COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein, June Andronick, Toby Murray, Rafal Kolanski

Slide 1

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

Slide 2

DATATYPES IN ISAR

Slide 3

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$_i$ $\vec{x}$”

Slide 4

Structural induction for type nat

```
show P n
proof (induct n)
   case 0          ≡  let ?case = P 0
   ...
   show ?case
next
   case (Suc n)    ≡  fix n assume Suc: P n
   ...                let ?case = P (Suc n)
   ... n ...
   show ?case
qed
```

Slide 5

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

```
show "⋀x. A n ⟹ P n"
proof (induct n)
   case 0                 ≡  fix x assume 0: "A 0"
   ...                       let ?case = "P 0"
   show ?case
next
   case (Suc n)           ≡  fix n and x
   ...                       assume Suc: "⋀x. A n ⟹ P n"
   ... n ...                             "A (Suc n)"
   ...                       let ?case = "P (Suc n)"
   show ?case
qed
```

Slide 6

Demo: Datatypes in Isar

Slide 7

Demo: Regular Expressions

Slide 8

We have seen today ...

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

Slide 9