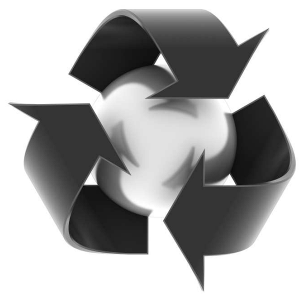
COMP 4161

NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein
Formal Methods



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

LAST TIME

➜ Sets in Isabelle

LAST TIME

- ➜ Sets in Isabelle
- ➜ Inductive Definitions

LAST TIME

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

LAST TIME

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

Exercises

Formalize the last lecture in Isabelle:

- Define **closed** $f\ A :: (\alpha\ \text{set} \Rightarrow \alpha\ \text{set}) \Rightarrow \alpha\ \text{set} \Rightarrow \text{bool}$
- Show closed $f\ A \wedge$ closed $f\ B \Longrightarrow$ closed $f\ (A \cap B)$ if f is monotone (**mono** is predefined)
- Define **lfpt** f as the intersection of all f-closed sets
- Show that lfpt f is a fixpoint of f if f is monotone
- Show that 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 lfpt $\hat{R}$

Rule Induction in Isar

INDUCTIVE DEFINITION IN ISABELLE

inductive $X :: \alpha \Rightarrow$ bool

where

rule$_1$: "$[\![X\ s; A]\!] \Longrightarrow X\ s'$"

$\vdots$

| rule$_n$: $\ldots$

show "$X\ x \Longrightarrow P\ x$"

proof (induct rule: X.induct)

  fix s and s' **assume** "$X\ s$" and "A" and "$P\ s$"

  ...

  show "$P\ s'$"

next

⋮

qed

Abbreviations

show "$X\ x \Longrightarrow P\ x$"

proof (induct rule: X.induct)

  case rule$_1$

  . . .

  show ?case

next

⋮

next

  case rule$_n$

  . . .

  show ?case

qed

IMPLICIT SELECTION OF INDUCTION RULE

assume A: "$X\ x$"

⋮

show "$P\ x$"

using A **proof** induct

⋮

qed

NICTA

assume A: "$X\ x$"

⋮

show "$P\ x$"

using A **proof** induct

⋮

qed

lemma assumes A: "$X\ x$" **shows** "$P\ x$"

using A **proof** induct

⋮

qed

Renaming free variables in rule

case (rule$_i$ $x_1 \ldots x_k$)

Renames first k variables in rule$_i$ to $x_1 \ldots x_k$.

A REMARK ON STYLE

- **case** (rule$_i$ x y) . . . **show** ?case
  is easy to write and maintain

A REMARK ON STYLE

- ➜ **case** (rule$_i$ x y) ... **show** ?case
  is easy to write and maintain

- ➜ **fix** x y **assume** $formula$ ... **show** $formula'$
  is easier to read:
  - all information is shown locally
  - no contextual references (e.g. ?case)

We have seen so far ...

➜ Formalising inductive sets and rule induction

WE HAVE SEEN SO FAR ...

➜ Formalising inductive sets and rule induction

➜ Rule induction in Isar

WE HAVE SEEN SO FAR ...

- Formalising inductive sets and rule induction
- Rule induction in Isar
- Implicit induction rule selection

We have seen so far ...

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

We have seen so far ...

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