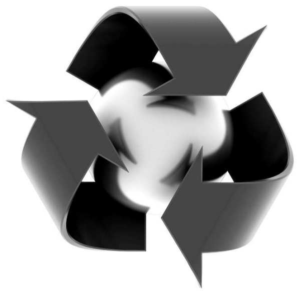
COMP 4161

NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein, June Andronick, Toby Murray, Rafal Kolanski



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

- Isar [5]
- Inductively defined sets, rule induction [6 ^{b}]
- Datatypes, recursion, induction [7 ^{c}, 8]
- Calculational reasoning, code generation [9]
- Hoare logic, proofs about programs [10 ^{d},11,12]

^{a}a1 due; ^{b}a2 due; ^{c}session break; ^{d}a3 due

Datatypes in Isar

Datatype case distinction

proof (cases $term$)

  case Constructor$_1$

  ⋮

next

⋮

next

  case (Constructor$_k$ $\vec{x}$)

  $\cdots$ $\vec{x}$ $\cdots$

qed

case (Constructor$_i$ $\vec{x}$) $\equiv$

fix $\vec{x}$ **assume** Constructor$_i$: "$term =$ Constructor$_i$ $\vec{x}$"

Structural induction for type nat

show $P\ n$

proof (induct n)

case 0 $\qquad\equiv\qquad$ **let** $?case$ = $P\ 0$

$\ldots$

show $?case$

next

case (Suc n) $\qquad\equiv\qquad$ **fix** n **assume** Suc: $P\ n$

$\ldots$ $\qquad\qquad$ **let** $?case$ = P (Suc n)

$\ldots\ n\ \ldots$

show $?case$

qed

Structural induction with $\Longrightarrow$ and $\bigwedge$

NICTA

show "$\bigwedge x.\ A\ n \Longrightarrow P\ n$"

proof (induct n)

  case 0 $\equiv$ **fix** x **assume** 0: "$A\ 0$"

  ... **let** $?case$ = "$P\ 0$"

  show $?case$

next

  case (Suc n) $\equiv$ **fix** n and x

  ... **assume** Suc: "$\bigwedge x.\ A\ n \Longrightarrow P\ n$"

  $\cdots\ n\ \cdots$ "A (Suc n)"

  ... **let** $?case$ = "P (Suc n)"

  show $?case$

qed

Demo: Datatypes in Isar

DEMO: REGULAR EXPRESSIONS

We have seen today ...

- ➜ Datatypes in Isar
- ➜ Defining regular wxpressions as a data type
- ➜ Playing with recursion and induction