higher-types.)

Core-HOL: Syntax

Basic HOL Syntax (1)

- Types:

$$\tau \ ::= \ bool \mid ind \mid \tau \Rightarrow \tau$$

  - $bool$ and ind are also called o and i in literature [Chu40, And86]
  - Isabelle allows definitions of new type constructors (e.g., $A \times B$)
  - Isabelle supports polymorphic type definitions, e.g., $list(\alpha)$

Basic HOL Syntax (1)

- Types:

$$\tau \;::=\; bool \mid ind \mid \tau \Rightarrow \tau$$

  - $bool$ and ind are also called o and i in literature [Chu40, And86]
  - Isabelle allows definitions of new type constructors (e.g., $A \times B$)
  - Isabelle supports polymorphic type definitions, e.g., $list(\alpha)$

- Terms: ($\mathcal{V}$ is set of variables, and $\mathcal{C}$ 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 $bool$ are called (well-formed) formulae.

Compare with Isabelle's Pure

Basic HOL Syntax (2)

- Constants are always supplied with types and include:

  $True, False : bool$

  $\_ = \_ : \tau \Rightarrow \tau \Rightarrow bool$ (for all types τ)

  $\_ \to \_ : bool \Rightarrow bool \Rightarrow bool$

  $\iota\_ : (\tau \Rightarrow bool) \Rightarrow \tau$ (for all types τ)

Basic HOL Syntax (2)

- Constants are always supplied with types and include:

  $True, False : bool$

  $\_ = \_ : \tau \Rightarrow \tau \Rightarrow bool$ (for all types τ)

  $\_ \rightarrow \_ : bool \Rightarrow bool \Rightarrow bool$

  $\iota\_ : (\tau \Rightarrow bool) \Rightarrow \tau$ (for all types τ)

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

Basic HOL Syntax (2)

- Constants are always supplied with types and include:

  $True, False : bool$

  $\_ = \_ : \tau \Rightarrow \tau \Rightarrow bool$ (for all types τ)

  $\_ \rightarrow \_ : bool \Rightarrow bool \Rightarrow bool$

  $\iota\_ : (\tau \Rightarrow bool) \Rightarrow \tau$ (for all types τ)

Note that the description operator ιf 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 τ" can be expressed by using polymorphic type variables α.

Core-HOL: Semantics

HOL Semantics

- Intuitively an extension of many-sorted semantics with functions
  - FOL: structure is domain and functions/relations

$$\langle \mathcal{D}, f_1, \ldots, f_k, r_1, \ldots, r_j \rangle$$

  - Many-sorted FOL: domains are sort-indexed

$$\langle \mathcal{D}_1, \ldots \mathcal{D}_n, f_1, \ldots, f_k, r_1, \ldots, r_j \rangle$$

    (if no relations then we have a heterogenous Algebra)
  - HOL extends idea: domain $\mathcal{D}$ 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 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 nonempty and a 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 $\{T, F\}$.

Frames

Definition 2 (Frame):

A frame is a collection $\mathcal{D}_\alpha$ of sets, $\mathcal{D}_\alpha \in \mathcal{U}$, for $\alpha \in \tau$ where:

- $\mathcal{D}_{bool} = \{T, F\}$
- $\mathcal{D}_{Ind} = X$ where X is some infinite set of individuals
- $\mathcal{D}_{\alpha \Rightarrow \beta} \subseteq \mathcal{D}_\alpha \Rightarrow \mathcal{D}_\beta$, i.e., some collection of functions from D_α to D_β

Frames

Definition 2 (Frame):

A frame is a collection $\mathcal{D}_\alpha$ of sets, $\mathcal{D}_\alpha \in \mathcal{U}$, for $\alpha \in \tau$ where:

- $\mathcal{D}_{bool} = \{T, F\}$
- $\mathcal{D}_{Ind} = X$ where X is some infinite set of individuals
- $\mathcal{D}_{\alpha \Rightarrow \beta} \subseteq \mathcal{D}_\alpha \Rightarrow \mathcal{D}_\beta$, i.e., some collection of functions from D_α to D_β

Example: $\mathcal{D}_{bool \Rightarrow bool}$ 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.

Interpretations

Definition 3 (Interpretation):

An interpretation $\langle \mathcal{D}_\alpha, \mathcal{J} \rangle$ is a frame $\mathcal{D}_\alpha$ and a denotation function $\mathcal{J}$ mapping each constant of type α to an element of $\mathcal{D}_\alpha$ where:

- $\mathcal{J}(\mathit{True}) = T$ and $\mathcal{J}(\mathit{False}) = F$
- $\mathcal{J}(=_{\alpha \Rightarrow \alpha \Rightarrow bool})$ is identity on $\mathcal{D}_\alpha$
- $\mathcal{J}(\rightarrow)$ denotes implication function over $\mathcal{D}_{bool}$.
  I.e., it sends $b, b' \in \{T, F\}$ to

$$b \longrightarrow b' = \begin{cases} F & \text{if } b = T \text{ and } b' = F \\ T & \text{otherwise} \end{cases}$$

- $\mathcal{J}(\iota_{(\alpha \Rightarrow bool) \Rightarrow \alpha}) \in (\mathcal{D}_\tau \Rightarrow \mathcal{D}_{bool}) \Rightarrow \mathcal{D}_\tau$ denotes the function

$$ch(f) = \begin{cases} a & \text{if } f = (\lambda x.x = a) \\ y & \text{otherwise} \end{cases}$$

for an arbitrary $y \in D_\alpha$ and an $f \in \mathcal{D}_\alpha \Rightarrow \mathcal{D}_{bool}$

- $\mathcal{J}(\iota_{(\alpha \Rightarrow bool) \Rightarrow \alpha}) \in (\mathcal{D}_\tau \Rightarrow \mathcal{D}_{bool}) \Rightarrow \mathcal{D}_\tau$ denotes the function

$$ch(f) = \begin{cases} a & \text{if } f = (\lambda x.x = a) \\ y & \text{otherwise} \end{cases}$$

for an arbitrary $y \in D_\alpha$ and an $f \in \mathcal{D}_\alpha \Rightarrow \mathcal{D}_{bool}$

Note: the notion of an interpretation generalizes the notion of a structure to a higher-order setting.

Generalized Models

Definition 4 (Generalized Models):

An interpretation $\mathcal{M} = \langle \mathcal{D}_\alpha, \mathcal{J} \rangle$ is a (general) model for HOL iff there is a function $\mathcal{V}_A^{\mathfrak{M}}$ such that for all type-indexed families of substitutions $\sigma = \{\sigma_\alpha\}_{\alpha \in \mathcal{T}}$ and terms, the following closure conditions hold:

1. $\mathcal{V}_A^{\mathfrak{M}}(x_\alpha) = \sigma(x_\alpha)$ (i.e., $\sigma_\alpha(x_\alpha)$)
2. $\mathcal{V}_A^{\mathfrak{M}}(c) = \mathcal{J}(c)$ for c a (primitive) constant
3. $\mathcal{V}_A^{\mathfrak{M}}(s_{\alpha \Rightarrow \beta} t_\alpha) = (\mathcal{V}_A^{\mathfrak{M}}(s))(\mathcal{V}_A^{\mathfrak{M}}(t))$
   i.e., the value of the function $\mathcal{V}_A^{\mathfrak{M}}(s)$ at the argument $\mathcal{V}_A^{\mathfrak{M}}(t)$

4. $\mathcal{V}_A^{\mathfrak{M}}(\lambda x_\alpha .\, t_\beta)$ = the function from $\mathcal{D}_\alpha$ into $\mathcal{D}_\beta$ whose value for each $z \in \mathcal{D}_\alpha$ is $\mathcal{V}_{\sigma[x \leftarrow z]}^{\mathcal{M}}(t)$

Generalized Models - Facts (1)

- **If** $\mathcal{M}$ is a general model and σ a substitution, **then** $\mathcal{V}_A^{\mathfrak{M}}$ is uniquely determined.
  $\mathcal{V}_A^{\mathfrak{M}}(t)$ is value of t in $\mathcal{M}$ wrt σ.
- Gives rise to the standard notion of satisfiability of formulae

$$\mathcal{V}_A^{\mathfrak{M}} \models \phi \text{iff } \mathcal{V}_A^{\mathfrak{M}}(\phi) = T$$

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 $\mathcal{D}_\alpha$ to $\mathcal{D}_\alpha$ must always belong to $\mathcal{D}_{\alpha\Rightarrow\alpha}$ so that $\mathcal{V}_A^{\mathfrak{M}}(\lambda x_\alpha .\, x)$ defined.

  closure under application:
  - if $\mathcal{D}_N$ is natural numbers and
  - $\mathcal{D}_{N\Rightarrow N\Rightarrow N}$ contains addition function p where $p\, x\, y = x + y$
  - then $\mathcal{D}_{N\Rightarrow N}$ must contain $k\, x = 2x + 5$ since $k = \mathcal{V}_A^{\mathfrak{M}}(\lambda x_N .\, f\,(f\, x\, x)\, y)$ 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.

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.

We can now re-introduce HOL in Isabelle/Pure.

Isabelle/HOL

The syntax of the core-language is introduced by:

Trueprop	::	bool $\Rightarrow$ prop	("(_)" 5)
Not	::	bool $\Rightarrow$ bool	("$\neg$ _" [40] 40)
True	::	bool	
False	::	bool	
If	::	[bool, 'a, 'a] $\Rightarrow$ 'a	("(if _ then _ else _)")
The	::	('a $\Rightarrow$ bool) $\Rightarrow$ 'a	(**binder** "THE " 10)
All	::	('a $\Rightarrow$ bool) $\Rightarrow$ bool	(**binder** "$\forall$ " 10)
Ex	::	('a $\Rightarrow$ bool) $\Rightarrow$ bool	(**binder** "$\exists$ " 10)
$=$	::	['a, 'a] $\Rightarrow$ bool	(**infixl** 50)
$\wedge$	::	[bool, bool] $\Rightarrow$ bool	(infixr 35)
$\vee$	::	[bool, bool] $\Rightarrow$ bool	(infixr 30)
$\longrightarrow$	::	[bool, bool] $\Rightarrow$ bool	(infixr 25)

The Axioms of HOL (1)

axioms

refl : "$t = t$"

subst: "$\llbracket\ s = t;\ P(s)\ \rrbracket \Longrightarrow P(t)$"

ext : "$(\bigwedge x.\ f\ x = g\ x) \Longrightarrow (\lambda x.\ f\ x) = (\lambda x.\ g\ x)$"

impl : "$(P \Longrightarrow Q) \Longrightarrow P \longrightarrow Q$"

mp: "$\llbracket\ P \longrightarrow Q;\ P\ \rrbracket \Longrightarrow Q$"

iff : "$(P \longrightarrow Q) \longrightarrow (Q \longrightarrow P) \longrightarrow (P = Q)$"

True_or_False : "$(P = True) \vee (P = False)$"

the_eq_trivial :"(THE x. x = a) = (a::'a)"

The Axioms of HOL (2)

Additionally, there is:

- universal α,β and η congruence on terms (implicitly),

The Axioms of HOL (2)

Additionally, there is:

- universal α,β and η congruence on terms (implicitly),
- the axiom of infinity,

The Axioms of HOL (2)

Additionally, there is:

- universal α,β and η congruence on terms (implicitly),
- the axiom of infinity,
- This is the entire basis!

Core-HOL: Meta-theoretic Properties

Meta-theoretic Properties of HOL

Theorem 1 (Soundness of HOL):

HOL is sound w.r.t. to generalized models.

$$\vdash_{HOL} \phi \Longrightarrow \mathcal{V}_A^{\mathfrak{M}} \models \phi$$

Meta-theoretic Properties of HOL

Theorem 1 (Soundness of HOL):

HOL is sound w.r.t. to generalized models.

$$\vdash_{HOL} \phi \Longrightarrow \mathcal{V}_A^{\mathfrak{M}} \models \phi$$

Theorem 2 (Completeness of HOL):

HOL is complete w.r.t. to generalized models.

$$\mathcal{V}_A^{\mathfrak{M}} \models \phi \Longrightarrow \vdash_{HOL} \phi$$

Meta-theoretic Properties of HOL

Theorem 1 (Soundness of HOL):

HOL is sound w.r.t. to generalized models.

$$\vdash_{HOL} \phi \Longrightarrow \mathcal{V}_A^{\mathfrak{M}} \models \phi$$

Theorem 2 (Completeness of HOL):

HOL is complete w.r.t. to generalized models.

$$\mathcal{V}_A^{\mathfrak{M}} \models \phi \Longrightarrow \vdash_{HOL} \phi$$

Theorem 3 (Completeness of HOL (without Infinity)):
HOL without the axiom of infinity is complete w.r.t. to standard models.

Theorem 3 (Completeness of HOL (without Infinity)): HOL without the axiom of infinity is complete w.r.t. to standard models.

Theorem 4 (Incompleteness of HOL): HOL is incomplete w.r.t. standard models.

Theorem 3 (Completeness of HOL (without Infinity)):
HOL without the axiom of infinity is complete w.r.t. to standard models.

Theorem 4 (Incompleteness of HOL):
HOL is incomplete w.r.t. standard models.

For the proofs, see [And86].

Core Definitions of HOL

True_def:	True	$\equiv ((\lambda x{::}bool.\ x) = (\lambda x.\ x))$
All_def:	All (P)	$\equiv (P = (\lambda x.\ True))$
Ex_def:	Ex(P)	$\equiv \forall Q.\ (\forall x.\ P\ x \longrightarrow Q) \longrightarrow Q$
False_def:	False	$\equiv (\forall P.\ P)$
not_def:	$\neg$ P	$\equiv P \longrightarrow False$
and_def:	$P \wedge Q$	$\equiv \forall R.\ (P \longrightarrow Q \longrightarrow R) \longrightarrow R$
or_def:	$P \vee Q$	$\equiv \forall R.\ (P \longrightarrow R) \longrightarrow (Q \longrightarrow R) \longrightarrow R$
if_def:	If P x y	$\equiv THE\ z{::}'a.\ (P{=}True \longrightarrow z{=}x) \wedge (P{=}False \longrightarrow z{=}y)$

Definitions can be understood either

semantically: so-called shallow semantic embedding, or as

derived rules: in Isabelle, i.e., by their properties.

Conclusion

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

- M. J. C. Gordon and T. F. Melham, Introduction to HOL: A theorem proving environment for higher order logic, Cambridge University Press, 1993.
- Peter B. Andrews, An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof, Academic Press, 1986.
- Tobias Nipkow and Lawrence C. Paulson and Markus Wenzel, Isabelle/HOL — A Proof Assistant for Higher-Order Logic, Springer-Verlag, LNCS 2283, 2002.

References

[And86] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proofs*. Academic Press, 1986.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[GM93] Michael J. C. Gordon and Tom F. Melham, editors. *Introduction to HOL*. Cambridge University Press, 1993.