Export of formal theory content in Isabelle/Scala

Makarius Wenzel
https://sketis.net

November 2019

`Isabelle/f7a9889068ff`

Abstract

This is an overview of the state of affairs of systematic export of formal theory content. The general motivation is to provide semantically enriched views on the Isabelle/AFP library, without requiring a running Isabelle process. A typical application could be search over a database of digested theories, e.g. for an AFP web service.

Introduction

Motivation

Aims:

- systematic support for export artefacts
- standard export of theory content
- optional export of proof terms

Note:

- raw file-systems are bad: stateful, fragile, not scalable
- Isabelle is not a file-oriented "compiler" (in contrast to `coqc`), but a session-oriented document processor
- proper data storage via session databases (SQLite, PostgreSQL, Scala/PIDE process)

Examples (1)

Export via batch build:

- `isabelle build -o export_theory FOL-ex`
- `isabelle export -l FOL-ex`
- `isabelle export -x '*:**' FOL-ex`

Export via `export_files` in `ROOT`:

- `isabelle build -e -d '$AFP' Buchi_Complementation`
- see `$AFP/Buchi_Complementation/ROOT`
  and `$AFP/Buchi_Complementation/Complementation_Build.thy`

Export via generated files:

- commands **generate_file** and **export_generated_files**
- e.g. see `$ISABELLE_HOME/src/Tools/Haskell/Haskell.thy`

Examples (2)

Export via headless PIDE process:

- `isabelle dump FOL-ex`

Export via PIDE editor (Isabelle/jEdit):

- URL `isabelle-export:`
- action `isabelle-export-browser`

export_code *List.filter List.map* **in** *SML*

Export in Isabelle/ML:

ML $\langle \mathit{Export.export}$ ***theory*** ***path_binding*** $\langle \mathit{test/a} \rangle$ $(\mathit{XML.blob}\ [\mathit{aaa}])\rangle$
ML $\langle \mathit{Export.export}$ ***theory*** ***path_binding*** $\langle \mathit{test/b} \rangle$ $(\mathit{XML.blob}\ [\mathit{bbb}])\rangle$

Semantic theory export

Aspects of `isabelle dump`:

- `latex`: generated LaTeX source (unused)
- `markup`: PIDE markup for theory sources
- `messages`: PIDE messages for commands (warning, error, . . .)
- `theory`: theory content (types, consts, thms, . . .)

Options for `isabelle build`:

- `export_document`
- `export_theory`
- `export_standard_proofs`
- `export_proofs`
- `prune_proofs`

Summary of concepts

Theory presentation

- arbitrary presentation functions over finished theory node
- access to theory for each commands (with source, positions)

Generated files

- named blobs within theory context
- small-scale content

Exported files

- named blobs associated with session
- scalable ML interface via `XML.body`, `XML.blob`
- scalable Scala storage via XZ compression and session database

Applications

Isabelle/MMT

Collaboration:

- Michael Kohlhase and Florian Rabe, Erlangen
- https://github.com/UniFormal/MMT

Approach:

- output of linear OMDoc and relational XML/RDF
- headless PIDE session (like `isabelle dump`)
- direct connection of Isabelle/Scala with `mmt.jar`
- bypass concrete syntax

Status:

- Nov-2018: https://sketis.net/2018/isabelle-mmt-export-of-isabelle-theories-and-import-as-omdoc-content
- Jun-2019: https://sketis.net/2019/mmt-as-component-for-isabelle2019
- Nov-2019: more stable, more scalable, more content

Isabelle/MMT content (Nov-2019)

- material: Isabelle + AFP (including `large`, `slow`, `very_slow`)
- repository versions: Isabelle/5ed8c7e826a2, AFP/1a0be182fdda
- 651 sessions, 6,839 theories, 135 MB theory text
- 5,080 locales, including 1,185 type classes
- 1,974,039 individuals:
  11,537 `type`, 234,375 `const`, 230,520 `axiom`, 1,459,926 `thm`
- 210,055,826 relations, including 196,081,510 dependencies
- 211 MB OMDoc output (XZ-compressed by factor $\approx$ 200)
- 42.8 GB uncompressed XML
- resource usage: 50 GB memory, 8 CPU cores, 18h20 elapsed time

Isabelle/Dedukti

Collaboration:

- Frédéric Blanqui and Michael Färber, Paris Cachan
- https://github.com/Deducteam/isabelle_dedukti

Approach:

- output primitive logic as huge λ-term in Dedukti
- batch-build with suitable export options
- generate Dedukti source ($\approx$ 3 formats)
- not scalable (yet)

Status:

- export works until $\approx$ `HOL.List` (problems with `HOL.Enum`)
- import works until $\approx$ `HOL.Nat` (problems with parsing)

Conclusion

Scalability

PIDE session:

- less scalable: more overhead
- more flexible
- more content
- occasional instabilities of Poly/ML and Java/VM (due to massive amount of material)
- 5 slices for Isabelle + AFP

Batch session:

- more scalable
- less flexible session/theory arrangements
- less content: no PIDE markup

Further applications

- connecting provers (e.g. OpenTheory output, Dedukti output)
- exploring libraries (e.g. Web search over Isabelle/AFP)
- refactoring via PIDE markup (e.g. update of term notation)
- refactoring via detailed dependencies (for sources and terms)
- data mining / machine learning of formal content
- external proof checking (open problems of scalability)