DATA 61

COMP4161: Advanced Topics in Software Verification

$\lambda^{\rightarrow}$ and HOL

Gerwin Klein, Johannes Åman Pohjola, Christine Rizkallah, Miki Tanaka
T3/2020

data61.csiro.au

CSIRO

Last time...

- Simply typed lambda calculus: $\lambda^{\rightarrow}$
- Typing rules for $\lambda^{\rightarrow}$, type variables, type contexts
- β-reduction in $\lambda^{\rightarrow}$ satisfies subject reduction
- β-reduction in $\lambda^{\rightarrow}$ always terminates
- Types and terms in Isabelle

Content

➜ Foundations & Principles

- Intro, Lambda calculus, natural deduction [1,2]
- Higher Order Logic, Isar (part 1) [2,3[a]]
- Term rewriting [3,4]

➜ Proof & Specification Techniques

- Inductively defined sets, rule induction, datatype induction, primitive recursion [4,5]
- General recursive functions, termination proofs [7[b]]
- Proof automation, Hoare logic, proofs about programs, invariants [8]
- C verification [9,10]
- Practice, questions, examp prep [10[c]]

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

Preview: Proofs in Isabelle

Proofs in Isabelle

General schema:

lemma name: "<goal>"
apply <method>
apply <method>
...
done

➜ Sequential application of methods until all **subgoals** are solved.

The Proof State

1. $\bigwedge x_1 \dots x_p . [\![A_1; \dots ; A_n]\!] \Longrightarrow B$
2. $\bigwedge y_1 \dots y_q . [\![C_1; \dots ; C_m]\!] \Longrightarrow D$

$x_1 \dots x_p$	Parameters
$A_1 \dots A_n$	Local assumptions
B	Actual (sub)goal

Isabelle Theories

Syntax:

`theory` *MyTh*
`imports` $ImpTh_1 \ldots ImpTh_n$
`begin`
(declarations, definitions, theorems, proofs, ...)*
`end`

- ➜ *MyTh*: name of theory. Must live in file *MyTh*`.thy`
- ➜ $ImpTh_i$: name of *imported* theories. Import transitive.

Unless you need something special:

`theory` *MyTh* `imports Main begin ... end`

Natural Deduction Rules

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

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

$$\frac{A \Longrightarrow B}{A \longrightarrow B}\ \text{impI} \qquad \frac{A \longrightarrow B \quad A \quad B \Longrightarrow C}{C}\ \text{impE}$$

For each connective ($\wedge$, $\vee$, etc):
introduction and **elimination** rules

Proof by assumption

apply assumption

proves

1. $[\![B_1; \ldots; B_m]\!] \Longrightarrow C$

by unifying C with one of the B_i

There may be more than one matching B_i and multiple unifiers.

Backtracking!

Explicit backtracking command: **back**

Intro rules

Intro rules decompose formulae to the right of $\Longrightarrow$.

apply (rule <intro-rule>)

Intro rule $[\![A_1; \ldots; A_n]\!] \Longrightarrow A$ means

- ➜ To prove A it suffices to show $A_1 \ldots A_n$

Applying rule $[\![A_1; \ldots; A_n]\!] \Longrightarrow A$ to subgoal C:

- ➜ unify A and C
- ➜ replace C with n new subgoals $A_1 \ldots A_n$

Elim rules

Elim rules decompose formulae on the left of $\Longrightarrow$.

apply (erule <elim-rule>)

Elim rule $[\![A_1; \ldots; A_n]\!] \Longrightarrow A$ means

- ➜ If I know A_1 and want to prove A it suffices to show $A_2 \ldots A_n$

Applying rule $[\![A_1; \ldots; A_n]\!] \Longrightarrow A$ to subgoal C:
Like **rule** but also

- ➜ unifies first premise of rule with an assumption
- ➜ eliminates that assumption

Demo

More Proof Rules

Iff, Negation, True and False

$$\frac{A \Longrightarrow B \quad B \Longrightarrow A}{A = B} \text{ iffI} \qquad \frac{A = B \quad [\![A \longrightarrow B; B \longrightarrow A]\!] \Longrightarrow C}{C} \text{ iffE}$$

$$\frac{A = B}{A \Longrightarrow B} \text{ iffD1} \qquad \frac{A = B}{B \Longrightarrow A} \text{ iffD2}$$

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

$$\frac{}{\mathit{True}} \text{ TrueI} \qquad \frac{\mathit{False}}{P} \text{ FalseE}$$

Equality

$$\frac{}{t = t}\ \text{refl} \qquad \frac{s = t}{t = s}\ \text{sym} \qquad \frac{r = s \quad s = t}{r = t}\ \text{trans}$$

$$\frac{s = t \quad P\ s}{P\ t}\ \text{subst}$$

Rarely needed explicitly — used implicitly by term rewriting

Classical

$$\overline{P = True \vee P = False} \text{ True-or-False}$$

$$\overline{P \vee \neg P} \text{ excluded-middle}$$

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

- **excluded-middle**, **ccontr** and **classical** not derivable from the other rules.
- if we include True-or-False, they are derivable

They make the logic "classical", "non-constructive"

Cases

$$\frac{}{P \vee \neg P} \text{ excluded-middle}$$

is a case distinction on type *bool*

Isabelle can do case distinctions on arbitrary terms:

apply (case_tac *term*)

Safe and not so safe

Safe rules preserve provability

conjI, impI, notI, iffI, refl, ccontr, classical, conjE, disjE

$$\frac{A \quad B}{A \wedge B} \text{ conjI}$$

Unsafe rules can turn a provable goal into an unprovable one

disjI1, disjI2, impE, iffD1, iffD2, notE

$$\frac{A}{A \vee B} \text{ disjI1}$$

Apply safe rules before unsafe ones

Demo

What we have learned so far...

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

- indent your proofs! (one space per subgoal)
- prefer implicit backtracking (chaining) or *rule_tac*, instead of *back*
- *prefer* and *defer*
- *oops* and *sorry*

