Creating Example ProofPower Databases

Abstract

This document contains the makefile, plus its associated files, that produces example databases for ProofPower. The “Interface” section describes the creation and use of this makefile from a user’s point of view.
1 DOCUMENT CONTROL

1.1 Contents List

1 DOCUMENT CONTROL ................................................................. 2
  1.1 Contents List ................................................................. 2
  1.2 Document Cross References .................................................. 3
  1.3 Changes History ............................................................... 3
  1.4 Changes Forecast ............................................................. 4

2 GENERAL ............................................................................. 5
  2.1 Scope .............................................................................. 5
  2.2 Introduction ....................................................................... 5
  2.3 Interface ........................................................................... 7

3 START OF MAKEFILE ............................................................... 10

4 PRELIMINARIES ................................................................... 10
  4.1 Macros .............................................................................. 10
  4.2 Default ............................................................................. 10
  4.3 Pattern Matching Rules ......................................................... 11

5 PREPARING CLEAN DATABASES ............................................... 12

6 MAKING THE TUTORIAL MANUAL (usr004) .................................. 13

7 MAKING THE Z TUTORIAL MANUAL (usr011) ................................ 14
  7.1 Z Tutorial Source Files ......................................................... 14
  7.2 Z Tutorial Make Entries ......................................................... 14

8 MAKING THE HOL TUTORIAL MANUAL (usr013) ....................... 15
  8.1 HOL Tutorial Source Files ..................................................... 15
  8.2 HOL Tutorial Make Entries ................................................... 15

9 MAKING TUTORIAL OVERHEADS (usr022) ................................ 16

10 MAKING Z TUTORIAL OVERHEADS (usr023) ........................... 17

11 MAKING HOL FORMALISED (spc001) ..................................... 17

12 MAKING HOL SEMANTICS (spc002) ........................................ 17

13 MAKING MODAL LOGIC (wrk022) ........................................... 17
1.2 Document Cross References


1.3 Changes History

Issues 1.1-1.3 (To 10th March 1993) Initial versions.

Issue 1.4 (22nd September 1993) Addition of new HOL tutorial, changed to work with new release mechanism.

Issue 1.5 (23rd September 1993) Reordering of dependencies for all_hol, addition of spc001.th, spc002.th
**Issue 1.6 (28th September 1993)** Change from usr011x.sml, usr013x.sml to usr011x.doc, usr013x.doc

**Issue 1.9 (25th August 1994)** Changed $INSTALLDIR to be $PPPPINSTALLDIR.

**Issue 1.15** Copyright and banner updates for open source release.

**Issue 1.16** PPHol-specific updates for open source release

**Issue 1.17** The demo scripts can now be run as tests by the hol and zed make files.

### 1.4 Changes Forecast

None.
2 GENERAL

2.1 Scope

This document provides documentation, a makefile and auxiliary scripts for the process of producing certain example databases for a ProofPower user.

2.2 Introduction

This document contains the literate script of a makefile (wrk051.mkf) and a number of auxiliary files which are scripts for other programs. This material allows the creation of example databases containing ProofPower tutorial and case study material. The user’s interface to this material is given in section 2.3.

To produce the makefile the program sieve is used to process this document, with the sml option. The result is the creation of a makefile, together with a number of small utility-files that are inputs for ProofPower. Most the the makefile entries are concerned with the creation or updating of various example databases.

The commentary upon the makefile assumes some knowledge of the UNIX make command. The key item of knowledge is the form and meaning of a standard makefile entry.

Example makefile entry

```
|target1  target2  :  dependency1  dependency2  dependency3
|    | command1
|    | command2
|    | command3
|    | command4
```

To use such an entry in a makefile the user might type:

```
csh
|make -f makefile target1
```

If so, make will first recursively use the rest of the makefile to do what processing is necessary for each of the dependencyi (this may be nothing). It will then determine whether target1 exists and has a “time of last change” after the “time of last change” of each dependencyi. If so make will halt. If not, then it will execute each of the commandi in sequence, in different shells, and then halt.

It is the above make algorithm, in combination with a single file (a database) containing all the effect of cumulative loads in ProofPower, that leads to the system of “flags” in the makefile entries in section 5 and beyond.
See UNIX documentation[7] for further details, such as a description of macros and pattern matching rules.
2.3 Interface

In the following descriptions the text marked by:

csh

|marked text|

may be cut and pasted without change from the source file of this document (wrk051.doc) into a SunView shell tool or command tool.

To produce the example databases, start in a directory with the necessary files of:

```
(1) usr004.doc
(2) usr011A.doc usr011B.doc usr011C.doc usr011D.doc
(3) usr011E.doc usr011T.doc usr011S.doc usr011X.doc
(4) usr013A.doc usr013B.doc usr013C.doc usr013D.doc usr013E.doc
(5) usr013F.doc usr013G.doc usr013H.doc usr013S.doc usr013X.doc
(6) usr022_slides.doc usr023_slides.doc
(7) wrk022.doc wrk043.doc wrk044.doc
(8) wrk051.doc (this file)
```

These should all be present in the subdirectory `docs` of the `ProofPower` installation directory. However, it is better to work with copies of these files, in a new directory, e.g. `examples`. If using a `ProofPower` installation made in the manner suggested in [2] then copies may be made by:

csh

```
cd $PPINSTALLDIR
mkdir examples
```

```
cd $PPINSTALLDIR/examples
cp usr004.doc usr011A.doc usr011B.doc usr011C.doc usr011D.doc
    usr011E.doc usr011T.doc usr011S.doc usr011X.doc
    usr013A.doc usr013B.doc usr013C.doc usr013D.doc usr013E.doc
    usr013F.doc usr013G.doc usr013H.doc usr013S.doc usr013X.doc
    usr022_slides.doc usr023_slides.doc wrk022.doc wrk043.doc wrk044.doc
    wrk044.doc wrk051.doc $PPINSTALLDIR/examples
```

Then create the makefile `wrk051.mkf` by:

csh

```
cd $PPINSTALLDIR/examples
sieve sml < wrk051.doc
```
Having produced the makefile (\texttt{wrk051.mkf}) and utilities, as above, there are then several options for further commands.

To print this document (assuming \texttt{pстex} without arguments sends output to the desired printer):

\begin{verbatim}
csh
| make -f wrk051.mkf wrk051.prt
\end{verbatim}

To create a child \texttt{ProofPower} database containing the effect of loading the tutorial scripts of \texttt{usr004} [1]:

\begin{verbatim}
csh
| make -f wrk051.mkf usr004.tutorial
\end{verbatim}

The resulting database can be invoked by:

\begin{verbatim}
hol -d example_hol
\end{verbatim}

Similarly for \texttt{usr011}[3], creating a database suitable for loading the \texttt{ProofPower-Z} exercises and trying solutions for them, by hand:

\begin{verbatim}
csh
| make -f wrk051.mkf usr011.tutorial
\end{verbatim}

The resulting database can be invoked by:

\begin{verbatim}
zed -d example_zed
\end{verbatim}

Similarly for \texttt{usr013}[4], creating a database suitable for loading the \texttt{ProofPower-HOL} exercises and trying solutions for them, by hand:

\begin{verbatim}
csh
| make -f wrk051.mkf usr013.tutorial
\end{verbatim}

The resulting database can be invoked by:

\begin{verbatim}
zed -d example_hol
\end{verbatim}

Similar entries also exist for \texttt{usr022}[5] and \texttt{usr023}[6], rather than \texttt{usr004}.

\begin{verbatim}
csh
| make -f wrk051.mkf usr022.tutorial
| make -f wrk051.mkf usr023.tutorial
\end{verbatim}
To create a database containing the effect of loading the exercise scripts for usr011 (which will also run the tutorial scripts for usr011, if not already done):

csh

|make −f wrk051.mkf usr011.exercises|

To create a database containing the effect of loading the solutions scripts for usr011 (which will also run the exercises, if not already done):

csh

|make −f wrk051.mkf usr011.solutions|

To create a database containing the effect of loading the exercise scripts for usr013 (which will also run the tutorial scripts for usr013, if not already done):

csh

|make −f wrk051.mkf usr013.exercises|

To create a database containing the effect of loading the solutions scripts for usr011 (which will also run the exercises, if not already done):

csh

|make −f wrk051.mkf usr011.solutions|

To tidy up afterwards, deleting all created databases and files, execute:

csh

|make −f wrk051.mkf clean_examples|

To remove all auxiliary files created by “sieving” this one:

csh

|rm −f wrk051.mkf \ |wrk051.delete_theories_usr004.sml \ |wrk051.delete_theories_usr011.sml \ |wrk051.delete_theories_zexercises.sml \ |wrk051.delete_theories_usr013.sml \ |wrk051.delete_theories_exercises.sml \ |wrk051.delete_theories_spec001.sml \ |wrk051.delete_theories_spec002.sml \ |wrk051.delete_theories_usr022.sml \ |wrk051.delete_theories_usr023.sml \ |wrk051.delete_theories_wrk022.sml \ |wrk051.delete_theories_wrk043.sml \ |wrk051.delete_theories_wrk044.sml \ |wrk051.delete_theories_wrk050.sml|
The file `pp_install`, to be found in the installation directory, copies all the files needed into a new directory and runs all the ProofPower scripts to generate the two databases (`arch/example_zed.db`) and (`arch/example_hol.db`). It copies these two databases into the `arch/bin` directory, and cleans up afterwards. This file needs to be run from the release directory.

It is advised that the `changeParent` command is run on the new databases, the new parents being `sun4pp_hol.db` and `sun4pp_zed`, found in the subdirectory `sun4bin`. The script `pp_install` does this.

3 START OF MAKEFILE

```makefile
# wrk051.mkf
# A makefile that produces example databases.
# Created from: /Users/rda/pp/opp/bld/RCS/wrk051.doc,v 1.17 2008/02/11 11:35:34 rda
```

4 PRELIMINARIES

4.1 Macros

The first two macros below define the base names of the example databases.

```makefile
EHOLDB = example_hol
EZEDDB = example_zed
```

The following are the comments returned at the end of a successful or failed make:

```makefile
SUCCESS_MESSAGE = "make command succeeded"
FAILURE_MESSAGE = "make command failed"
```

4.2 Default

The default action of the makefile is to print a message. The default is indicated by it being the first target in the makefile.
Text appended to file wrk051.mkf

default:
    @echo Use:
    @echo "make -f wrk051.mkf all"
    @echo to create example ProofPower databases for HOL and Z.
    @echo See document DS/FMU/IED/WRK051 for further details.

4.3 Pattern Matching Rules

See make documentation for how the following work. They provide the “standard” processing rules for:

.doc - documents,
.sml - Standard ML files, usually derived from documents,
.tex - \LaTeX files derived from documents,
.dvi - files displayable by dvipage or printable by pstex,
.prt - dummy files, used to invoke printing.

Text appended to file wrk051.mkf

.SUFFIXES: .doc .sml .tex .dvi .prt
%.tex: %.doc
doctex $*

%.dvi: %.tex
texdvi $*
    - bibtex $*
texdvi $*
texdvi $*

%.sml: %.doc
docsml $*

%.prt: %.dvi
pstex $*
5 PREPARING CLEAN DATABASES

The two flags, \texttt{wrk051\_hol\_flag} and \texttt{wrk051\_zed\_flag} are two empty files, whose “time of last change” is the time of creation of the example databases. Various items later in this makefile have these flags amongst their dependencies. This means that, amongst other reasons for reloading, such items are reloaded if their previous load was made prior to the creation of the appropriate example database. If the dependencies of these items were the database filenames themselves, then a reload would be needed on any change to the databases, such as another, orthogonal, load. However, the databases themselves will still be recreated if the flags are used as dependencies when they are out of step with their parents (or are not there at all!).
6 MAKING THE TUTORIAL MANUAL (usr004)

The following pattern of file creation and makefile entry is used in most of the following sections, though we only comment upon it in this one.

The following command creates a file: \texttt{wrk051\_delete\_theories\_usr004.sml} that is used to delete, if present, the theories that would be created by loading \texttt{usr004.sml}. If the theories were present the load of \texttt{usr004.sml} would fail. A similar file is created for each of the items that may be added into the example databases.

\begin{verbatim}
Text dumped to file wrk051_delete_theories_usr004.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory ”Peanissimo” handle Fail _ => ());
(force_delete_theory ”ordinals” handle Fail _ => ());

The following creates, if necessary, an example HOL database (via \texttt{wrk051\_hol\_flag}). It then creates an .sml file from \texttt{usr004.doc}, loads the above theory-deleting file into the example database, and then loads this formal text of \texttt{usr004.doc}. It then saves the database with the results of the above loads. Finally, it touches a flag, \texttt{usr004\_tutorial}, so that the “time of last change” for the flag is the time of last loading \texttt{usr004.sml}.

Text appended to file wrk051.mkf

\texttt{usr004.tutorial}: \texttt{wrk051\_hol\_flag usr004.doc usr004.sml}
\texttt{hol -d \$(EHOLDB) \(-f\) wrk051\_delete\_theories\_usr004 \(-f\) usr004}
\texttt{touch usr004.tutorial}
\end{verbatim}
7 MAKING THE Z TUTORIAL MANUAL (usr011)

7.1 Z Tutorial Source Files

<table>
<thead>
<tr>
<th>usr011.*</th>
<th>top level document</th>
</tr>
</thead>
<tbody>
<tr>
<td>usr011A.*</td>
<td>Ch. 2 - Z Language</td>
</tr>
<tr>
<td>usr011B.*</td>
<td>Ch. 3 - Z Predicate Calculus</td>
</tr>
<tr>
<td>usr011C.*</td>
<td>Ch. 4 - Z Expressions</td>
</tr>
<tr>
<td>usr011D.*</td>
<td>Ch. 5 - Z Paragraphs</td>
</tr>
<tr>
<td>usr011E.*</td>
<td>Ch. 6 - An Example Specification and Proof</td>
</tr>
<tr>
<td>usr011F.*</td>
<td>Ch. 7 - The Z ToolKit</td>
</tr>
<tr>
<td>usr011X.*</td>
<td>Exercises</td>
</tr>
<tr>
<td>usr011S.*</td>
<td>Solutions</td>
</tr>
</tbody>
</table>

7.2 Z Tutorial Make Entries

Text dumped to file wrk051_delete_theories_usr011.sml

```sml
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)

(force_delete_theory ”usr011” handle Fail _ => ());
```

Text dumped to file wrk051_delete_theories_exercises.sml

```sml
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)

(force_delete_theory ”z_exercises_1” handle Fail _ => ());
(force_delete_theory ”z_exercises_2” handle Fail _ => ());
(force_delete_theory ”z_exercises_3” handle Fail _ => ());
(force_delete_theory ”z_exercises_4” handle Fail _ => ());
```
8 MAKING THE HOL TUTORIAL MANUAL (usr013)

8.1 HOL Tutorial Source Files

<table>
<thead>
<tr>
<th>usr013.*</th>
<th>top level document</th>
</tr>
</thead>
<tbody>
<tr>
<td>usr013A.*</td>
<td>Ch. 2 - Introduction</td>
</tr>
<tr>
<td>usr013B.*</td>
<td>Ch. 3 - HOL type system</td>
</tr>
<tr>
<td>usr013C.*</td>
<td>Ch. 4 - HOL terms</td>
</tr>
<tr>
<td>usr013D.*</td>
<td>Ch. 5 - Theories</td>
</tr>
<tr>
<td>usr013E.*</td>
<td>Ch. 6 - Forward proof</td>
</tr>
<tr>
<td>usr013F.*</td>
<td>Ch. 7 - Goal Oriented proof</td>
</tr>
<tr>
<td>usr013G.*</td>
<td>Ch. 8 - Predicate calculus</td>
</tr>
<tr>
<td>usr013H.*</td>
<td>Ch. 9 - Induction</td>
</tr>
<tr>
<td>usr013X.*</td>
<td>Exercises</td>
</tr>
<tr>
<td>usr013S.*</td>
<td>Solutions</td>
</tr>
</tbody>
</table>

8.2 HOL Tutorial Make Entries

Text dumped to file wrk051.delete_theories_usr013.sml

```
(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "usr013" handle Fail _ => ());
```
9 MAKING TUTORIAL OVERHEADS (usr022)

Text dumped to file wrk051_delete_theories_usr022.sml

Text appended to file wrk051.mkf

usr022.tutorial: wrk051_hol_flag usr022_slides.doc usr022_slides.sml
    hol -d $(EHOLDB) -f wrk051_delete_theories_usr022.sml -f usr022_slides
    touch usr022.tutorial
10 MAKING Z TUTORIAL OVERHEADS (usr023)

Text dumped to file wrk051_delete_theories_usr023.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory ”usr023” handle Fail _ => ());

Text appended to file wrk051.mkf

usr023.tutorial: wrk051_zed.flag usr023_slides.doc usr023_slides.sml
    zed -d $(EZEDDB) -f wrk051_delete_theories_usr023.sml -f usr023_slides
    touch usr023.tutorial

11 MAKING HOL FORMALISED (spc001)

Text dumped to file wrk051_delete_theories_spc001.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory ”spc001” handle Fail _ => ());

Text appended to file wrk051.mkf

spc001.th: wrk051_hol.flag spc001.doc spc001.sml
    hol -d $(EHOldb) -f wrk051_delete_theories_spc001.sml -f spc001.sml
    touch spc001.th

12 MAKING HOL SEMANTICS (spc002)

Text dumped to file wrk051_delete_theories_spc002.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory ”spc002” handle Fail _ => ());

Text appended to file wrk051.mkf

spc002.th: wrk051_hol.flag spc002.doc spc002.sml
    hol -d $(EHOldb) -f wrk051_delete_theories_spc002.sml -f spc002.sml
    touch spc002.th

13 MAKING MODAL LOGIC (wrk022)

Text dumped to file wrk051_delete_theories_wrk022.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory ”wrk022” handle Fail _ => ());
14 MAKING RAMSEYS THEOREM (wrk043)

Text dumped to file wrk051_delete_theories_wrk043.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "ramsey" handle Fail _ => ());

Text appended to file wrk051.mkf

| wrk043.th: wrk051_hol_flag wrk043.doc wrk043.sml |
| hol -d $(EHOLDB) -f wrk051_delete_theories_wrk043.sml -f wrk043.sml |
| touch wrk043.th |

15 MAKING FINITENESS THEOREMS (wrk044)

Text dumped to file wrk051_delete_theories_wrk044.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "fin_thms" handle Fail _ => ());

Text appended to file wrk051.mkf

| wrk044.th: wrk051_hol_flag wrk044.doc wrk044.sml |
| hol -d $(EHOLDB) -f wrk051_delete_theories_wrk044.sml -f wrk044.sml |
| touch wrk044.th |

16 MAKING FINITENESS THEOREMS (wrk046)

Text dumped to file wrk051_delete_theories_wrk046.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)
(force_delete_theory "lib_thms" handle Fail _ => ());
17 MAKING CASE STUDY (wrk050)

Text dumped to file wrk051.delete_theories_wrk050.sml

(* created from wrk051.doc %Z% 1.17 wrk051.doc,v 2008/02/11 11:35:34 *)

(force_delete_theory "wrk050" handle Fail => ());

Text appended to file wrk051.mkf

wrk050.th: wrk051_zed.flag wrk050.doc wrk050.sml
    zed -d $(EZEDDB) -f wrk051_delete_theories_wrk050.sml -f wrk050.sml
    touch wrk050.th

wrk050.dvi is created by the standard rule for .dvi files.

18 CREATE ALL DATABASES

The following create and populates the example HOL and Z databases.

Text appended to file wrk051.mkf

all_hol: spc001.th spc002.th usr004.tutorial usr013.exercises \ 
    wrk022.th wrk044.th wrk043.th wrk046.th
    @echo All module tests passed.

all_zed: usr011.zexercises wrk050.th
    @echo All module tests passed.

all : all_hol all_zed

all can be “undone” by clean_examples.
19 TIDYING UP THE FILE STORE

Clean out all example database material that might be created during an execution of this makefile:

```bash
Text appended to file wrk051.mkf

clean_examples :
    rm −f $(EHOLDB) $(EZEDDB)
    rm −f wrk051_hol.flag wrk051_zed.flag
    rm −f usr004.tutorial usr022.tutorial usr023.tutorial
    rm −f usr011.tutorial usr011_exercises usr011_solutions
    rm −f usr013.tutorial usr013_exercises usr013_solutions
    rm −f spc001.th spc002.th
    rm −f wrk022.th wrk043.th wrk044.th wrk046.th wrk050.th
    rm −f usr022_slides.sml usr023_slides.sml usr004.sml
    rm −f usr011X.sml usr011S.sml
    rm −f usr013X.sml usr013S.sml wrk050.sml
```

To remove all auxiliary files created by “sieving” wrk051.doc (we refrain from having a makefile entry delete its own makefile!):

```bash
csh
| rm −f wrk051.mkf wrk051_*, *
```

20 make ENDINGS

The following text causes make to output the appropriate message upon finished processing with either a successful or failed “make”.

```bash
Text appended to file wrk051.mkf

.FAILED :
    @ echo "*** $(FAILURE_MESSAGE) ***" 1>&2
    exit 1

.DONE :
    @ echo "*** $(SUCCESS_MESSAGE) ***" 1>&2
```
21 INDEX

.DONE .................................................. 20
.FAILED .................................................. 20
all_hol ................................................. 19
all_zed .................................................. 19
all ......................................................... 19
clean_examples ................................. 20
spc001.th .................................. 17
spc002.th .................................. 17
usr004.tutorial ................................. 13
usr011.tutorial ................................. 15
usr011.exercises ...................... 15
usr011.solutions ...................... 15
usr011A.doc ................................. 12
usr011B.doc ................................. 12
usr011C.doc ................................. 12
usr011D.doc ................................. 12
usr011E.doc ................................. 12
usr011T.doc ................................. 12
usr013.exercises ...................... 16
usr013.solutions ...................... 16
usr013.tutorial ......................... 16
usr013A.doc ................................. 12
usr013B.doc ................................. 12
usr013C.doc ................................. 12
usr013D.doc ................................. 12
usr013E.doc ................................. 12
usr013F.doc ................................. 12
usr013G.doc ................................. 12
usr013H.doc ................................. 12
usr022.tutorial ......................... 16
usr023.tutorial ......................... 17
wrk022.th ................................. 18
wrk043.th ................................. 18
wrk044.th ................................. 18
wrk046.th ................................. 19
wrk050.th ................................. 19
wrk051_hol.flag ...................... 13
wrk051_zed.flag ...................... 13