When automation fails

Split proof up into smaller steps.

Or explore by **apply**:
When automation fails

Split proof up into smaller steps.

Or explore by **apply**:

```
  have ... using ...
```

```
  apply -
  apply auto
```

---

When automation fails

Split proof up into smaller steps.

Or explore by **apply**:

```
  have ... using ...
  apply -
```

```
  apply auto
```

```
  apply ...
```

At the end:
When automation fails

Split proof up into smaller steps.

Or explore by **apply**:

* `have . . . using . . .`
* `apply -`
  * to make incoming facts
  * part of proof state
  * or whatever
* `apply auto`
* `apply . . .`

At the end:

* **done**

---

**Local lemmas**

**10 Streamlining Proofs**

Pattern Matching and Quotations

Top down proof development

Local lemmas

```
**have** \( B \) **if** name: \( A_1 \ldots A_m \) **for** \( x_1 \ldots x_n \)
```
Local lemmas

\textbf{have} \: B \textbf{if name:} \: A_1 \ldots A_m \textbf{ for } x_1 \ldots x_n \textbf{ proves } \left[ A_1; \ldots ; A_m \right] \implies B

Local lemmas

\textbf{have} \: B \textbf{ if name:} \: A_1 \ldots A_m \textbf{ for } x_1 \ldots x_n \textbf{ proves } \left[ A_1; \ldots ; A_m \right] \implies B \textbf{ where all } x_i \textbf{ have been replaced by } ?x_i.

Proof state and Isar text

In general: \textbf{proof method}

Proof state and Isar text
In general: \textbf{proof method} Apply method and generates subgoal(s):
\[ \wedge x_1 \ldots x_n. \ [A_1; \ldots; A_m] \Rightarrow B \]

How to prove each subgoal:
\begin{itemize}
  \item \texttt{fix} \ x_1 \ldots x_n
  \item \texttt{assume} \ A_1 \ldots A_m
  \item \vdots
  \item \texttt{show} \ B
\end{itemize}
Isar by example

Proof patterns

Streamlining Proofs

Proof by Cases and Induction

Isar_Induction_Demo.thy

Proof by cases

Datatype case analysis

datatype \( t = C_1 ~ \cdots ~ \) ...

proof (cases "term")
\[
\begin{align*}
\text{case } (C_1 ~ x_1 ~ \cdots ~ x_k) \\
\quad \cdots x_j ~ \cdots \\
\text{next} \\
\vdots \\
\text{qed}
\end{align*}
\]

where case \((C_i ~ x_1 ~ \cdots ~ x_k)\) \(\equiv\)

fix \(x_1 ~ \cdots ~ x_k\)
assume \(C_i; \begin{array}{c}
\text{label} \quad \text{term} = (C_i ~ x_1 ~ \cdots ~ x_k) \\
\end{array}\)
Isar_Induction_Demo.thy

Structural induction for \textit{nat}

show $P(n)$
proof (induction $n$)
  case 0  ≡  let ?case = $P(0)$
  ...
  show ?case
next
  case (Suc $n$)
  ...
  ...
  show ?case
qed

show $P(n)$
proof (induction $n$)
  case 0  ≡  let ?case = $P(0)$
  ...
  show ?case
next
  case (Suc $n$)
  ...
  ...
  show ?case
qed
Structural induction for \textit{nat}

\begin{align*}
\text{show } P(n) \\
\text{proof } (\text{induction } n) \\
\text{case } 0 & \equiv \text{ let } ?\text{case} = P(0) \\
\text{case } (\text{Suc } n) & \equiv \text{ let } ?\text{case} = P(\text{Suc } n) \\
\text{case } (\text{ \ldots } \text{ case } (\text{Suc } n) \ldots ) & \equiv \text{ let } ?\text{case} = P(\text{Suc } n) \\
\text{qed}
\end{align*}

Structural induction with $\implies$

\begin{align*}
\text{show } A(n) \implies P(n) \\
\text{proof } (\text{induction } n) \\
\text{case } 0 & \equiv \text{ assume } 0 : A(0) \\
& \text{ let } ?\text{case} = P(0) \\
\text{case } (\text{Suc } n) & \equiv \text{ fix } n \\
& \text{ assume } \text{Suc. } A(n) \implies P(n) \\
& \text{ assume } A(\text{Suc } n) \\
& \text{ let } ?\text{case} = P(\text{Suc } n) \\
\text{next} & \equiv \text{ show } ?\text{case} \\
\text{qed}
\end{align*}

Named assumptions

In a proof of $A_1 \implies \ldots \implies A_n \implies B$

by structural induction:

In the context of case $C$

we have

$C.IH$ the induction hypotheses

Named assumptions

In a proof of $A_1 \implies \ldots \implies A_n \implies B$

by structural induction:

In the context of case $C$

we have

$C.IH$ the induction hypotheses

$C.prems$ the premises $A_i$
Named assumptions

In a proof of
\[ A_1 \implies \ldots \implies A_n \implies B \]
by structural induction:
In the context of
\textbf{case} \( C \)
we have
- \( C.IH \) the induction hypotheses
- \( C.prems \) the premises \( A_i \)
\[ C \quad C.IH + C.prems \]

Computation induction

If function \( f \) is defined by \textbf{fun} with \( n \) equations:
\textbf{proof}(induction \( s \) \( t \) ... rule: \( f.induct \))

Generates cases named \( i = 1 \ldots n \):
\textbf{case} \((i \; x \; y \; \ldots)\)
Computation induction

If function \emph{f} is defined by \textbf{fun} with \emph{n} equations:

\begin{verbatim}
proof(induction s t ... rule: fi.induct)
\end{verbatim}

Generates cases named \emph{i = 1 \ldots n}:

\begin{verbatim}
case (i x y ...)
\end{verbatim}

Isabelle/jEdit generates Isar template for you!

\begin{verbatim}
\end{verbatim}

Correctness of sorting

\textbf{Specification of} \emph{sort :: ('a::linorder) list ⇒ 'a list}:

\begin{verbatim}
sorted [] = True
sorted (x # xs) = ((\forall y\in set xs. x \leq y) \land sorted xs)
\end{verbatim}
Correctness of sorting

Specification of \( \text{sort} :: (\forall a : \text{linorder}) \; \text{list} \Rightarrow 'a \; \text{list} : \)
\[
\text{sorted} \; (\text{sort} \; xs)
\]

Is that it?

Correctness of sorting

Specification of \( \text{sort} :: (\forall a : \text{linorder}) \; \text{list} \Rightarrow 'a \; \text{list} : \)
\[
\text{sorted} \; (\text{sort} \; xs)
\]

Is that it? How about
\[
\text{set} \; (\text{sort} \; xs) = \text{set} \; xs
\]

Correctness of sorting

Specification of \( \text{sort} :: (\forall a : \text{linorder}) \; \text{list} \Rightarrow 'a \; \text{list} : \)
\[
\text{sorted} \; (\text{sort} \; xs)
\]

Is that it? How about
\[
\text{set} \; (\text{sort} \; xs) = \text{set} \; xs
\]

Better: every \( x \) occurs as often in \( \text{sort} \; xs \) as in \( xs \).

More succinctly:
\[
\text{mset} \; (\text{sort} \; xs) = \text{mset} \; xs
\]

where \( \text{mset} :: 'a \; \text{list} \Rightarrow 'a \; \text{multiset} \)

What are multisets?

Sets with (possibly) repeated elements

Some operations:
\[
\begin{align*}
\{\#\} :: 'a \; \text{multiset} \\
\text{add_mset} :: 'a \Rightarrow 'a \; \text{multiset} \Rightarrow 'a \; \text{multiset} \\
+ :: 'a \; \text{multiset} \Rightarrow 'a \; \text{multiset} \Rightarrow 'a \; \text{multiset}
\end{align*}
\]
What are multisets?

Sets with (possibly) repeated elements

Some operations:

\[
\begin{align*}
\{\#\} & \quad \text{`a multiset} \\
add\_mset & \quad \text{`a } \rightarrow \text{`a multiset } \Rightarrow \text{`a multiset} \\
+ & \quad \text{`a multiset } \Rightarrow \text{`a multiset } \Rightarrow \text{`a multiset} \\
mset & \quad \text{`a list } \Rightarrow \text{`a multiset} \\
set\_mset & \quad \text{`a multiset } \Rightarrow \text{`a set}
\end{align*}
\]

Thys/Sorting.thy

Insertion sort correctness