NICTA Advanced Course

Slide 1

Theorem Proving
Principles, Techniques, Applications

locales

Content

Slide 2

- ➜ Intro & motivation, getting started with Isabelle
- ➜ Foundations & Principles
  - Lambda Calculus
  - Higher Order Logic, natural deduction
  - Term rewriting
- ➜ **Proof & Specification Techniques**
  - Inductively defined sets, rule induction
  - Datatypes, recursion, induction
  - More recursion, Calculational reasoning
  - Hoare logic, proofs about programs
  - **Locales, Presentation**

Last Time

Slide 3

- ➜ Syntax and semantics of IMP
- ➜ Hoare logic rules
- ➜ Soundness of Hoare logic
- ➜ Verification conditions
- ➜ Example program proofs

Isar Is Based On Contexts

Slide 4

theorem $\bigwedge x.\ A \Longrightarrow C$
proof -
  fix x
  assume Ass: A
  ⋮
  from Ass **show** C ...
qed

x and Ass are visible inside this context

Beyond Isar Contexts

Slide 5

Locales are extended contexts

- ➜ Locales are **named**
- ➜ Fixed variables may have **syntax**
- ➜ It is possible to **add** and **export** theorems
- ➜ Locale expression: **combine** and **modify** locales

Context Elements

Slide 6

Locales consist of **context elements**.

fixes	Parameter, with syntax
assumes	Assumption
defines	Definition
notes	Record a theorem
includes	Import other locales (locale expressions)

Declaring Locales

Slide 7

Declaring **locale** (named context) loc:

locale loc =
$loc1$ + Import
fixes . . . Context elements
assumes . . .

Declaring Locales

Slide 8

Theorems may be stated relative to a named locale.

lemma (**in** loc) P [simp]: $proposition$
$proof$

- ➜ Adds theorem P to context loc.
- ➜ Theorem P is in the simpset in context loc.
- ➜ Exported theorem $loc.P$ visible in the entire theory.

Slide 9

DEMO: LOCALES 1

PARAMETERS MUST BE CONSISTENT!

- ➜ Parameters in **fixes** are distinct.
- ➜ Free variables in **assumes** and **defines** occur in preceding **fixes**.
- ➜ Defined parameters cannot occur in preceding **assumes** nor **defines**.

Slide 10

LOCALE EXPRESSIONS

Locale name:	n
Rename:	$e\ q_1 \ldots q_n$
	Change names of parameters in e.
Merge:	$e_1 + e_2$
	Context elements of e_1, then e_2.

- ➜ Syntax is lost after rename (**currently**).

Slide 11

Slide 12

DEMO: LOCALES 2

Normal Form of Locale Expressions

Locale expressions are converted to flattened lists of locale names.

- ➜ With full parameter lists
- ➜ **Duplicates removed**

Slide 13

Allows for **multiple inheritance**!

Instantiation

Move from **abstract** to **concrete**.

instantiate $label$**:** loc

Slide 14

- ➜ From chained fact $loc\ t_1 \ldots t_n$ instantiate locale loc.
- ➜ Imports all theorems of loc into current context.
  - Instantiates the parameters with $t_1 \ldots t_n$.
  - Interprets attributes of theorems.
  - Prefixes theorem names with $label$
- ➜ **Currently only works inside Isar contexts.**

Slide 15

Demo: Locales 3

Slide 16

Presentation

ISABELLE'S BATCH MODE

Slide 17

- ➜ used to process and check larger number of theories
- ➜ no interactive niceties (no sorry, no quick_and_dirty)
- ➜ controlled by file `ROOT.ML` and script set `isatool`
- ➜ can save state for later use (images)
- ➜ can generate HTML and LaTeX documentation

ISATOOL

```
isatool <tool> <options>
```

Slide 18

Get help with:

`isatool`	shows available tools
`isatool <tool> -?`	shows options for `<tool>`

Interesting tools:

`isatool mkdir`	create session directory
`make/makeall`	run make for directory/all logics
`usedir`	batch session (documents, HTML, session graph)
`document/latex`	run LaTeX for generated sources

GENERATING LaTeX FROM ISABELLE

```
<..>/isatool usedir -d pdf HOL <session>

   <..>/<session>/ROOT.ML
   <..>/<session>/MyTheory.thy
   <..>/<session>/document/root.tex
```

Slide 19

- ➜ In `ROOT.ML`:
  ```
  no\_document use_thy "MyLibrary";
  use_thy "MyTheory";
  ```
- ➜ In `document/root.tex`:
  - include Isabelle style packages (isabelle.sty, isabellesym.sty)
  - include generated files
    `session.tex` (for all theories) or
    `MyTheory.tex`

DEMO: EXAMPLE

Slide 20

LARGE DEVELOPMENTS

Creating Images:

```
<..>/<session>/isatool usedir -b HOL <session>

   <..>/<session>/ROOT.ML
   <..>/<session>/MyLibrary.thy
```

Slide 21

- ➜ Processes `ROOT.ML`
- ➜ Saves state after processing in `~/isabelle/heaps/<ML-system>/HOL-<session>`
- ➜ Makes `HOL-<session>` available as logic in menu Isabelle→Logics
- ➜ Direct start of Isabelle with new logic: `Isabelle -l HOL-<session>`

MARKUP COMMANDS

- ➜ document structure commands:

  header section subsection subsubsection
  (meaning defined in `isabelle.sty`)

- ➜ normal text

  text $\{*\ldots*\}$ **text_raw** $\{*\ldots*\}$

Slide 22

- ➜ text inside proofs

  txt $\{*\ldots*\}$ **txt_raw** $\{*\ldots*\}$

- ➜ formal comments

  `--` $\{*\ldots*\}$

- ➜ make text invisible:

  $(*<*)\ \ldots\ (*>*)$

ANTIQUOTATIONS

Inside LaTeX you can go back to Isabelle commands and syntax.

Useful Antiquotations:

Slide 23

`@{typ` τ`}`	print type τ
`@{term` t`}`	print term t
`@{prop` ϕ`}`	print proposition ϕ
`@{prop [display]` ϕ`}`	print proposition ϕ with linebreaks
`@{prop [source]` ϕ`}`	check proposition ϕ, print its input
`@{thm` a`}`	print fact a
`@{thm` a `[no_vars]}`	print fact a, fixing schematic variables
`@{thm [source]` a`}`	check availability of a, print its name
`@{text` s`}`	print uninterpreted text s

WRITING ABOUT ISABELLE THEORIES

To document definitions and proofs:

- ➜ put comments explanations directly in original theory
- ➜ keep explanations short and to the point
- ➜ formal definitions, lemmas, syntax should speak for themself

Slide 24

To write a paper/thesis **about** a formal development

- ➜ use a separate theory/document on top of the development
- ➜ only talk about the interesting parts
- ➜ use antiquoations for theorems and definitions
- ➜ use extra locales, definitions, syntax for polish
- ➜ make full proof document available separately

POLISH

Know your audience. Use the right notation.

Slide 25

➜ Change LaTeX symbol interpretations

```
\renewcommand{\isasymLongrightarrow}
             {\isamath{\longrightarrow}}
```

➜ Declare special LaTeX output syntax:
syntax (latex) Cons :: "'a ⇒ 'a list ⇒ 'a list" ("_ ·/ _" [66,65] 65)

➜ Use translations to change output syntax:
syntax (latex) notEx :: "('a ⇒ bool) ⇒ bool" (binder "\<notex>" 10)
translations "\<notex>x. P" <= "¬(∃x. P)"

in document/root.tex:

```
\newcommand{\isasymnotex}{\isamath{\neg\exists}}
```

USING LOCALES

making large developments more accessible

Math textbook:

Let $(A, \cdot, 0)$ in the following be a group with $x \cdot y = y \cdot x$

Isabelle:

Slide 26

➜ Use locales to formalize contexts

➜ Antiquotations are sensitive to current locale context

➜ **Example:**
locale agroup = group + **assumes** com: "$x \cdot y = y \cdot x$"
...
$(* < *)$ **lemma** (in agroup) True $(* > *)$
txt $\{* \ldots *\}$
$(* < *)$ **oops** $(* > *)$

Slide 27

DEMO

WE HAVE SEEN TODAY ...

Slide 28

➜ Locale Declarations + Theorems in Locales

➜ Locale Expressions + Inheritance

➜ Locale Instantiation

➜ Generating LaTeX

➜ Writing a thesis/paper in Isabelle

EXERCISES

➜ No Exercise Today

Slide 29

Theorem Proving
Principles, Techniques, Applications

The End