Course 2D1453, 2006-07

Advanced Formal Methods

Lecture 6: Isabelle - HOL

Mads Dam
KTH/CSC

Material from L. Paulson

What Is Higher Order Logic?

Propositional logic
- No quantifiers
- All variables have type bool

First Order Logic
- Quantification over values of base type
- Terms and formulas are syntactically distinct

Higher Order Logic
- Quantification over functions and predicates
- Consistency by typing
- Formula = term of type bool
- Predicate = function with codomain bool
- $\lambda_{\rightarrow}$ + a few types and constants

Natural Deduction

Two kinds of rules for each logical operator $\oplus$

Introduction rules:
- How can $A \oplus B$ be proved?

Elimination rules:
- What can be inferred from $A \oplus B$?

Natural deduction calculus:
- Proof trees may have unproven leaves = assumptions
- Assumptions can be introduced and discharged

Sequent calculus:
- All assumptions (and alternative conclusions) represented explicitly in proof judgments

Rule Notation

Write $\dfrac{A_1 \quad \ldots \quad A_n}{A}$ RuleName

Instead of $[\![A_1 ; \ldots ; A_n]\!] \Rightarrow A$

In other words:
- Stipulating an inference rule "RuleName"

Same as:
- Declaring an Isabelle metalogic term $[\![A_1 ; \ldots ; A_n]\!] \Rightarrow A$ to be provable by named rule

Derived rule $[\![A_1 ; \ldots ; A_n]\!] \Rightarrow A$
- Rule is provable in Isabelle's metalogic

Natural Deduction, Propositional Logic

$\dfrac{A \quad B}{A \wedge B}\ \wedge I$

$\dfrac{A \wedge B \quad [\![A;B]\!] \Rightarrow C}{C}\ \wedge E$

$\dfrac{A}{A \vee B} \quad \dfrac{B}{A \vee B}\ \vee I1/2$

$\dfrac{A \vee B \quad A \Rightarrow C \quad B \Rightarrow C}{C}\ \vee E$

$\dfrac{A \Rightarrow B}{A \rightarrow B}\ \Rightarrow I$

$\dfrac{A \Rightarrow B \quad A \quad B \Rightarrow C}{C}\ \Rightarrow E$

$\dfrac{A \Rightarrow B \quad B \Rightarrow A}{A = B}\ iffI$

$\dfrac{A = B}{A \Rightarrow B} \quad \dfrac{A = B}{B \Rightarrow A}\ iffD1/2$

$\dfrac{A \Rightarrow False}{\neg A}\ \neg I$

$\dfrac{\neg A \quad A}{C}\ \neg E$

D for "definition"

Equality

$\dfrac{-}{t = t}\ =I$

$\dfrac{s = t \quad A[s/x]}{A[t/x]}\ =E$

Exercise 1: Prove that the following rules are derived:

$\dfrac{s = t}{t = s}\ Sym$

$\dfrac{r = s \quad s = t}{r = t}\ Trans$

$\dfrac{s = t \quad A[s/x] \quad A[t/x] \Rightarrow C}{C}\ =E'$

More Rules

$$\frac{A \to B \quad A}{B}\ \text{mp}$$

$$\frac{\neg A \Rightarrow False}{A}\ \text{ccontr} \qquad \frac{\neg A \Rightarrow A}{A}\ \text{classical}$$

ccontr and classical not derivable from other rules
They make the logic "classical", i.e. non-constructive

Proof by Assumption

Implicit in Isabelle's metalogic

$[\![A_1 ; ... ; A_n]\!] \Rightarrow A_i$ provable for any i: $1 \leq i \leq n$

In isabelle:
apply assumption
proves
1. $[\![B_1 ; ... ; B_n]\!] \Rightarrow C$
by unifying C with some B_i, $1 \leq i \leq n$

Note: This may cause backtracking!

Rule Application

Rule: $[\![A_1 ; ... ; A_n]\!] \Rightarrow A$
Subgoal:
1. $[\![B_1 ; ... ; B_m]\!] \Rightarrow C$

Substitution:
$\sigma(A) == \sigma(C)$
(recall: == means "same term as")
New subgoals:
1. $\sigma([\![B_1 ; ... ; B_m]\!] \Rightarrow A_1)$
...
n. $\sigma([\![B_1 ; ... ; B_m]\!] \Rightarrow A_n)$

Command:
apply (rule <RuleName>)

Exercises

Exercise 2: Prove the following in HOL. Pen and paper is fine. If you use Isabelle, use only basic HOL rules corresponding to rules given in previous slides – no simplifiers

1. $A \vee (B \vee C) \to (A \vee B) \vee C$
2. $(A \to (B \to C)) \to (A \wedge B) \to C$
3. $A \vee A \to A \wedge A$
4. $A \vee B \to \neg A \to B$
5. $A \wedge (B \vee C) \to (A \wedge B) \vee C$
6. $(A \wedge \neg B) \vee (B \wedge \neg A) = (A = \neg B)$
7. $\neg(A \wedge B) \to (\neg A) \vee (\neg B)$

Elimination Rules in Isabelle

Tactic erule assumes that first rule premise is assumption to be eliminated:
apply (erule <RuleName>):

Example:
Rule: $[\![?P \wedge ?Q ; [\![?P; ?Q]\!] \Rightarrow ?R]\!] \Rightarrow ?R$
Subgoal: $[\![X ; A \wedge B ; Y]\!] \Rightarrow Z$
Unifier: ?R == Z, ?P == A, ?Q == B
New subgoal: $[\![X; Y]\!] \Rightarrow [\![A; B]\!] \Rightarrow Z$
Same as: $[\![X; Y; A; B]\!] \Rightarrow Z$

Safe and Unsafe Rules

Recall: Rules applied bottom up

Safe rules: Provability is preserved (in bottom up direction)

Examples: $\wedge$I, $\to$I, $\neg$I, iffI, refl, ccontr, classical, $\wedge$E, $\vee$E

Unsafe rules: Can turn provable goal into unprovable one:

Examples: $\vee$I1, $\vee$I2, $\to$E, iffD1, iffD2, $\neg$E

⇒ vs. →

Theorems should be written as

$⟧ A_1 ; ... ; A_n ⟧ ⇒ A$

Not as

$A_1 \wedge ... \wedge A_n \to A$

Exception: Induction variable must not occur in premises

Example:

⟦ A; B(x)⟧ ⇒ C(x), not good

Use instead: A ⇒ B(x) → C(x)

Predicate Logic - Parameters

Subgoal:

1. $\bigwedge x_1 ... x_n$. *Formula*

The x_i are parameters of the subgoal

Intuition: Local constants, arbitrary, fixed values

Rules automatically lifted over $\bigwedge x_1 ... x_n$ and applied directly to *Formula*

Scope

Scope of parameters: Whole subgoal

Scope of HOL connectives:

Never extend to meta-level

I.e. ends with ; or ⇒

⋀ x y. ⟦ ∀y. P y → Q z y; Q x y⟧ ⇒ ∃x. Q x y

means

$\bigwedge$ x y.⟦ ($\forall y_1$. P y_1 → Q z y_1); Q x y⟧ ⇒ $\exists x_1$. Q x_1 y

Natural Deduction, Predicate Logic

$$\frac{\bigwedge x.(P\ x)}{\forall x.(P\ x)}\ \forall I \qquad \frac{\forall x.(P\ x) \quad (P\ ?x) \Rightarrow R}{R}\ \forall E$$

$$\frac{(P\ ?x)}{\exists x.(P\ x)}\ \exists I \qquad \frac{\exists\, x.(P\ x) \quad \bigwedge x.(P\ x) \Rightarrow R}{R}\ \exists E$$

- ∀I and ∃E introduce new parameters (⋀x)
- ∃I and ∀E introduce new unknowns (?x)

Instantiating Rules

apply (rule_tac x = t in <rule>)

Acts as <rule>, but ?x in <rule> is instantiated to t before application

erule_tac is similar

So: x is in <rule>, not in the goal

Two Successful Proofs

1. ∀x. ∃y. x = y

apply (rule ∀I)

1. ⋀x. ∃y. x = y

Best practice	*Exploration*
apply (rule_tac x = "x" in ∃I)	**apply** (rule ∃I)
1. ⋀x. x = x	1. ⋀x. x = ?y x
apply (rule refl)	**apply** (rule refl)
	?y ↦ λz.z
Simpler and clearer	*Shorter and trickier*

Two Unsuccessful Proofs

1. ∃ y. ∀ x. x = y

apply (rule tac x = ??? in ∃I)
???

apply (rule ∃I)
1. ∀x. x = ?y
apply (rule ∀I)
1. ∧ x. x = ?y
apply (rule refl)
?y ↦ x yields ∧x'. x' = x
???

Safe and Unsafe Rules

Safe: ∀I, ∃E
Unsafe: ∀E, ∃I
Create parameters first, unknowns later

Exercises, Predicate Logic

Exercise 3. Prove or disprove the following formulas. If you prove the formulas, use Isabelle, as in exercise 2. For a disproof it is sufficient to show that the formulas are false in ordinary first-order logic.

1. ∀x.∀y. R x y = ∀y.∀x. R x y
2. (∃x. P x) ∨ (∃y. Q y) = ∃z. (P z) ∨ (Q z)
3. ¬ ∀x. P x ⇒ ∃y.¬(P y)
4. ∃x.(P x → ∀y.P y)

Renaming Parameters

Careful with Isabelle-generated names

1. ∀ x. ∃ y. x = y

apply (rule ∀I)
1. ∧x. ∃y. x = y

apply (rule tac x = ”x” in ∃I)

What if the above used in context which already knows some x? Instead:

apply (rename tac xxx)
1. ∧xxx. ∃y. x = y

apply (rule tac x = ”xxx” in ∃I)

Forward Proof

”Forward” rule:	$A_1 \Rightarrow A$
Subgoal:	1. $⟦ B_1 ; ... ; B_m ⟧ \Rightarrow C$
Substitution:	$\sigma(B_i) == \sigma(A_1)$
New subgoal:	1. $\sigma(⟦ B_1 ; ... ; B_n ; A ⟧ \Rightarrow C)$

Command:
apply (frule <rule>)
Like frule but deletes B_i:
apply (drule <rule>)