



COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein, June Andronick, Toby Murray, Rafal Kolanski

HOL

Slide 1



Last time...

- natural deduction rules for \wedge , \vee , \longrightarrow , \neg , iff...
- proof by assumption, by intro rule, elim rule
- safe and unsafe rules

Slide 2



Content

- Intro & motivation, getting started [1]
- Foundations & Principles
 - Lambda Calculus, natural deduction [1,2]
 - Higher Order Logic [3]
 - Term rewriting [4^a]
- Proof & Specification Techniques
 - Inductively defined sets, rule induction [5]
 - Datatypes, recursion, induction [6^b, 7]
 - Code generation, type classes [7]
 - Hoare logic, proofs about programs, refinement [8,9^c,10^d]
 - Isar, locales [11,12]

^aa1 due; ^ba2 due; ^csession break; ^da3 due

Slide 3



QUANTIFIERS

Slide 4

Scope

- Scope of parameters: whole subgoal
- Scope of \forall, \exists, \dots : ends with ; or \implies

Example:

$$\wedge x y. [\forall y. P y \longrightarrow Q z y; Q x y] \implies \exists x. Q x y$$

means

$$\wedge x y. [(\forall y_1. P y_1 \longrightarrow Q z y_1); Q x y] \implies (\exists x_1. Q x_1 y)$$

Slide 5



Natural deduction for quantifiers

$$\frac{\wedge x. P x}{\forall x. P x} \text{ allI} \quad \frac{\forall x. P x \quad P ?x \implies R}{R} \text{ allE}$$

$$\frac{P ?x}{\exists x. P x} \text{ exI} \quad \frac{\exists x. P x \quad \wedge x. P x \implies R}{R} \text{ exE}$$

- **allI** and **exE** introduce new parameters ($\wedge x$).
- **allE** and **exI** introduce new unknowns ($?x$).

Slide 6



Instantiating Rules

apply (rule_tac x = "term" in rule)

Like **rule**, but $?x$ in *rule* is instantiated by *term* before application.

Similar: **erule_tac**

! x is in rule, not in goal !

Slide 7



Two Successful Proofs

1. $\forall x. \exists y. x = y$

apply (rule allI)

1. $\wedge x. \exists y. x = y$

best practice

apply (rule_tac x = "x" in exI)

1. $\wedge x. x = x$

apply (rule refl)

simpler & clearer

exploration

apply (rule exI)

1. $\wedge x. x = ?y x$

apply (rule refl)

$?y \mapsto \lambda u. u$

shorter & trickier

Slide 8



Two Unsuccessful Proofs



1. $\exists y. \forall x. x = y$

apply (rule_tac x = ??? in exI)

apply (rule exI)

1. $\forall x. x = ?y$

apply (rule allI)

1. $\wedge x. x = ?y$

apply (rule refl)

?y \mapsto x yields $\wedge x'. x' = x$

Principle:

?f $x_1 \dots x_n$ can only be replaced by term t

if $params(t) \subseteq x_1, \dots, x_n$

Slide 9

Safe and Unsafe Rules



Safe allI, exE

Unsafe allE, exI

Create parameters first, unknowns later

Slide 10

DEMO: QUANTIFIER PROOFS



Slide 11

Parameter names



Parameter names are chosen by Isabelle

1. $\forall x. \exists y. x = y$

apply (rule allI)

1. $\wedge x. \exists y. x = y$

apply (rule_tac x = "x" in exI)

Brittle!

Slide 12

Renaming parameters



1. $\forall x. \exists y. x = y$

apply (rule allI)

1. $\wedge x. \exists y. x = y$

apply (rename_tac N)

1. $\wedge N. \exists y. N = y$

apply (rule_tac x = "N" in exI)

In general:

(**rename_tac** $x_1 \dots x_n$) renames the rightmost (inner) n parameters to $x_1 \dots x_n$

Slide 13

Examples for Forward Rules



$$\frac{P \wedge Q}{P} \text{ conjunct1} \quad \frac{P \wedge Q}{Q} \text{ conjunct2}$$

$$\frac{P \longrightarrow Q \quad P}{Q} \text{ mp}$$

$$\frac{\forall x. P \ x}{P \ ?x} \text{ spec}$$

Slide 15

Forward Proof: frule and drule



apply (frule < rule >)

Rule: $[A_1; \dots; A_m] \Longrightarrow A$

Subgoal: 1. $[B_1; \dots; B_n] \Longrightarrow C$

Substitution: $\sigma(B_i) \equiv \sigma(A_1)$

New subgoals: 1. $\sigma([B_1; \dots; B_n]) \Longrightarrow A_2$

⋮

m-1. $\sigma([B_1; \dots; B_n]) \Longrightarrow A_m$

m. $\sigma([B_1; \dots; B_n; A]) \Longrightarrow C$

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

Slide 14

Forward Proof: OF



r [OF $r_1 \dots r_n$]

Prove assumption 1 of theorem r with theorem r_1 , and assumption 2 with theorem r_2 , and ...

Rule r $[A_1; \dots; A_m] \Longrightarrow A$

Rule r_1 $[B_1; \dots; B_n] \Longrightarrow B$

Substitution $\sigma(B) \equiv \sigma(A_1)$

r [OF r_1] $\sigma([B_1; \dots; B_n; A_2; \dots; A_m]) \Longrightarrow A$

Slide 16

Forward proofs: THEN

r_1 [THEN r_2] means r_2 [OF r_1]



Slide 17

Hilbert's Epsilon Operator



(David Hilbert, 1862-1943)

$\varepsilon x. Px$ is a value that satisfies P (if such a value exists)

ε also known as **description operator**.

In Isabelle the ε -operator is written `SOME $x. Px$`

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



Slide 19

More Epsilon

ε implies Axiom of Choice:

$$\forall x. \exists y. Q x y \implies \exists f. \forall x. Q x (f x)$$

Existential and universal quantification can be defined with ε .

Isabelle also knows the definite description operator **THE** (aka ι):

$$\frac{}{(\text{THE } x. x = a) = a} \text{ the_eq_trivial}$$



Slide 20

DEMO: FORWARD PROOFS

Slide 18



Some Automation



More Proof Methods:

apply (intro <intro-rules>) repeatedly applies intro rules

apply (elim <elim-rules>) repeatedly applies elim rules

apply clarify applies all safe rules that do not split the goal

apply safe applies all safe rules

apply blast an automatic tableaux prover (works well on predicate logic)

apply fast another automatic search tactic

Slide 21

We have learned so far...



- Proof rules for predicate calculus
- Safe and unsafe rules
- Forward Proof
- The Epsilon Operator
- Some automation

Slide 23



EPSILON AND AUTOMATION DEMO

Slide 22

Assignment



Assignment 1 is out today!

Reminder: **DO NOT COPY**

- Assignments and exams are take-home. This does NOT mean you can work in groups. Each submission is personal.
- For more info, see Plagiarism Policy

Slide 24