Structured Induction Proofs in Isabelle/Isar

Makarius

April 2006

Motivation

Introduction

Isabelle/Pure: simple logical framework
(models abstract syntax and primitive inferences)

Isabelle/Isar: framework for human-readable structured proofs
(interprets declarative proof texts in terms of Pure concepts)

Observation: realistic applications routinely use compound inductive predicates, including

- local parameters $\bigwedge x.\ \ldots$
- local premises $A \Longrightarrow \ldots$
- local definitions $x \equiv a\ y$
- simultaneous goals $P\ x\ \&\ Q\ y$

Example: Induction is trivial?

Natural deduction rule:

$$nat\text{-}induct\text{: } P\ 0 \Longrightarrow (\bigwedge n.\ P\ n \Longrightarrow P\ (Suc\ n)) \Longrightarrow P\ n$$

Canonical Isar proof:

lemma
  fixes $n :: nat$
  shows $P\ n$
proof $(rule\ nat\text{-}induct)$
  show $P\ 0$ $\langle proof \rangle$
next
  fix n
  assume $P\ n$
  show $P\ (Suc\ n)$ $\langle proof \rangle$
qed

Example: Induction is non-trivial!

lemma
  fixes $n :: nat$ **and** $x :: {}'a$
  assumes $A\ n\ x$
  shows $P\ n\ x$
proof $-$
  have $\forall\, x.\ A\ n\ x \longrightarrow P\ n\ x$
  proof $(rule\ nat\text{-}induct)$
    show $\forall\, x.\ A\ 0\ x \longrightarrow P\ 0\ x$
    proof
      fix x **show** $A\ 0\ x \longrightarrow P\ 0\ x$
      proof
        assume $A\ 0\ x$
        show $P\ 0\ x$ $\langle proof \rangle$
      qed
    qed
  next

fix n **assume** $raw\text{-}hyp$: $\forall x.\ A\ n\ x \longrightarrow P\ n\ x$
have hyp: $\bigwedge x.\ A\ n\ x \Longrightarrow P\ n\ x$
proof $-$
  fix x **from** $raw\text{-}hyp$ **have** $A\ n\ x \longrightarrow P\ n\ x$ **..**
  also assume $A\ n\ x$
  finally show $P\ n\ x$ **.**
qed
show $\forall x.\ A\ (Suc\ n)\ x \longrightarrow P\ (Suc\ n)\ x$
proof
  fix x **show** $A\ (Suc\ n)\ x \longrightarrow P\ (Suc\ n)\ x$
  proof
    assume $prem$: $A\ (Suc\ n)\ x$
    show $P\ (Suc\ n)\ x$ $\langle proof \rangle$
  qed
qed
qed
then have $A\ n\ x \longrightarrow P\ n\ x$ **..**
also note $\langle A\ n\ x \rangle$
finally show $P\ n\ x$ **.**

qed

Discussion

Anything wrong with Isabelle/Isar?

- Primitive natural deduction exhibits many details.
- Object-level connectives $\forall$, $\longrightarrow$ demand extra work.
- "..., but this can be automated." (Really?)

Other systems:

- Old-style Isabelle tactic scripts often refer to adhoc automation, e.g. $[\mathit{rule\text{-}format}]$, $(\mathit{intro\ strip})$, blast.
- Coq $\mathit{induction}$ seems to be slightly better: full proof context may participate in the induction.

Proper Isar approach:

$\rightarrow$ Natural Induction as specific Isar proof method.

$\rightarrow$ Sane proof structure instead of ad-hoc automation.

Example: Induction is trivial!

lemma
  fixes $n :: nat$ **and** $x :: {}'a$
  assumes $A\ n\ x$
  shows $P\ n\ x$ **using** $\langle A\ n\ x \rangle$
proof $(induct\ n\ fixing\colon x)$
  case 0
  from $\langle A\ 0\ x \rangle$
  show $P\ 0\ x\ \ \langle proof \rangle$
next
  case $(Suc\ n)$
  from $\langle \bigwedge x.\ A\ n\ x \Longrightarrow P\ n\ x \rangle$
    and $\langle A\ (Suc\ n)\ x \rangle$
  show $P\ (Suc\ n)\ x\ \ \langle proof \rangle$
qed

The Isabelle/Isar framework

Pure logic

$\Rightarrow$	function type constructor
$\bigwedge :: (\alpha \Rightarrow prop) \Rightarrow prop$	universal quantifier
$\Longrightarrow :: prop \Rightarrow prop \Rightarrow prop$	implication

$$\frac{\begin{array}{c}[x] \\ \vdots \\ B(x)\end{array}}{\bigwedge x.\ B(x)}\ (\bigwedge I) \qquad \frac{\bigwedge x.\ B(x)}{B(a)}\ (\bigwedge E)$$

$$\frac{\begin{array}{c}[A] \\ \vdots \\ B\end{array}}{A \Longrightarrow B}\ (\Longrightarrow I) \qquad \frac{A \Longrightarrow B \quad A}{B}\ (\Longrightarrow E)$$

$\equiv :: \alpha \Rightarrow \alpha \Rightarrow prop$	equality ($\alpha\beta\eta$-conversion)
$\& :: prop \Rightarrow prop \Rightarrow prop$	ephemeral conjunction

Isar contexts

Idea: elaborate Γ of natural deduction judgments $\Gamma \vdash \varphi$.

{
  fix x
  have $B\ x$ $\langle proof \rangle$
}
note $\langle \bigwedge x.\ B\ x \rangle$

{
  def $x \equiv a$
  have $B\ x$ $\langle proof \rangle$
}
note $\langle B\ a \rangle$

{
  assume A
  have B $\langle proof \rangle$
}
note $\langle A \Longrightarrow B \rangle$

{
  obtain x **where** $A\ x$ $\langle proof \rangle$
  have B $\langle proof \rangle$
}
note $\langle B \rangle$

Abbreviations: **case** $(a\ \vec{x})$ invokes context expression a being defined in the context

Isar proofs

Idea: interpretation of algebraic expressions of facts/goals/rules.

have $A \wedge B$
proof $(rule\ \langle A \Longrightarrow B \Longrightarrow A \wedge B\rangle)$
  show A $\langle proof \rangle$
  show B $\langle proof \rangle$
qed

have A $\langle proof \rangle$
then have $A \wedge B$
proof $(rule\ \langle A \Longrightarrow B \Longrightarrow A \wedge B\rangle)$
  show B $\langle proof \rangle$
qed

have A **and** B $\langle proof \rangle$
then have $A \wedge B$
  by $(rule\ \langle A \Longrightarrow B \Longrightarrow A \wedge B\rangle)$

The *induct* method

Method syntax

Idea: sophisticated wrapper for Pure $rule$ method.

Method format:

> $facts$
> ($\textit{induct}$ $insts$ $\textit{fixing:}$ $vars$ $\textbf{\textit{rule:}}$ $rule$)

- $facts$: current facts passed to any Isar method (cf. **then**, **using**)
- $insts$: induction variables x, optionally with definition $x \equiv a$
- $vars$: fixed variables
- $rule$: actual induction rule

Note: all arguments are optional.

Method operations (1)

1. context: declare local $defs$ for defined induction variables $x \equiv a$
2. rule: apply $insts$ according to conclusion $P\ x\ y\ z$
3. rule: expand $defs$ in major premises
4. rule: consume prefix of $facts$ according to major premises
5. goal: insert remaining $facts$ and $defs$
6. goal: closeup fixed variables, using $(\bigwedge x.\ B\ x) \Longrightarrow B\ a$
7. goal: internalize $\bigwedge / \Longrightarrow / \equiv$ into the object-logic
8. rule: unify conclusion against goal ($\rightarrow$ fully-instantiated rule)
9. rule: carefully recover internalized $\bigwedge / \Longrightarrow / \equiv$ in the inductive cases
10. context: extract inductive cases from rule (for **case**)
11. context: discharge $defs$
12. goal: apply fully-instantiated rule

Method operations (2) — simultaneous goals

1. goal: internalize $A \ \& \ B$ into object-logic
2. goal: apply induction rule
3. goal: recover $A \ \& \ B$ and apply congruences wrt. $\bigwedge/\Longrightarrow$
4. goal: eliminate $\&$ by $currying$
5. context: extract nested cases, numbered for each conjunct

Observation: $induct$ has its complexities, but is algorithmic — no automated reasoning here!

Common induction patterns

Local premises and parameters

lemma
  fixes $n :: nat$ **and** $x :: 'a$
  assumes $A\ n\ x$
  shows $P\ n\ x$ **using** $\langle A\ n\ x \rangle$
proof $(induct\ n\ \mathit{fixing}{:}\ x)$
  case 0
  note $prem = \langle A\ 0\ x \rangle$
  show $P\ 0\ x\ \ \langle proof \rangle$
next
  case $(Suc\ n)$
  note $hyp = \langle \bigwedge x.\ A\ n\ x \Longrightarrow P\ n\ x \rangle$
    and $prem = \langle A\ (Suc\ n)\ x \rangle$
  show $P\ (Suc\ n)\ x\ \ \langle proof \rangle$
qed

Local definitions

lemma
  fixes $a :: {}'a \Rightarrow nat$
  assumes $A\ (a\ x)$
  shows $P\ (a\ x)$ **using** $\langle A\ (a\ x)\rangle$
proof $(induct\ n \equiv a\ x\ \mathit{fixing}\text{: } x)$
  case 0
  note $prem = \langle A\ (a\ x)\rangle$ **and** $def = \langle 0 = a\ x\rangle$
  show $P\ (a\ x)$ $\langle \mathit{proof} \rangle$
next
  case $(Suc\ n)$
  note $hyp = \langle \bigwedge x.\ A\ (a\ x) \Longrightarrow n = a\ x \Longrightarrow P\ (a\ x)\rangle$
    and $prem = \langle A\ (a\ x)\rangle$ **and** $def = \langle Suc\ n = a\ x\rangle$
  show $P\ (a\ x)$ $\langle \mathit{proof} \rangle$
qed

Simultaneous goals

lemma
  fixes $n :: nat$
  shows $\bigwedge x::'a.\ A\ n\ x \Longrightarrow P\ n\ x$
  and $\bigwedge y::'b.\ B\ n\ y \Longrightarrow Q\ n\ y$
proof $(induct\ n)$
  case 0
  { case 1
    note $prem = \langle A\ 0\ x\rangle$
    show $P\ 0\ x\ \ \langle proof\rangle$ **}**
  { case 2
    note $prem = \langle B\ 0\ y\rangle$
    show $Q\ 0\ y\ \ \langle proof\rangle$ **}**
next
  case $(Suc\ n)$
  note $hyps = \langle \bigwedge x.\ A\ n\ x \Longrightarrow P\ n\ x\rangle\ \langle \bigwedge y.\ B\ n\ y \Longrightarrow Q\ n\ y\rangle$
  then have $some\text{-}interemediate\text{-}result\ \ \langle proof\rangle$

{ **case** 1
note $prem = \langle A\ (Suc\ n)\ x\rangle$
show $P\ (Suc\ n)\ x\ \ \langle proof\rangle$ }
{ **case** 2
note $prem = \langle B\ (Suc\ n)\ y\rangle$
show $Q\ (Suc\ n)\ y\ \ \langle proof\rangle$ }
qed

Conclusion

Stocktaking

- Isabelle/Isar framework is sufficiently flexible to support domain specific proof patterns
- Minimal requirements on induction rule format, possible extensions include:
  - nominal induction: additional "freshness" context ($nominal\text{-}induct\ x\ avoiding\colon\ a\ b\ c\ fixing\colon\ u\ v$)
  - coinduction: dualized version (not fully implemented yet) ($coinduct\ x\ fixing\colon\ u\ v$)
- Further examples: cf. POPLmark solutions by Berghofer ($induct$), and Urban ($nominal\text{-}induct$)
- Paper available: http://isabelle.in.tum.de/Isar/Isar-induct.pdf