COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Toby Murray, June Andronick, Gerwin Klein

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[a]]
  - Term rewriting [4]
- Proof & Specification Techniques
  - Inductively defined sets, rule induction [5]
  - Datatypes, recursion, induction [6, 7]
  - Hoare logic, proofs about programs, C verification [8[b],9]
  - (mid-semester break)
  - Writing Automated Proof Methods [10]
  - Isar, codegen, typeclasses, locales [11[c],12]

[a] a1 due; [b] a2 due; [c] a3 due

Slide 3

Quantifiers

Slide 4

Scope

- Scope of parameters: whole subgoal
- Scope of $\forall, \exists, \ldots$: ends with ; or $\Longrightarrow$

Example:

$$\bigwedge x\ y.\ \llbracket\ \forall y.\ P\ y \longrightarrow Q\ z\ y;\ \ Q\ x\ y\ \rrbracket\ \Longrightarrow\ \exists x.\ Q\ x\ y$$

means

$$\bigwedge x\ y.\ \llbracket\ (\forall y_1.\ P\ y_1 \longrightarrow Q\ z\ y_1);\ \ Q\ x\ y\ \rrbracket\ \Longrightarrow\ (\exists x_1.\ Q\ x_1\ y)$$

Slide 5

Natural deduction for quantifiers

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

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

- **allI** and **exE** introduce new parameters ($\bigwedge x$).
- **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. $\bigwedge x.\ \exists y.\ x = y$

best practice	exploration
apply (rule_tac x = "x" in exI)	**apply** (rule exI)
1. $\bigwedge x.\ x = x$	1. $\bigwedge x.\ x = ?y\ x$
apply (rule refl)	**apply** (rule refl)
	$?y \mapsto \lambda u.u$
simpler & clearer	**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. $\bigwedge x.\ x = ?y$

apply (rule refl)

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

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. $\bigwedge x.\ \exists y.\ x = y$

apply (rule_tac x = "x" in exI)

Brittle!

Slide 12

Renaming parameters

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

apply (rule allI)

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

apply (rename_tac N)

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

apply (rule_tac x = "N" in exI)

In general:
(rename_tac $x_1 \ldots x_n$) renames the rightmost (inner) n parameters to $x_1 \ldots x_n$

Slide 13

Forward Proof: frule and drule

apply (frule $< rule >$)

Rule:	$[\![A_1; \ldots; A_m]\!] \Longrightarrow A$
Subgoal:	1. $[\![B_1; \ldots; B_n]\!] \Longrightarrow C$
Substitution:	$\sigma(B_i) \equiv \sigma(A_1)$
New subgoals:	1. $\sigma([\![B_1; \ldots; B_n]\!] \Longrightarrow A_2)$
	$\vdots$
	m-1. $\sigma([\![B_1; \ldots; B_n]\!] \Longrightarrow A_m)$
	m. $\sigma([\![B_1; \ldots; B_n; A]\!] \Longrightarrow C)$

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

Slide 14

Examples for Forward Rules

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

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

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

Slide 15

Forward Proof: OF

r [**OF** $r_1 \ldots r_n$]

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

Rule r	$[\![A_1; \ldots; A_m]\!] \Longrightarrow A$
Rule r_1	$[\![B_1; \ldots; B_n]\!] \Longrightarrow B$
Substitution	$\sigma(B) \equiv \sigma(A_1)$
r [OF r_1]	$\sigma([\![B_1; \ldots; B_n; A_2; \ldots; A_m]\!] \Longrightarrow A)$

Slide 16

Forward proofs: THEN

NICTA

r_1 [THEN r_2] means r_2 [OF r_1]

Slide 17

NICTA

DEMO: FORWARD PROOFS

Slide 18

Hilbert's Epsilon Operator

NICTA

(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 $x.\ P\ x$

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

Slide 19

More Epsilon

NICTA

$\mathcal{E}$ implies Axiom of Choice:

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

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

Copyright NICTA 2014, provided under Creative Commons Attribution License

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

Epsilon and Automation Demo

Slide 22

We have learned so far...

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

Slide 23

Assignment

Assignement 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