COMP 4161
NICTA Advanced Course

Advanced Topics in Software Verification

Simon Winwood, Toby Murray, June Andronick, Gerwin Klein

wf_rec

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
  - **More recursion, Calculational reasoning**
  - Hoare logic, proofs about programs
  - Locales, Presentation

Datatypes in Isar

Datatype case distinction

```
proof (cases $term$)
  case Constructor$_1$
  ⋮
next
⋮
next
  case (Constructor$_k$ $\vec{x}$)
  ⋯ $\vec{x}$ ⋯
qed
```

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

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

Structural induction for type nat

NICTA

```
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 ⟹ and ⋀

NICTA

```
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

NICTA

DEMO

Slide 7