

**COMP 4161**

NICTA Advanced Course

**Advanced Topics in Software Verification**

Gerwin Klein

Formal Methods

**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, Computational reasoning**
  - Hoare logic, proofs about programs
  - Locales, Presentation

# DATATYPES IN ISAR

# DATATYPE CASE DISTINCTION

```

proof (cases term)
  case Constructor1
  ⋮
next
  ⋮
next
  case (Constructork  $\vec{x}$ )
  ...  $\vec{x}$  ...
qed
  
```

**case** (Constructor<sub>*i*</sub>  $\vec{x}$ )  $\equiv$

**fix**  $\vec{x}$  **assume** Constructor<sub>*i*</sub> : "*term* = Constructor<sub>*i*</sub>  $\vec{x}$ "

# STRUCTURAL INDUCTION FOR TYPE NAT

**show**  $P\ n$

**proof** (induct  $n$ )

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

...

**show**  $?case$

**next**

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

...

**let**  $?case = P\ (\text{Suc } n)$

...  $n$  ...

**show**  $?case$

**qed**

# STRUCTURAL INDUCTION WITH $\implies$ AND $\wedge$

**show** " $\wedge x. A\ n \implies P\ n$ "

**proof** (induct  $n$ )

**case** 0

...

**show**  $?case$

**next**

**case** (Suc  $n$ )

...

...  $n$  ...

...

**show**  $?case$

**qed**

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

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

$\equiv$  **fix**  $n$  and  $x$

**assume** Suc: " $\wedge x. A\ n \implies P\ n$ "

" $A\ (\text{Suc } n)$ "

**let**  $?case = "P\ (\text{Suc } n)"$

**DEMO**