COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Simon Winwood, Toby Murray, June Andronick, Gerwin Klein

HOL

Slide 1

Defining Higher Order Logic

Slide 2

What is Higher Order Logic?

- **Propositional Logic:**
  - no quantifiers
  - all variables have type bool
- **First Order Logic:**
  - quantification over values, but not over functions and predicates,
  - terms and formulas syntactically distinct
- **Higher Order Logic:**
  - quantification over everything, including predicates
  - consistency by types
  - formula = term of type bool
  - definition built on $\lambda^{\rightarrow}$ with certain default types and constants

Slide 3

Defining Higher Order Logic

Default types:

bool $\qquad$ $\_ \Rightarrow \_$ $\qquad$ ind

- **bool** sometimes called o
- $\Rightarrow$ sometimes called fun

Default Constants:

$$\begin{array}{lll} \longrightarrow & :: & bool \Rightarrow bool \Rightarrow bool \\ = & :: & \alpha \Rightarrow \alpha \Rightarrow bool \\ \epsilon & :: & (\alpha \Rightarrow bool) \Rightarrow \alpha \end{array}$$

Slide 4

Higher Order Abstract Syntax

Problem: Define syntax for binders like $\forall$, $\exists$, ε

One approach: $\forall :: var \Rightarrow term \Rightarrow bool$
Drawback: need to think about substitution, α conversion again.

But: Already have binder, substitution, α conversion in meta logic

$$\lambda$$

So: Use λ to encode all other binders.

Slide 5

Higher Order Abstract Syntax

Example:

ALL $:: (\alpha \Rightarrow bool) \Rightarrow bool$

HOAS	**usual syntax**
ALL $(\lambda x.\ x = 2)$	$\forall x.\ x = 2$
ALL P	$\forall x.\ P\ x$

Isabelle can translate usual binder syntax into HOAS.

Slide 6

Side Track: Syntax Declarations in Isabelle

- **mixfix:**
  consts drvbl $:: ct \Rightarrow ct \Rightarrow fm \Rightarrow bool$ $(\text{"}\_, \_ \vdash \_\text{"})$
  Legal syntax now: $\Gamma, \Pi \vdash F$
- **priorities:**
  pattern can be annotated with priorities to indicate binding strength
  Example: drvbl $:: ct \Rightarrow ct \Rightarrow fm \Rightarrow bool$ $(\text{"}\_, \_ \vdash \_\text{"}\ [30, 0, 20]\ 60)$
- **infixl/infixr**: short form for left/right associative binary operators
  Example: or $:: bool \Rightarrow bool \Rightarrow bool$ (infixr " $\vee$ " 30)
- **binders:** declaration must be of the form
  $c :: (\tau_1 \Rightarrow \tau_2) \Rightarrow \tau_3$ (binder "B" $< p >$)
  $B\ x.\ P\ x$ translated into $c\ P$ (and vice versa)
  Example ALL $:: (\alpha \Rightarrow bool) \Rightarrow bool$ (binder "$\forall$" 10)

More (including pretty printing) in Isabelle Reference Manual (7.3)

Slide 7

Back to HOL

Base: $bool, \Rightarrow, ind \qquad =, \longrightarrow, \varepsilon$

And the rest is definitions:

True	$\equiv (\lambda x :: bool.\ x) = (\lambda x.\ x)$
All P	$\equiv P = (\lambda x.$ True$)$
Ex P	$\equiv \forall Q.\ (\forall x.\ P\ x \longrightarrow Q) \longrightarrow Q$
False	$\equiv \forall P.\ P$
$\neg P$	$\equiv P \longrightarrow$ False
$P \wedge Q$	$\equiv \forall R.\ (P \longrightarrow Q \longrightarrow R) \longrightarrow R$
$P \vee Q$	$\equiv \forall R.\ (P \longrightarrow R) \longrightarrow (Q \longrightarrow R) \longrightarrow R$
If $P\ x\ y$	$\equiv$ SOME $z.\ (P =$ True $\longrightarrow z = x) \wedge (P =$ False $\longrightarrow z = y)$
inj f	$\equiv \forall x\ y.\ f\ x = f\ y \longrightarrow x = y$
surj f	$\equiv \forall y.\ \exists x.\ y = f\ x$

Slide 8

The Axioms of HOL

$$\frac{}{t = t}\ \text{refl} \qquad \frac{s = t \quad P\ s}{P\ t}\ \text{subst} \qquad \frac{\bigwedge x.\ f\ x = g\ x}{(\lambda x.\ f\ x) = (\lambda x.\ g\ x)}\ \text{ext}$$

$$\frac{P \Longrightarrow Q}{P \longrightarrow Q}\ \text{impl} \qquad \frac{P \longrightarrow Q \quad P}{Q}\ \text{mp}$$

$$\frac{}{(P \longrightarrow Q) \longrightarrow (Q \longrightarrow P) \longrightarrow (P = Q)}\ \text{iff}$$

$$\frac{}{P = \text{True} \vee P = \text{False}}\ \text{True\_or\_False}$$

$$\frac{P\ ?x}{P\ (\text{SOME}\ x.\ P\ x)}\ \text{someI}$$

$$\frac{}{\exists f :: ind \Rightarrow ind.\ \text{inj}\ f \wedge \neg\text{surj}\ f}\ \text{infty}$$

Slide 9

DEMO: THE DEFINITIONS IN ISABELLE

Slide 11

That's it.

- ➜ 3 basic constants
- ➜ 3 basic types
- ➜ 9 axioms

With this you can define and derive all the rest.

Isabelle knows 2 more axioms:

$$\frac{x = y}{x \equiv y}\ \text{eq\_reflection} \qquad \frac{}{(\text{THE}\ x.\ x = a) = a}\ \text{the\_eq\_trivial}$$

Slide 10

Deriving Proof Rules

In the following, we will

- ➜ look at the definitions in more detail
- ➜ derive the traditional proof rules from the axioms in Isabelle

Convenient for deriving rules: **named assumptions in lemmas**

lemma $[name :]$
assumes $[name_1 :]$ "$< prop >_1$"
assumes $[name_2 :]$ "$< prop >_2$"
$\vdots$
shows " $< prop >$ " $< proof >$

proves: $[\![< prop >_1; < prop >_2; \ldots]\!] \Longrightarrow < prop >$

Slide 12

True

consts True :: $bool$

True $\equiv (\lambda x :: bool.\ x) = (\lambda x.\ x)$

Intuition:

right hand side is always true

Proof Rules:

$$\frac{}{\text{True}}\ \text{TrueI}$$

Proof:

$$\frac{\overline{(\lambda x :: bool.\ x) = (\lambda x.\ x)}\ \text{refl}}{\text{True}}\ \text{unfold True\_def}$$

Slide 13

DEMO

Slide 14

Universal Quantifier

consts ALL :: $(\alpha \Rightarrow bool) \Rightarrow bool$

ALL $P \quad \equiv \quad P = (\lambda x.\ \text{True})$

Intuition:

- ALL P is Higher Order Abstract Syntax for $\forall x.\ P\ x$.
- P is a function that takes an x and yields a truth values.
- ALL P should be true iff P yields true for all x, i.e. if it is equivalent to the function λx. True.

Proof Rules:

$$\frac{\bigwedge x.\ P\ x}{\forall x.\ P\ x}\ \text{allI} \qquad \frac{\forall x.\ P\ x \quad P\ ?x \Longrightarrow R}{R}\ \text{allE}$$

Proof: Isabelle Demo

Slide 15

False

consts False :: $bool$

False $\quad \equiv \quad \forall P.P$

Intuition:

Everything can be derived from *False*.

Proof Rules:

$$\frac{\text{False}}{P}\ \text{FalseE} \qquad \frac{}{\text{True} \neq \text{False}}$$

Proof: Isabelle Demo

Slide 16

Negation

consts Not :: $bool \Rightarrow bool$ $(\neg\ \_)$
$\neg P \equiv P \longrightarrow$ False

Intuition:
Try $P = True$ and $P = False$ and the traditional truth table for $\longrightarrow$.

Proof Rules:

$$\frac{A \Longrightarrow False}{\neg A}\ \text{notI} \qquad \frac{\neg A \quad A}{P}\ \text{notE}$$

Proof: Isabelle Demo

Slide 17

Existential Quantifier

consts EX :: $(\alpha \Rightarrow bool) \Rightarrow bool$
EX $P \quad \equiv \quad \forall Q.\ (\forall x.\ P\ x \longrightarrow Q) \longrightarrow Q$

Intuition:

- EX P is HOAS for $\exists x.\ P\ x.$ (like $\forall$)
- Right hand side is characterization of $\exists$ with $\forall$ and $\longrightarrow$
- Note that inner $\forall$ binds wide: $(\forall x.\ P\ x \longrightarrow Q)$
- Remember lemma from last time: $(\forall x.\ P\ x \longrightarrow Q) = ((\exists x.\ P\ x) \longrightarrow Q)$

Proof Rules:

$$\frac{P\ ?x}{\exists x.\ P\ x}\ \text{exI} \qquad \frac{\exists x.\ P\ x \quad \bigwedge x.\ P\ x \Longrightarrow R}{R}\ \text{exE}$$

Proof: Isabelle Demo

Slide 18

Conjunction

consts And :: $bool \Rightarrow bool \Rightarrow bool$ $(\_ \wedge \_)$
$P \wedge Q \equiv \forall R.\ (P \longrightarrow Q \longrightarrow R) \longrightarrow R$

Intuition:

- Mirrors proof rules for $\wedge$
- Try truth table for P, Q, and R

Proof Rules:

$$\frac{A \quad B}{A \wedge B}\ \text{conjI} \qquad \frac{A \wedge B \quad [\![A; B]\!] \Longrightarrow C}{C}\ \text{conjE}$$

Proof: Isabelle Demo

Slide 19

Disjunction

consts Or :: $bool \Rightarrow bool \Rightarrow bool$ $(\_ \vee \_)$
$P \vee Q \equiv \forall R.\ (P \longrightarrow R) \longrightarrow (Q \longrightarrow R) \longrightarrow R$

Intuition:

- Mirrors proof rules for $\vee$ (case distinction)
- Try truth table for P, Q, and R

Proof Rules:

$$\frac{A}{A \vee B}\ \frac{B}{A \vee B}\ \text{disjI1/2} \qquad \frac{A \vee B \quad A \Longrightarrow C \quad B \Longrightarrow C}{C}\ \text{disjE}$$

Proof: Isabelle Demo

Slide 20

If-Then-Else

consts If :: $bool \Rightarrow \alpha \Rightarrow \alpha \Rightarrow \alpha$ (if_ then _ else _)
If $P\ x\ y\ \equiv$ SOME $z.\ (P = \text{True} \longrightarrow z = x) \wedge (P = \text{False} \longrightarrow z = y)$

Intuition:

- for $P =$ True, right hand side collapses to SOME $z.\ z = x$
- for $P =$ False, right hand side collapses to SOME $z.\ z = y$

Proof Rules:

$$\frac{}{\text{if True then } s \text{ else } t = s}\ \text{ifTrue} \qquad \frac{}{\text{if False then } s \text{ else } t = t}\ \text{ifFalse}$$

Proof: Isabelle Demo

Slide 21

That was HOL

Slide 22

More on Automation

Last time: safe and unsafe rule, heuristics: use safe before unsafe

This can be automated

Syntax:

[<kind>!]	for safe rules (<kind> one of intro, elim, dest)
[<kind>]	for unsafe rules

Application (roughly):
do safe rules first, search/backtrack on unsafe rules only

Example:

declare attribute globally	**declare** conjI [intro!] allE [elim]
remove attribute gloablly	**declare** allE [rule del]
use locally	**apply** (blast intro: someI)
delete locally	**apply** (blast del: conjI)

Slide 23

Demo: Automation

Slide 24

We have learned today ...

NICTA

- ➜ Defining HOL
- ➜ Higher Order Abstract Syntax
- ➜ Deriving proof rules
- ➜ More automation

Slide 25