Isabelle/Isar: from Primitive Natural Deduction to Structured Mathematical Reasoning

Makarius

27-June-2005

1. Representing Proofs

Primitive Natural Deduction (1)

$$\frac{A \quad B}{A \wedge B}\,(\wedge I) \qquad \frac{A \wedge B}{A}\,(\wedge E_1) \quad \frac{A \wedge B}{B}\,(\wedge E_2)$$

$$\frac{A}{A \vee B}\,(\vee I_1) \quad \frac{B}{A \vee B}\,(\vee I_2) \qquad \frac{A \vee B \quad \begin{array}{c}[A]\\ \vdots \\ C\end{array} \quad \begin{array}{c}[B]\\ \vdots \\ C\end{array}}{C}\,(\vee E)$$

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

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

Primitive Natural Deduction (2)

Observations:

- nice in theory
- cute in small teaching tools
- cumbersome in realistic applications
- not quite natural after all . . .

$$\frac{B(t)}{\exists\, x.\ B(x)}\,(\exists\, I) \qquad \frac{\exists\, x.\ B(x) \qquad \begin{matrix}[x,\ B(x)] \\ \vdots \\ C\end{matrix}}{C}\,(\exists\, E)$$

Mathematical vernacular

Example: [Davey and Priestley, 1990, pages 93–94]

The Knaster-Tarski Fixpoint Theorem. Let L be a complete lattice and $f\colon L \to L$ an order-preserving map. Then $\bigsqcap\{x \in L \mid f(x) \le x\}$ is a fixpoint of f.

Proof. Let $H = \{x \in L \mid f(x) \le x\}$ and $a = \bigsqcap H$. For all $x \in H$ we have $a \le x$, so $f(a) \le f(x) \le x$. Thus $f(a)$ is a lower bound of H, whence $f(a) \le a$. We now use this inequality to prove the reverse one (!) and thereby complete the proof that a is a fixpoint. Since f is order-preserving, $f(f(a)) \le f(a)$. This says $f(a) \in H$, so $a \le f(a)$.

Question: How can we do actual formalized mathematics?

The Mizar system

Mizar [A. Trybulec *et al.*, since $\approx$ 1973]

- Original motivation: verification environment for ALGOL programs (the name MIZAR is a pun on that)
- Large library of formalized mathematics — "Journal of Formalized Mathematics" [Vol. 1–12, 1990–2004]
- Mathematical proof language (!)
- Logical foundations:
  - classical first-order logic
  - builtin classical reasoning (decomposition and terminal steps)
  - some special support for "schemes" (e.g. induction)
  - typed set-theory (Tarski-Grothendieck)
  - builtin concept of abstract mathematical structures
- Main problem: monolithic system (no formal record on derivations, no interfaces for extensions, program sources not generally available)

The Isabelle/Isar system

Isabelle [L.C. Paulson and T. Nipkow, since $\approx$ 1986]

- Generic logical framework for higher-order Natural Deduction
- Syntax: simply-typed λ-calculus with $\alpha\beta\eta$-conversion, builtin support for higher-order unification
- Deduction: minimal higher-order logic with implication $A \Longrightarrow B$, quantification $\bigwedge x.\ B(x)$, and equality $t \equiv u$

Isar [M. Wenzel, since $\approx$ 1999]

- "Intelligible semi-automated reasoning"
- simple logical foundations, inherited from Isabelle/Pure
- generic – common object-logics may benefit from Isar immediately
- succinct language design, few basic principles, several derived concepts
- incremental proof processing, interactive development and debugging
- final proof texts intelligible without replay on the machine (requires some care of the author)

Example: Isabelle/Isar proof text

theorem $\mathit{Knaster\text{-}Tarski}$:
  assumes mono: $\bigwedge x\ y.\ x \le y \Longrightarrow f\ x \le f\ y$
  shows $f\ (\bigsqcap\{x.\ f\ x \le x\}) = \bigsqcap\{x.\ f\ x \le x\}$ (**is** $f\ ?a = ?a$)
proof $-$
  have $*$: $f\ ?a \le ?a$ (**is** - $\le \bigsqcap\ ?H$)
  proof
    fix x **assume** H: $x \in ?H$
    then have $?a \le x$ **..**
    also from H **have** $f\ \ldots \le x$ **..**
    moreover note mono **finally show** $f\ ?a \le x$ **.**
  qed
  also have $?a \le f\ ?a$
  proof
    from mono **and** $*$ **have** $f\ (f\ ?a) \le f\ ?a$ **.**
    then show $f\ ?a \in ?H$ **..**
  qed
  finally show $f\ ?a = ?a$ **.**
qed

Example: Isabelle/Pure proof term

$$
\begin{array}{l}
\mathit{Knaster\text{-}Tarski} \equiv \\
\boldsymbol{\lambda} H\colon \text{-}. \\
\ \mathit{order\text{-}antisym} \cdot \text{-} \cdot \text{-} \bullet \\
\ \ (\mathit{Inter\text{-}greatest} \cdot \text{-} \cdot \text{-} \bullet \\
\ \ \ \ (\boldsymbol{\lambda} X\ Ha\colon \text{-}. \\
\ \ \ \ \ \ \mathit{order\text{-}subst2} \cdot \text{-} \cdot \text{-} \cdot f \cdot \text{-} \bullet (\mathit{Inter\text{-}lower} \cdot \text{-} \cdot \text{-} \bullet Ha) \bullet \\
\ \ \ \ \ \ \ (\mathit{iffD1} \cdot \text{-} \cdot \text{-} \bullet (\mathit{mem\text{-}Collect\text{-}eq} \cdot \text{-} \cdot (\lambda x.\ f\ x \leq x)) \bullet Ha) \bullet \\
\ \ \ \ \ \ \ H)) \bullet \\
\ \ (\mathit{Inter\text{-}lower} \cdot \text{-} \cdot \text{-} \bullet \\
\ \ \ \ (\mathit{iffD2} \cdot \text{-} \cdot \text{-} \bullet (\mathit{mem\text{-}Collect\text{-}eq} \cdot \text{-} \cdot (\lambda u.\ f\ u \leq u)) \bullet \\
\ \ \ \ \ \ (H \cdot f\ (\bigsqcap \{x.\ f\ x \leq x\}) \cdot \bigsqcap \{x.\ f\ x \leq x\} \bullet \\
\ \ \ \ \ \ \ \ (\mathit{Inter\text{-}greatest} \cdot \text{-} \cdot \text{-} \bullet \\
\ \ \ \ \ \ \ \ \ \ (\boldsymbol{\lambda} X\ Ha\colon \text{-}. \\
\ \ \ \ \ \ \ \ \ \ \ \ \mathit{order\text{-}subst2} \cdot \text{-} \cdot \text{-} \cdot f \cdot \text{-} \bullet (\mathit{Inter\text{-}lower} \cdot \text{-} \cdot \text{-} \bullet Ha) \bullet \\
\ \ \ \ \ \ \ \ \ \ \ \ \ (\mathit{iffD1} \cdot \text{-} \cdot \text{-} \bullet (\mathit{mem\text{-}Collect\text{-}eq} \cdot \text{-} \cdot (\lambda x.\ f\ x \leq x)) \bullet Ha) \bullet \\
\ \ \ \ \ \ \ \ \ \ \ \ \ H)))))
\end{array}
$$

2. Isabelle/Isar Foundations

Isabelle/Pure syntax and rules

$prop$	type of propositions
$\Longrightarrow :: prop \Rightarrow prop \Rightarrow prop$	implication (right-associative infix)
$\bigwedge :: (\alpha \Rightarrow prop) \Rightarrow prop$	universal quantifier (binder)
$\equiv :: \alpha \Rightarrow \alpha \Rightarrow prop$	equality relation (infix)

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

$$\frac{\begin{matrix}[x] \\ \vdots \\ B(x)\end{matrix}}{\bigwedge x.\ B(x)}\,(\bigwedge I) \qquad \frac{\bigwedge x.\ B(x)}{B(t)}\,(\bigwedge E)$$

Axioms for $t \equiv u$: α, β, η, $refl$, $subst$, ext, iff

Pure formulae vs. inferences (1)

Define the following sets:

$\boldsymbol{x}$	variables
$\boldsymbol{A}$	atomic formulae, i.e. no outermost $\Longrightarrow/\bigwedge$
$\bigwedge \boldsymbol{x}^*.\ \boldsymbol{A}^* \Longrightarrow \boldsymbol{A}$	Horn Clauses
$\boldsymbol{H} \stackrel{\text{def}}{=} \bigwedge \boldsymbol{x}^*.\ \boldsymbol{H}^* \Longrightarrow \boldsymbol{A}$	Harrop Formulas
$\boldsymbol{G} \stackrel{\text{def}}{=} \boldsymbol{H} \cup \#\boldsymbol{H}$	Goal Clauses ($\# \equiv \lambda A.\ A$)

Notes:

- Outermost quantification $\bigwedge x.\ B\ x$ is always represented via schematic variables $B\ ?x$
- $(A \Longrightarrow (\bigwedge x.\ B\ x)) \equiv (\bigwedge x.\ A \Longrightarrow B\ x)$ holds, i.e. every Pure formula may be put into Harrop Form
- the goal marker $\#$ makes any Harrop Formula appear atomic

Pure formulae vs. inferences (2)

Examples:

Horn: $A \Longrightarrow B \Longrightarrow A \wedge B$ $\qquad \dfrac{A \quad B}{A \wedge B}$

Harrop: $(A \Longrightarrow B) \Longrightarrow A \longrightarrow B$ $\qquad \dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \longrightarrow B}$

Harrop: $P\ 0 \Longrightarrow (\bigwedge n.\ P\ n \Longrightarrow P\ (Suc\ n)) \Longrightarrow P\ n$ $\qquad \dfrac{P\ 0 \quad \begin{matrix}[n,\ P\ n]\\ \vdots \\ P\ (Suc\ n)\end{matrix}}{P\ n}$

Goal: $(A \Longrightarrow B \Longrightarrow B) \Longrightarrow (A \Longrightarrow B \Longrightarrow A) \Longrightarrow \#(A \wedge B \longrightarrow B \wedge A)$

Rules for goal directed proof (1)

$$\frac{}{A \Longrightarrow \#A}\,(init) \qquad \frac{\#A}{A}\,(conclude)$$

$$\frac{\begin{array}{rl} \mathit{rule}: & \vec{A}\ \vec{a} \Longrightarrow B\ \vec{a} \\ \mathit{goal}: & (\bigwedge \vec{x}.\ \vec{H}\ \vec{x} \Longrightarrow B'\ \vec{x}) \Longrightarrow C \\ \mathit{goal\ unifier}: & (\lambda \vec{x}.\ B\ (\vec{a}\ \vec{x}))\,\theta = B'\theta \end{array}}{(\bigwedge \vec{x}.\ \vec{H}\ \vec{x} \Longrightarrow \vec{A}\ (\vec{a}\ \vec{x}))\,\theta \Longrightarrow C\,\theta}\,(resolve)$$

$$\frac{\begin{array}{rl} \mathit{goal}: & (\bigwedge \vec{x}.\ \vec{H}\ \vec{x} \Longrightarrow A\ \vec{x}) \Longrightarrow C \\ \mathit{assm\ unifier}: & A\,\theta = H_i\theta\ \ \text{(for some } H_i\text{)} \end{array}}{C\,\theta}\,(assumption)$$

Example: tactical proving in Isabelle

```
lemma A ∧ B ⟶ B ∧ A
  apply (rule impI)
  apply (erule conjE)
  apply (rule conjI)
    apply assumption
   apply assumption
  done
```

```
lemma (∃ x. ∀ y. R x y) ⟶ (∀ y. ∃ x. R x y)
  apply (rule impI)
  apply (erule exE)
  apply (rule allI)
  apply (erule allE)
  apply (rule exI)
  apply assumption
  done
```

Rules for goal directed proof (2)

The key rule for Isar:

$$\frac{\begin{array}{rl} \mathit{subproof}\!: & \vec{G}\ \vec{a} \Longrightarrow B\ \vec{a} \\ \mathit{goal}\!: & (\bigwedge \vec{x}.\ \vec{H}\ \vec{x} \Longrightarrow B'\ \vec{x}) \Longrightarrow C \\ \mathit{goal\ unifier}\!: & (\lambda \vec{x}.\ B\ (\vec{a}\ \vec{x}))\,\theta = B'\theta \\ \mathit{assm\ unifiers}\!: & (\lambda \vec{x}.\ G_j\ (\vec{a}\ \vec{x}))\,\theta = \#H_i\theta\ \ \text{(for marked } G_j \text{ some } \#H_i\text{)} \end{array}}{(\bigwedge \vec{x}.\ \vec{H}\ \vec{x} \Longrightarrow \vec{G}'\ (\vec{a}\ \vec{x}))\,\theta \Longrightarrow C\,\theta}\ (\mathit{refine})$$

Corresponds to canonical proof decomposition:

have $\bigwedge x.\ A\ x \Longrightarrow B\ x$
proof $-$
  fix x
  assume $A\ x$
  show $B\ x$ $\langle \mathit{proof} \rangle$
qed

3. The Isar Proof Language

Isar primitives

apply $meth$	unstructured refinement
done	unstructured ending
proof $meth^?$	structured refinement
qed $meth^?$	structured ending
$\{$	open block
$\}$	close block
next	switch block
let $pat = t$	term abbreviation
note $a = bs$	reconsidered facts
fix $\vec{x}$	universal parameters
assm $\ll rule \gg\ a\colon \vec{A}$	generic assumptions
then	indicate forward-chaining of facts
have $a\colon A$	local claim
show $a\colon A$	local claim, result refines goal

Derived elements

assume	=	**assm** $\ll discharge\#\gg$
presume	=	**assm** $\ll discharge\gg$
def $x \equiv t$	=	**fix** x **assm** $\ll expand\gg$ $x \equiv t$
hence	=	**then have**
thus	=	**then show**
from a	=	**note** a **then**
with a	=	**from** a **and** $this$
by $meth_1$ $meth_2$	=	**proof** $meth_1$ **qed** $meth_2$
..	=	**by** $rule$
.	=	**by** $this$

$$\frac{\Gamma \cup \vec{A} \vdash C}{\Gamma \vdash \#\vec{A} \Longrightarrow C}(discharge\#) \qquad \frac{\Gamma \cup \vec{A} \vdash C}{\Gamma \vdash \vec{A} \Longrightarrow C}(discharge)$$

$$\frac{\Gamma \cup x \equiv t \vdash C\ t}{\Gamma \vdash C\ x}(expand)$$

The Isar/VM interpretation process

Isar/VM = much book-keeping + some Isabelle/Pure inferences

Important fields in the machine state (block-structured):

$fixes$	context of locally fixed variables
$assms$	context of local assumptions, each with discharge rule
$facts$	environment of local facts
$goal$	(optional) enclosing problem to be worked on

Some notable $facts$:

"$prems$"	current assumptions
"$this$"	most recently established fact
"$calculation$"	scratch-pad for calculational reasoning

Example: structured proofs in Isar

```
lemma A ∧ B ⟶ B ∧ A
proof
  assume A ∧ B
  then show B ∧ A
  proof
    assume B and A
    then show B ∧ A ..
  qed
qed
```

```
lemma (∃ x. ∀ y. R x y) ⟶ (∀ y. ∃ x. R x y)
proof
  assume ∃ x. ∀ y. R x y
  then show ∀ y. ∃ x. R x y
  proof
    fix a
    assume *: ∀ y. R a y
    show ∀ y. ∃ x. R x y
    proof
      fix y
      show ∃ x. R x y
      proof
        fix b
        from * show R a b ..
      qed
    qed
  qed
qed
```

4. Advanced Techniques

Generalized elimination

obtain $\vec{x}$ **where** $\vec{B}\ \vec{x}\ \ \langle proof \rangle\ \ \stackrel{\mathsf{def}}{=}$

have $reduction$: $\bigwedge thesis.\ (\bigwedge \vec{x}.\ \vec{B}\ \vec{x} \Longrightarrow thesis) \Longrightarrow thesis\ \langle proof \rangle$
fix $\vec{x}$ **assm** $\ll eliminate\ reduction \gg\ \vec{B}\ \vec{x}$

$$\frac{\Gamma \vdash \bigwedge thesis.\ (\bigwedge \vec{x}.\ \vec{B}\ \vec{x} \Longrightarrow thesis) \Longrightarrow thesis \qquad \Gamma \cup \vec{B}\ \vec{y} \vdash C}{\Gamma \vdash C}\ (eliminate)$$

Canonical proof patterns:

assume $\exists\, x.\ B\ x$
then obtain x **where** $B\ x$ **..**

assume $A \wedge B$
then obtain A **and** B **..**

Example: forward elimination

lemma $A \wedge B \longrightarrow B \wedge A$
proof
  assume $A \wedge B$
  then obtain B **and** A **..**
  then show $B \wedge A$ **..**
qed

lemma $(\exists\, x.\ \forall\, y.\ R\ x\ y) \longrightarrow (\forall\, y.\ \exists\, x.\ R\ x\ y)$
proof
  assume $\exists\, x.\ \forall\, y.\ R\ x\ y$
  then obtain a **where** $*$: $\forall\, y.\ R\ a\ y$ **..**
  { fix b **from** $*$ **have** $R\ a\ b$ **..**
    then have $\exists\, x.\ R\ x\ b$ **.. }**
  then show $\forall\, y.\ \exists\, x.\ R\ x\ y$ **..**
qed

Calculational reasoning

also	$=$	**note** $\mathit{calculation} = \mathit{this}$	initially
also	$=$	**note** $\mathit{calculation} = r \cdot (\mathit{calculation}\ @\ \mathit{this})$	for $r \in \boldsymbol{T}$
finally	$=$	**also from** $\mathit{calculation}$	
moreover	$=$	**note** $\mathit{calculation} = \mathit{calculation}\ @\ \mathit{this}$	
ultimately	$=$	**moreover from** $\mathit{calculation}$	

$$\boldsymbol{T} \stackrel{\mathsf{def}}{=} \{x = y \Longrightarrow y = z \Longrightarrow x = z,\ x \leq y \Longrightarrow y \leq z \Longrightarrow x \leq z,\ \ldots\}$$

Canonical proof pattern:

have $a = b$ $\langle \mathit{proof} \rangle$
also have $\ldots = c$ $\langle \mathit{proof} \rangle$
also have $\ldots = d$ $\langle \mathit{proof} \rangle$
finally have $a = d$ **.**

Note: term "$\ldots$" abbreviates right-hand side of last statement

Mathematical structures as structured proof contexts

Idea: expressions for Isar proof contexts

Concrete syntax:

locale $name = expr + elem^*$

$expr ::= name \mid expr + expr \mid expr\ name^*$
$elem ::=$ **fixes** $vars \mid$ **assumes** $props \mid$ **defines** $terms \mid$ **notes** $facts$

- locale activation turns **fixes** into **fix**, and **assumes** into **assume** etc.
- special form **theorem** (**in** a) augments the context dynamically by further **notes** (no change of logical content)

Example: locales and calculational reasoning

locale *group* =
  fixes *prod* (**infixl** $\cdot$ 70)
    and *inv* ((-$^{-1}$) [1000] 999)
    and *one* (1)
  assumes *assoc*: $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
    and *left-inv*: $x^{-1} \cdot x = 1$
    and *left-one*: $1 \cdot x = x$

theorem (**in** *group*) *right-inv*: $x \cdot x^{-1} = 1$ ⟨*proof*⟩

theorem (**in** *group*) *right-one*: $x \cdot 1 = x$
proof −
  have $x \cdot 1 = x \cdot (x^{-1} \cdot x)$ **by** (*simp only*: *left-inv*)
  also have $\ldots = (x \cdot x^{-1}) \cdot x$ **by** (*simp only*: *assoc*)
  also have $\ldots = 1 \cdot x$ **by** (*simp only*: *right-inv*)
  also have $\ldots = x$ **by** (*simp only*: *left-one*)
  finally show $x \cdot 1 = x$ **.**
qed

Isar statements

(1) allow logical statements to express their context using Isar locale elements:
theorem $elem^*$ **shows** $props$

Example:

lemma
fixes x **and** y **and** z
defines $x \equiv y + z$
assumes A **and** B
shows C

(2) introduce the following abbreviation:

obtains $\vec{x}$ **where** $\vec{B}\ \vec{x}$ **or** $\ldots \stackrel{\text{def}}{=}$

fixes $thesis$
assumes $\bigwedge\vec{x}.\ \vec{B}\ \vec{x} \Longrightarrow thesis$ **and** $\ldots$
shows $thesis$

Natural Deduction rules as Isar statements

$conjI$: **assumes** A **and** B **shows** $A \wedge B$
$conjE$: **assumes** $A \wedge B$ **obtains** A **and** B

$disjI_1$: **assumes** A **shows** $A \vee B$
$disjI_2$: **assumes** B **shows** $A \vee B$
$disjE$: **assumes** $A \vee B$ **obtains** A **or** B

$impI$: **assumes** $A \Longrightarrow B$ **shows** $A \longrightarrow B$
$impE$: **assumes** $A \longrightarrow B$ **and** A **obtains** B

$allI$: **assumes** $\bigwedge x.\ B\ x$ **shows** $\forall\, x.\ B\ x$
$allE$: **assumes** $\forall\, x.\ B\ x$ **obtains** $B\ t$

exI: **assumes** $B\ t$ **shows** $\exists\, x.\ B\ x$
exE: **assumes** $\exists\, x.\ B\ x$ **obtains** x **where** $B\ x$

$\longrightarrow$ Towards logic-free reasoning?

Conclusion

Isabelle/Isar applications

Present state:

- 2000–2005: considerable amounts of Isabelle/Isar theories have emerged, see also "The Archive of Formal Proofs" http://afp.sourceforge.net/
- Everybody uses the Isabelle/Isar toplevel — with Proof General
- Some people do actual structured proof development

Future work:

- More tool support for quick composition of formal proof sketches
- More documentation
- More instructions
- . . .