COMP 4161

NICTA Advanced Course

Advanced Topics in Software Verification

Gerwin Klein

Formal Methods

locales

Content

- ➜ 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

- Hoare logic rules
- Soundness of Hoare logic
- Verification conditions
- Example program proofs

Isar Is Based On Contexts

theorem $\bigwedge x.\ A \Longrightarrow C$

proof -

fix x	
assume Ass: A	
⋮	x and Ass are visible
from Ass **show** C . . .	inside this context

qed

Beyond Isar Contexts

Locales are extended contexts

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

Context Elements

Locales consist of **context elements**.

fixes	Parameter, with syntax
assumes	Assumption
defines	Definition
notes	Record a theorem

Declaring Locales

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

locale loc =

$loc1$ + Import

fixes . . . Context elements

assumes . . .

Declaring Locales

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.

Demo: Locales 1

Parameters Must Be Consistent!

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

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**).

Demo: Locales 2

Normal Form of Locale Expressions

Locale expressions are converted to flattened lists of locale names.

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

Allows for **multiple inheritance**!

Instantiation

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

interpretation label: loc [”parameter 1” . . . ”parameter n”]

- ➜ Instantiates locale **loc** with provided parameters.
- ➜ Imports all theorems of **loc** into current context.
  - Instantiates theorems with provided parameters.
  - Interprets attributes of theorems.
  - Prefixes theorem names with **label**
- ➜ version for local Isar proof: **interpret**

Demo: Locales 3

PRESENTATION

Isabelle's Batch Mode

- ➜ 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>
```

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
```

➜ 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

Creating Images:

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

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

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

Markup Commands

- document structure commands:

  header **section** **subsection** **subsubsection**

  (meaning defined in `isabelle.sty`)

- normal text

  text {*...*} **text_raw** {*...*}

- text inside proofs

  txt {*...*} **txt_raw** {*...*}

- formal comments

  -- {*...*}

- make text invisible:

  (* < *) ... (* > *)

Antiquotations

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

Useful Antiquotations:

`@{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

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.

- 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:

- ➜ Use locales to formalize contexts
- ➜ Antiquotations are sensitive to current locale context
- ➜ **Example:**

  locale agroup = group + **assumes** com: "$x \cdot y = y \cdot x$"

  ...

  text (in agroup) $\{*...*\}$

- ➜ **More Examples**: `http://afp.sf.net`

DEMO

We have seen today ...

- ➜ Locale Declarations + Theorems in Locales
- ➜ Locale Expressions + Inheritance
- ➜ Locale Instantiation
- ➜ Generating LaTeX
- ➜ Writing a thesis/paper in Isabelle