Pure Reasoning in Isabelle/Isar

Makarius Wenzel
TU München

January 2009

1. The Pure framework
2. Pure rules everywhere
3. Isar statements
4. Inductive definitions

Introduction

Aims

- improved understanding how Isabelle and Isar really work (Isabelle $\neq$ HOL)
- natural reasoning, less formal overhead in applications
- native representations of statements and definitions
- reduced demand for "logical encodings"
- less arbitrary "automated reasoning"

Isabelle/Pure framework (Paulson 1989)

Logical framework: 3 levels of λ-calculus

$\alpha \Rightarrow \beta$	terms depending on terms
$\bigwedge x.\ B\ x$	proofs depending on terms
$A \Longrightarrow B$	proofs depending on proofs

Rule composition: via higher-order unification

$resolution$: mixed forward-back chaining

$assumption$: closing branches

Note: arbitrary nesting of rules

Isabelle/Isar proof language (Wenzel 1999)

Main idea: Pure rules turned into proof schemes

from $facts_1$ **have** $props$ **using** $facts_2$
proof $(rule)$
  $body$
qed

Solving sub-problems: within $body$

fix $vars$
assume $props$
show $props$ $\langle proof \rangle$

Abbreviations:

then	$\equiv$	**from** $this$
..	$\equiv$	**proof qed**

The Pure framework

Pure syntax and primitive rules

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

$$\frac{\begin{array}{c}[x :: \alpha] \\ \vdots \\ b(x) :: \beta\end{array}}{\lambda x.\ b(x) :: \alpha \Rightarrow \beta}\,(\Rightarrow I) \qquad \frac{b :: \alpha \Rightarrow \beta \quad a :: \alpha}{b(a) :: \beta}\,(\Rightarrow E)$$

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

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

Pure equality

$\equiv :: \alpha \Rightarrow \alpha \Rightarrow prop$

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

Unification: solving equations modulo $\alpha\beta\eta$

- Huet: full higher-order unification (infinitary enumeration!)
- Miller: higher-order patterns (unique result)

(Example: Pure primitives)

Hereditary Harrop Formulas (HHF)

Define the following sets:

$\boldsymbol{x}$	variables
$\boldsymbol{A}$	atomic formulae (without $\Longrightarrow/\bigwedge$)
$\bigwedge \boldsymbol{x}^*.\ \boldsymbol{A}^* \Longrightarrow \boldsymbol{A}$	Horn Clauses
$\boldsymbol{H} \stackrel{\text{def}}{=} \bigwedge \boldsymbol{x}^*.\ \boldsymbol{H}^* \Longrightarrow \boldsymbol{A}$	Hereditary Harrop Formulas (HHF)

Conventions for results:

- outermost quantification $\bigwedge x.\ B\ x$ is rephrased via schematic variables $B\ ?x$
- equivalence $(A \Longrightarrow (\bigwedge x.\ B\ x)) \equiv (\bigwedge x.\ A \Longrightarrow B\ x)$ produces canonical HHF

Pure rules everywhere

Natural Deduction rules

Examples:

$$\frac{A \quad B}{A \wedge B} \qquad\qquad A \Longrightarrow B \Longrightarrow A \wedge B$$

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

$$\frac{P\ 0 \quad \begin{array}{c}[n][P\ n]\\ \vdots \\ P\ (Suc\ n)\end{array}}{P\ n} \qquad\qquad P\ 0 \Longrightarrow (\bigwedge n.\ P\ n \Longrightarrow P\ (Suc\ n)) \Longrightarrow P\ n$$

Implicit rules in Isar proofs

have A **and** B $\langle proof \rangle$
then have $A \wedge B$ **..**

have $A \longrightarrow B$
proof $(rule\ impI)$
  assume A
  show B $\langle proof \rangle$
qed

fix $n :: nat$
have $P\ n$
proof $(induct\ n)$
  show $P\ 0$ $\langle proof \rangle$
  fix n **assume** $P\ n$
  show $P\ (Suc\ n)$ $\langle proof \rangle$
qed

Goal state as rule

Protective marker:

$\# :: prop \Rightarrow prop$
$\# \equiv \lambda A :: prop.\ A$

Initialization:

$$\frac{}{C \Longrightarrow \# C}\,(init)$$

General situation: subgoals imply main goal

$$B_1 \Longrightarrow \ldots \Longrightarrow B_n \Longrightarrow \# C$$

Finalization:

$$\frac{\# C}{C}\,(finish)$$

(Example: Goal directed proof and rule composition)

Rule composition (back-chaining)

$$\frac{\vec{A} \Longrightarrow B \quad B' \Longrightarrow C \quad B\,\theta = B'\theta}{\vec{A}\theta \Longrightarrow C\,\theta}\,(compose)$$

$$\frac{\vec{A} \Longrightarrow B}{(\vec{H} \Longrightarrow \vec{A}) \Longrightarrow (\vec{H} \Longrightarrow B)}\,(\Longrightarrow\text{-}lift)$$

$$\frac{\vec{A}\ \vec{a} \Longrightarrow B\ \vec{a}}{(\bigwedge\vec{x}.\ \vec{A}\ (\vec{a}\ \vec{x})) \Longrightarrow (\bigwedge\vec{x}.\ B\ (\vec{a}\ \vec{x}))}\,(\bigwedge\text{-}lift)$$

General higher-order resolution

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

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

Both inferences are omnipresent in Isabelle/Isar:

- *resolution*: e.g. *OF* attribute, *rule* method, **also** command
- *assumption*: e.g. *assumption* method, implicit proof ending

Application: calculational reasoning

$$
\begin{array}{rcl}
\textbf{also}_0 & = & \textbf{note}\ \mathit{calculation} = \mathit{this} \\
\textbf{also}_{n+1} & = & \textbf{note}\ \mathit{calculation} = \mathit{trans}\ [\mathit{OF}\ \mathit{calculation}\ \mathit{this}] \\
\textbf{finally} & = & \textbf{also from}\ \mathit{calculation}
\end{array}
$$

Example:

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 "..." abbreviates the argument of the last statement

(Example: Calculations)

Isar statements

From contexts to statements

Idea:

- Avoid unwieldy logical formula, i.e.
  no object-logic: $\forall x.\ A\ x \longrightarrow B\ x$
  no meta-logic: $\bigwedge x.\ A\ x \Longrightarrow B\ x$
- Use native Isar context & conclusion elements
  fixes x **assumes** $A\ x$ **shows** $B\ x$ corresponding to $x,\ A\ x \vdash B\ x$

Example:

theorem
  fixes x **and** y
  assumes a: $A\ x$ **and** b: $B\ y$
  shows $C\ x\ y$
proof $-$
  from a **and** b **show** *?thesis* $\langle proof\rangle$
qed

Proof context elements

Universal: **fix** and **assume**

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

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

Existential: **obtain**

{
obtain a **where** $B\ a$ $\langle proof \rangle$
have C $\langle proof \rangle$
}
note $\langle C \rangle$

Clausal Isar statements

Big clauses: **fixes** x **assumes** $A\ x$ **shows** $B\ x$
based on primitive Isar context elements

Dual clauses: **obtains** a **where** $B\ a$ **|** $\ldots$ expands to
fixes $thesis$ **assumes** $\bigwedge a.\ B\ a \Longrightarrow thesis$ **and** $\ldots$ **shows** $thesis$

Small clauses: $B\ x$ **if** $A\ x$ **for** x as second-level rule structure
$\bigwedge x.\ A\ x \Longrightarrow B\ x$ within big clauses

Experimental!

Example: Isar statements for predicate logic

theorem $impI$: **assumes** B **if** A **shows** $A \longrightarrow B$
theorem $impE$: **assumes** $A \longrightarrow B$ **and** A **shows** B

theorem $allI$: **assumes** $B\ x$ **for** x **shows** $\forall\, x.\ B\ x$
theorem $allE$: **assumes** $\forall\, x.\ B\ x$ **shows** $B\ a$

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

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

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

Inductive definitions

Primitive definitions

Definitional approach: everything produced from first principles (of Higher-Order Logic, Set-Theory etc.)

Example: composition of relations

definition $comp :: (\alpha \Rightarrow \beta \Rightarrow bool) \Rightarrow (\beta \Rightarrow \gamma \Rightarrow bool) \Rightarrow \alpha \Rightarrow \gamma \Rightarrow bool$
  where $comp\ R\ S\ x\ z \leftrightarrow (\exists\, y.\ R\ x\ y \wedge S\ y\ z)$
theorem $compI$: $R\ x\ y \Longrightarrow S\ y\ z \Longrightarrow comp\ R\ S\ x\ z$
  unfolding $comp\text{-}def$ **by** $auto$

theorem $compE$: $comp\ R\ S\ x\ z \Longrightarrow (\bigwedge y.\ R\ x\ y \Longrightarrow S\ y\ z \Longrightarrow C) \Longrightarrow C$
  unfolding $comp\text{-}def$ **by** $auto$

Question: Can we avoid this redundancy?

Inductive definitions

Idea: the least predicate closed under user-specified rules (according to Knaster-Tarski)

Example: transitive-reflexive closure

inductive trcl **for** $R :: \alpha \Rightarrow \alpha \Rightarrow \mathit{bool}$
  where
    $\mathit{trcl}\ R\ x\ x$ **for** x
  $\mid$ $\mathit{trcl}\ R\ x\ z$ **if** $R\ x\ y$ **and** $\mathit{trcl}\ R\ y\ z$ **for** $x\ y\ z$

Derived rules based on internal definition:

$$
\begin{aligned}
&\mathit{trcl} \equiv \\
&\lambda R.\ \mathit{lfp}\ (\lambda p\ x_1\ x_2. \\
&\qquad\qquad (\exists\, x.\ x_1 = x \wedge x_2 = x) \vee \\
&\qquad\qquad (\exists\, x\ y\ z.\ x_1 = x \wedge x_2 = z \wedge R\ x\ y \wedge p\ y\ z))
\end{aligned}
$$

Non-recursive inductive definitions

Example (1): composition of relations (concise version)

inductive $comp$ **for** $R :: \alpha \Rightarrow \beta \Rightarrow bool$ **and** $S :: \beta \Rightarrow \gamma \Rightarrow bool$
where $comp\ R\ S\ x\ z$ **if** $R\ x\ y$ **and** $S\ y\ z$ **for** $x\ y\ z$

Example (2): logical connectives (imitating Coq)

inductive and **for** $A\ B :: bool$
where $and\ A\ B$ **if** A **and** B

inductive or **for** $A\ B :: bool$
where $or\ A\ B$ **if** A $\mid$ $or\ A\ B$ **if** B

inductive $exists$ **for** $B :: \alpha \Rightarrow bool$
where $exists\ B$ **if** $B\ a$ **for** a

(Example: Inductive definitions)

Conclusion

Summary

Advantages of native Pure/Isar rules:

- Scalable specifications
- Reduced complexity for formal proofs in
  1. proving / using the results
  2. structured Isar proofs / tactic scripts / internal proof objects

Consequences:

- Reduced formality — towards "logic-free reasoning"
- May have to unlearn predicate logic!

Related Work

- Proofs:
  - Continuation of well-known *Natural Deduction* concepts (Gentzen 1935, and others)
  - Common principles shared with λ-Prolog (Miller 1991)
- Statements:
  - Coherent logic (cf. Coquand, Bezem, dates back to Skolem)
  - Euclid's Elements (cf. Avigad)
- Definitions:
  - Inductive definitions in Coq, HOL, Isabelle etc. (many variations)