COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein, June Andronick, Toby Murray

\[ \lambda \rightarrow \text{and HOL} \]
Exercises from last time

- Construct a type derivation tree for the term \( \lambda x \ y \ z. \ z \ x \ (y \ x) \)
- Find a unifier (substitution) such that \( \lambda x \ y \ z. \ ?F \ y \ z = \lambda x \ y \ z. \ z \ (?G \ x \ y) \)
Content

→ Intro & motivation, getting started [1]

→ Foundations & Principles
  • Lambda Calculus, natural deduction [2,3,4a]
  • Higher Order Logic [5,6b,7]
  • Term rewriting [8,9,10c]

→ Proof & Specification Techniques
  • Isar [11,12d]
  • Inductively defined sets, rule induction [13e,15]
  • Datatypes, recursion, induction [16,17f,18,19]
  • Calculational reasoning, mathematics style proofs [20]
  • Hoare logic, proofs about programs [21g,22,23]

a a1 out; b a1 due; c a2 out; d a2 due; e session break; f a3 out; g a3 due
PREVIEW: PROOFS IN ISABELLE
General schema:

```isar
text
```

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

→ Sequential application of methods until all subgoals are solved.
The Proof State

1. $\wedge x_1 \ldots x_p. [A_1; \ldots; A_n] \Rightarrow B$

2. $\wedge y_1 \ldots y_q. [C_1; \ldots; C_m] \Rightarrow D$

$x_1 \ldots x_p$ Parameters

$A_1 \ldots A_n$ Local assumptions

$B$ Actual (sub)goal
Isabelle Theories

Syntax:

theory \textit{MyTh} \\
imports \textit{ImpTh}_1 \ldots \textit{ImpTh}_n \\
begin \\
(declarations, definitions, theorems, proofs, ...)* \\
end

\rightarrow \textit{MyTh}: name of theory. Must live in file \textit{MyTh.thy} \\\n\rightarrow \textit{ImpTh}_i: name of imported theories. Import transitive.

Unless you need something special:

theory \textit{MyTh} imports Main begin ... end
Natural Deduction Rules

\begin{align*}
\frac{A \quad B}{A \land B} & \text{ conjI} \\
\frac{A}{A \lor B} \quad \frac{B}{A \lor B} & \text{ disjI1/2} \\
\frac{A \rightarrow B}{A \rightarrow B} & \text{ impl} \\
\frac{A \land B \quad [A; B] \implies C}{C} & \text{ conjE} \\
\frac{A \lor B \quad A \implies C \quad B \implies C}{C} & \text{ disjE} \\
\frac{A \implies B \quad A \quad B \implies C}{C} & \text{ impE}
\end{align*}

For each connective (\(\land, \lor\), etc):

**introduction** and **elimination** rules
Proof by assumption

apply assumption

proves

1. \([B_1; \ldots; B_m] \Rightarrow C\)

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 decompose formulae to the right of $\implies$.

**apply** (rule $\langle\text{intro-rule}\rangle$)

Intro rule $\left[ A_1; \ldots; A_n \right] \implies A$ means

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

Applying rule $\left[ A_1; \ldots; A_n \right] \implies A$ to subgoal $C$:

$\implies$ unify $A$ and $C$

$\implies$ replace $C$ with $n$ new subgoals $A_1 \ldots A_n$
Elim rules decompose formulae on the left of \( \implies \).

**apply** (erule \(<\text{elim-rule}\>)

Elim rule \([A_1; \ldots; A_n] \implies A\) means

\(\implies\) If I know \(A_1\) and want to prove \(A\) it suffices to show \(A_2 \ldots A_n\)

Applying rule \([A_1; \ldots; A_n] \implies A\) to subgoal \(C\):

Like **rule** but also

\(\implies\) unifies first premise of rule with an assumption

\(\implies\) eliminates that assumption
DEMO
MORE PROOF RULES
Iff, Negation, True and False

\[
\begin{align*}
A \implies B & \quad B \implies A \quad \text{iffI} \\
A = B & \quad \text{iffE} \\
A = B & \quad \text{iffD1} \\
A \implies B & \quad \text{iffD2} \\
A \implies \text{False} & \quad \text{notI} \\
\neg A & \quad \text{notE} \\
\text{True} & \quad \text{Truel} \\
\text{False} & \quad \text{FalseE}
\end{align*}
\]
Equality

\[
\begin{align*}
\text{refl} & : \quad t = t \\
\text{sym} & : \quad s = t \quad \rightarrow \quad t = s \\
\text{trans} & : \quad r = s \quad s = t \quad \rightarrow \quad r = t \\
\text{subst} & : \quad s = t \quad P \quad s \quad \rightarrow \quad P \quad t
\end{align*}
\]

Rarely needed explicitly — used implicitly by term rewriting
Classical

\[ P = \text{True} \lor P = \text{False} \]

True-False

\[ P \lor \neg P \]

excluded-middle

\[ \neg A \implies \text{False} \]
\[ A \]

\[ \neg A \implies A \]

\[ A \]

ccontr

\[ \neg A \implies A \]

classical

⇒ excluded-middle, ccontr and classical

not derivable from the other rules.

⇒ if we include True-False, they are derivable

They make the logic “classical”, “non-constructive”
$P \lor \neg P$ excluded-middle

is a case distinction on type $bool$

Isabelle can do case distinctions on arbitrary terms:

\textbf{apply} (case\_tac \textit{term})
Safe and not so safe

**Safe rules** preserve provability

- conjI, impI, notI, iffi, refl, ccontr, classical, conjE, disjE

\[
\frac{A \quad B}{A \land B} \quad \text{conjI}
\]

**Unsafe rules** can turn a provable goal into an unprovable one

- disjI1, disjI2, impE, iffD1, iffD2, notE

\[
\frac{A}{A \lor B} \quad \text{disjI1}
\]

Apply safe rules before unsafe ones
DEMO
What we have learned so far...

- natural deduction rules for $\land$, $\lor$, $\rightarrow$, $\neg$, iff...
- proof by assumption, by intro rule, elim rule
- safe and unsafe rules
Exercises

➜ Redo the demo alone + exercises
➜ Assignment 1 is out today!

➜ Reminder: DO NOT CHEAT

- 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