COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Simon Winwood, Toby Murray, June Andronick, Gerwin Klein

Slide 1

Content

- Intro & motivation, getting started with Isabelle
- Foundations & Principles
  - Lambda Calculus
  - Higher Order Logic, natural deduction
  - Term rewriting
- Proof & Specification Techniques
  - Inductively defined sets, rule induction
  - Datatypes, recursion, induction
  - Well founded recursion, Calculational reasoning
  - Hoare logic, proofs about programs
  - Locales, Presentation

Slide 2

Last Time

- Sets in Isabelle
- Inductive Definitions
- Rule induction
- Fixpoints

Slide 3

Exercises

Formalize the last lecture in Isabelle:
- Define \( \text{closed} f \ A \mapsto (\alpha \text{ set} \Rightarrow \alpha \text{ set} \Rightarrow \text{bool} \)
- Show \( \text{closed} f \ A \land \text{closed} f \ B \Rightarrow \text{closed} f (A \cap B) \) if \( f \) is monotone (\( \text{mono} \) is predefined)
- Define \( \text{lfpt} f \) as the intersection of all \( f \)-closed sets
- Show that \( \text{lfpt} f \) is a fixpoint of \( f \) if \( f \) is monotone
- Show that \( \text{lfpt} f \) is the least fixpoint of \( f \)
- Declare a constant \( R : (\alpha \text{ set} \times \alpha \text{ set}) \)
- Define \( \hat{R} : \alpha \text{ set} \Rightarrow \alpha \text{ set} \) in terms of \( R \)
- Show soundness of rule induction using \( R \) and \( \text{lfpt} \hat{R} \)

Slide 4
Inductive definition in Isabelle

\begin{verbatim}
inductive X :: α ⇒ bool
where
  rule₁: "[X s₁ A] ⇒ X s₁′" 
  ... 
| ruleₙ: ... 
\end{verbatim}

Rule induction

\begin{verbatim}
show "X x ⇒ P x"
proof (induct rule: X.induct)
  fix s and s' 
  assume "X s" and "A" and "P s"
  ... 
  show "P s'"
next
| 
qed
\end{verbatim}

Abbreviations

\begin{verbatim}
show "X x ⇒ P x"
proof (induct rule: X.induct)
  case rule₁ 
  ... 
  show ?case
next 
| 
next
  case ruleₙ 
  ... 
  show ?case
qed
\end{verbatim}
Implicit selection of induction rule

\[
\text{assume A: } "X \, x" \\
\text{show } "P \, x" \\
\text{using A proof induct} \\
\text{qed}
\]

A remark on style

- case (rule, \(x\) \(y\)) \ldots show \(?\)case is easy to write and maintain
- fix \(x\) \(y\) assume \(f\)ormula \ldots show \(f\)ormula' is easier to read:
  - all information is shown locally
  - no contextual references (e.g. \(?\)case)

Renaming free variables in rule

\[
\text{case (rule, } x_1 \ldots x_k) \\
\text{Renames first } k \text{ variables in rule, to } x_1 \ldots x_k.
\]

We have seen so far ...

- Formalising inductive sets and rule induction
- Rule induction in Isar
- Implicit induction rule selection
- Case abbreviations
- Renaming case variables