The Isar Proof Language in 2016

Makarius Wenzel
sketis.net

August 2016

Introduction

History of Isar

1999: first usable version

- primary notion of proof document (not "proof script")
- secondary notion of proof method (not "tactic")

2000–2001: various refinements

2006: minor reforms

- **unfolding**, **obtains**, literal facts: $\langle prop \rangle$
- advanced $induct$ method

2016: major renovations

where "2016" means . . .

- current release: Isabelle2016 (February 2016)
- coming release: Isabelle2016-1 (November/December 2016)

Structured statements

Structured assumptions

Postfix notation for Horn-clauses:

- **assume** B **if** A_1 **and** A_2 **for** $a_1\ a_2$
  - corresponds to **assume** $\bigwedge a_1\ a_2.\ A_1 \Longrightarrow A_2 \Longrightarrow B$
  - vacuous quantifiers are omitted
- similar for **obtain**, **define**
- similar for **inductive**, **definition**, **function** etc.

Example: structured specifications

inductive_set $star$ $(\_\star\ [100]\ 100)$ **for** $R :: ('a \times 'a)\ set$
  where
    $base$: $(x,\ x) \in R\star$ **for** x
  $|$ $step$: $(x,\ z) \in R\star$ **if** $(x,\ y) \in R$ **and** $(y,\ z) \in R\star$ **for** $x\ y\ z$

function $gcd :: nat \Rightarrow nat \Rightarrow nat$
  where
    $gcd\ x\ 0 = x$
  $|$ $gcd\ 0\ y = y$
  $|$ $gcd\ (Suc\ x)\ (Suc\ y) = gcd\ (Suc\ x)\ (y - x)$ **if** $x < y$
  $|$ $gcd\ (Suc\ x)\ (Suc\ y) = gcd\ (x - y)\ (Suc\ y)$ **if** $\neg\ x < y$

Structured conclusions (goals)

Notation for Isar "eigen-context":

- premises: **have** B **if** A_1 A_2
- parameters: **have** B **for** a_1 a_2

- corresponds to $\{$ **fix** a_1 a_2 **assume** $that$: A_1 A_2 **have** B $\}$
- analogous to **lemma fixes** a_1 a_2 **assumes** $that$: A_1 A_2 **shows** B

Example: Natural Deduction with structured conclusions

- conjunction introduction:
  have $A \wedge B$ **if** A **and** B
- existential introduction:
  have $\exists\, x.\ B\ x$ **if** $B\ a$ **for** a
- disjunction elimination:
  from $\langle A \vee B \rangle$ **have** C **if** $A \Longrightarrow C$ **and** $B \Longrightarrow C$ **for** C
- existential elimination:
  from $\langle \exists\, x.\ B\ x \rangle$ **have** C **if** $\bigwedge x.\ B\ x \Longrightarrow C$ **for** C

Elimination statements

consider $\overline{x}$ **where** $\overline{A}\ \overline{x}$ $\mid$ $\overline{y}$ **where** $\overline{B}\ \overline{y}$ $\mid$ $\ldots$ $\equiv$

  have thesis

    if $\bigwedge \overline{x}.\ \overline{A}\ \overline{x} \Longrightarrow \mathit{thesis}$

    and $\bigwedge \overline{y}.\ \overline{B}\ \overline{y} \Longrightarrow \mathit{thesis}$

    for thesis

Examples:

- existential elimination:
  from $\langle \exists\, x.\ B\ x \rangle$ **consider** x **where** $B\ x$
- conjunction elimination:
  from $\langle A \wedge B \rangle$ **consider** A **and** B
- disjunction elimination:
  from $\langle A \vee B \rangle$ **consider** $A \mid B$

Elimination and cases

- method "*cases*" detects its rule from chained facts
- command "**case**" allows name and attribute specification

Example:

```
consider $x$ where $A\ x \mid y$ where $B\ y$ $\langle proof \rangle$
then have $something$
proof $cases$
  case $prems$: $1$
  show $?thesis$ using $prems$ $\langle proof \rangle$
next
  case $prems$: $2$
  show $?thesis$ using $prems$ $\langle proof \rangle$
qed
```

Obtain

$$\boxed{\begin{array}{l}\textbf{obtain}\ \overline{x}\ \textbf{where}\ \overline{A\ \overline{x}}\ \langle \mathit{proof}\rangle \ \equiv \\ \quad \textbf{consider}\ \overline{x}\ \textbf{where}\ \overline{A}\ \overline{x}\ \langle \mathit{proof}\rangle \\ \quad \textbf{fix}\ \overline{x}\ \textbf{assume}^{*}\ \overline{A}\ \overline{x}\end{array}}$$

- old meaning is unchanged, but foundation simplified
- **is** patterns now supported (with λ-lifting over the parameters)
- **if** / **for** notation available as well

Define

$$\boxed{\begin{array}{l}\textbf{define}\ c\ \textbf{where}\ c\ \overline{x} = t\ \textbf{for}\ \overline{x}\ \ \equiv \\ \quad \textbf{def}\ c \equiv \lambda \overline{x}.\ t\end{array}}$$

- syntax like **obtain**
- analogous to **definition** (e.g. object-logic equalities)
- old **def** is declared legacy

Strong vs. weak premises

- strong premises (cf. **assume**): **show** B **if** A_1 **and** A_2
- weak premises (cf. **presume**): **show** B **when** A_1 **and** A_2

- **show** $A_1 \Longrightarrow A_2 \Longrightarrow B$ becomes free for re-interpretation:

```
have ‹A ⟶ B›
proof
  show ‹B› if ‹A› ⟨proof⟩  — strong premise (new in 2016)
qed

have ‹A ⟶ B›
proof
  show ‹A ⟹ B› ⟨proof⟩   — strong premise (changed in 2016)
qed
```

Proof structure

Simplified block structure

Nesting levels:

$+$ goal statement (**have**, **show** etc.)

$=$ backwards refinement (**using**, **apply**, **supply** etc.)

$+$ **proof**

$+$ **{**

$-$ **}**

$--$ **qed**

Some consequences:

- cases in proof methods no longer special (regular context update)
- Eisbach: *match* method can use generic context for bookkeeping
- Isabelle/jEdit: clarified text folding and indentation

Structured backwards refinement

$\langle goal \rangle$
subgoal premises $prems$ **for** $x_1\ x_2\ \ldots$
  $\langle proof \rangle$

Example: structured apply "scripts"

$\langle goal \rangle$
  subgoal by $method_1$
  subgoal by $method_2$
  done

$\langle goal \rangle$
  subgoal premises $prems$ **for** $x\ y$ **using** $prems$ $\langle proof_1 \rangle$
  subgoal premises $prems$ **for** $u\ v\ w$ **using** $prems$ $\langle proof_2 \rangle$
  done

Proof method facts

Used facts of method expression:

- get via dynamic fact *method_facts*
  (useful for Eisbach method definitions)
- set via method *use*, e.g.
  $(use\ \ldots$ **in** $simp)$
  $(use\ \ldots$ **in** $\langle simp\ add{:}\ \ldots\rangle)$
- special fact *nothing* may help in odd situations

Example:

have $a{:}\ A\ \ \langle proof\rangle$
have B **by** $(rule\ r)\ (use\ a$ **in** $auto)$

Isar document language

Document structure

Markup

- section headings (6 levels like in HTML):
  chapter, **section**, **subsection**, . . . , **subparagraph**
- text blocks: **text**, **txt**, **text_raw**
- uncontrolled LaTeX macros (rare)

Markdown

- implicit paragraphs and lists: itemize, enumerate, description

Document antiquotations

full form: `@{`$name$ $[options]$ $arguments$ $\ldots$`}`

short form:

1. cartouche argument: `\<^`$name$`>`$\langle argument\rangle$
2. no argument: `\<^`$name$`>`
3. standard name: $\langle argument\rangle$

Notable examples:

- *cartouche*, *theory_text*: self-presentation of Isar
- *bold*, *emph*, *verbatim*, *footnote*: text styles (with proper nesting)
- *noindent*, *smallskip*, *medskip*, *bigskip*: spacing
- *cite*: formal BibTEX items
- *path*, *file*, *dir*, *url*, *doc*: system resources

Conclusions

Conclusions

- raw proof blocks $\{ \dots \}$ are rarely required:
  superseded by structured conclusions with **if** / **for** eigen-context
- big-bang integration of several cases "**by** *blast*" is obsolete:
  superseded by **consider** and enhanced method *cases*
- integration of unstructured **apply**-scripts into structured proofs works better (notably via **subgoal**)
- auxiliary method *insert* is mostly obsolete:
  superseded by (*use* … **in** *method*)

- Isar document reforms make it very easy to make presentations on the spot, without LaTeX tinkering

TODO

- error-recovery according to block structure of proof document
- re-unify *where* / *of* in Eisbach vs. Pure
- re-unify *atomize* and *atomize_elim* as *compact*
- make *compact* the default of automated methods
- re-unify of *induct* / *induction* and *coinduct* / *coinduction*
- eliminate *induct_tac*, *case_tac*, *rule_tac* eventually
- proper HTML document output (e.g. presentations with **reveal.js**)
- interactive document preparation (with PIDE)