PROD: A PROlog Documentation, and Delivery Tool

Tim Menzies

Lane Department of Computer Science, West Virginia University, PO Box 6109, Morgantown, WV, 26506-6109, USA;
http://tim.menzies.us; tim@menzies.us

Wp ref: ~menzies/src/pl/prod/prod.tex, January 5, 2003.

Abstract PROD, a Prolog documentation system, is motivated and described.

Contents

List of Figures

1 What?

PROD can be used to *document* or *deliver* a Prolog application:

Delivery: The current PROD distribution comes with a set of standard Prolog applications shown in Figure 1. These programs are written in the *PROD-compatible £le convention* (described in §2.2.1) which simpli£es using Prolog code from different programmers. PROD £les are valid Prolog code that can be loaded into a Prolog interpreter, without modi£cation.

Documentation: A PROD £le also contains LaTeX commands inside Prolog's comment characters; i.e. on a line after the % character or between *...*\ characters. That is, as a programmer writes their code they can add in comments which, subsequently, can be typeset.

The typeset form of a PROD document looks just like this document and includes a table of contents; a list of £gures; automatic numbering of sections, £gures, and citations. Also, all the Prolog code is displayed as verbatim text (in a typewriter font).

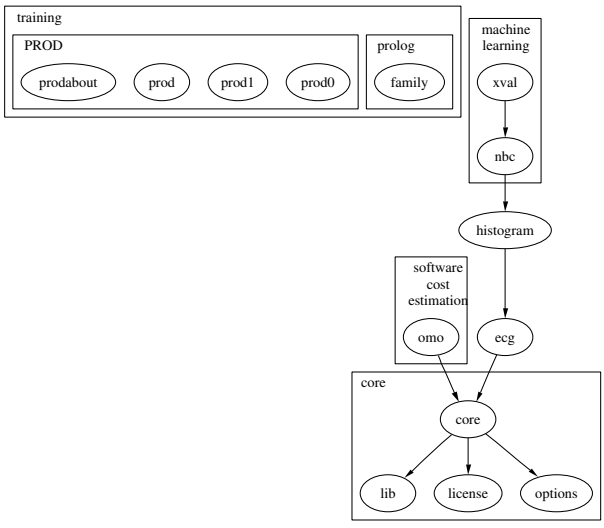

Fig. 1 Applications within the current release of PROD. $Y \rightarrow X$ indicates that Y has to £rst load X.

```
 1 /*\documentclass[twocolumn,global]{svjour}
 2 \usepackage{prod}\begin{document}
 3
 4 \theprogram{NAME}
 5 \thetocdepth{2}
 6 \thepapers{refs}
 7 \thewp{PATHNAME}
 8 \thetitle{TITLE}
 9 \theauthor{AUTHOR1\inst{1},AUTHOR2\inst{2}}
10 \theinstitute{WHERE AUTHOR1 WORKS;\\
11           \url{author1@email1.com},
12           \url{http://where.to.find.author1}
13              \and
14              WHERE AUTHOR2 WORKS}
15 \thereference{WVU, CSEE, AI lab memo \#3. Available from
16         \url{http://tim.menzies.com.pdf/03prod0.pdf}}
17 \theacknowledgement{ACKNOWLEDGEMENTS}
18 \theabstract{ONE PARAGRAPH SUMMARY}
19 */
20
21 %%%% SECTION1 heading
22 /*
23 BODY OF DOCUMENT WITH A REFERNCE~\cite{swiprolog}.
24 */
25 %\input{prod0a}
26 We can include text like that shown in \fig{prod0a.tex}.
27 \SRC{prod0a.tex}{A sample include file.}
28
29 /* Some text to be typeset
30 */
31 %%% SECTION2 heading
32 /* Some text to be typeset
33 */
34 %% SECTION3 heading
35 /* Some text to be typeset
36 */
37 %%%% SECTION1 heading %<
38 somePrologCode :-
39     subGoal1,
40     subGoal2.
41 %>
42 /* Some text between code.
43 */
44 %<
45 someMorePrologCode :-
46     subGoal1,
47     subGoal2.
48 %>
49 /*
50 \theend
51 \end{document}
52 */
```

Fig. 2 `prod0.pl`, a sample PROD £le.

PROD is distributed under the GNU General Public License. Every PROD document automatically includes that license as part of its appendix.

Figure 7, at the end of this document, lists other documents relating to PROD.

1.1 Inside a PROD £le

Figure 2 shows a sample PROD £le. When typeset, LaTeX converts this document to the PDF £le shown at `http://tim.menzies.com/pdf/prod0.pdf`. This £le contains a *header*, a *preamble*, a *body*, and a *footer*.

1.1.1 The header and footer A PROD £le begins with a standard *header*:

```
/*\documentclass[twocolumn,global]{svjour}
\usepackage{prod}\begin{document}
```

which starts a LaTeX document and loads the `prod.sty` style £le. The £le also ends with a standard *footer*:

```
/*
\theend
\end{document}
*/
```

1.1.2 The Premable In between the footer and the header there is a *preamble* and a *body*. The preamble de£nes certain key parameters of the £le using the following commands. For a detailed example of the use of these commands, see Figure 3.

`\theprogram{NAME}` : De£nes the `NAME` of the program being described. I use very short names for my programs (less than 3 letters).

`\thetocdepth{LEVEL}` : Controls how detailed is the table of contents. A `LEVEL=N`, the table of contents only includes down to level `N`. For very short tables of contents, use `N=1`.

`\theref{FILE}` : Shows the location of the £le `FILE.bib` which contains the citations for this £le. For those not familiar with LaTeX's citation system, Figure 4 shows a sample of the `refs.bib` database.

`\thewp{PATHNAME}` : Shows where to £nd the source code £le for this document.

`\thetitle{TITLE}` : De£nes the `TITLE` of the paper.

`\theauthor{AUTHOR1 \inst{1}, AUTHOR2\inst{2}}` : De£nes the `AUTHOR`s and maps those authors to their `INSTITUTIONS`.

`\theinstitute{WORK PLACE}` : De£nes where the `AUTHOR`s work. Multiple `INSTITUTIONS` are separated by "`\and`".

`\thereference{REFERENCE}` : Where this paper appears and where it can it be downloaded from.

`\theacknowledgement{ACKNOWLEDGEMENTS}` : Credit given where credit is due.

`\theabstract{ONE PARAGRAPH SUMMARY}` : A short summary of the paper.

Some of the above commands can be entered in a different order but, for safety's sake, it is best to use the above ordering for the preamble.

1.1.3 The Body The *body* of a PROD £le contains Prolog source code embedded in LaTeX commands. Within the body, the following conventions hold:

- Anything found between `%<` and `%>` is preserved as verbatim text (e.g. see lines 44 to 48 of Figure 2).
- A line starting with `%\command` is converted to `\command` (e.g. line 25 of Figure 2).
- A *level 1 heading* is declared for text found after `%%%%` (e.g. line 37 of Figure 2).
- A *level 2 heading* is declared for text found after `%%%` (e.g. line 31 of Figure 2).
- A *level 3 heading* is declared for text found after `%%` (e.g. line 34 of Figure 2).

In the case of level 1,2,3 headings:

```
\theprogram{PROD1}
\thetocdepth{2}
\therefs{refs}
\thewp{~menzies/src/pl/prod/prod0.tex}

\thetitle{An example of the {\PROD}\newline Prolog
            delivery and documentation system}

\theauthor{Tim Menzies\inst{1}, Sant A. Clause\inst{2}}

\theinstitute{Lane Department of Computer Science,
            University of West Virginia,
            PO Box 6109, Morgantown,
            WV, 26506-6109, USA;\\
            \url{http://tim.menzies.us};
            \url{tim@menzies.us}
            \and
           Artic Software Solutions:
             no ice cube too small, no iceberg too big;\\
            \url{http://north.pole/~santac};
            \url{santa@north.pole}
}

\thereference{WVU, CSEE, AI lab memo \#1, 2003.
           Available from
           \url{http://tim.menzies.com/pdf/03prod1.pdf}
}

\theacknowledgement{This research was conducted at

           West Virginia University under NASA
           contract NCC2-0979.
           The work was sponsored by the NASA
           Office of Safety and Mission Assurance
           under the Software Assurance Research
           Program led by the NASA IV\&V Facility.
           Reference herein to any specific
           commercial product, process, or
           service by  trade name, trademark,
           manufacturer, or otherwise, does not
           constitute or imply its endorsement
           by the United States Government.
}

\theabstract{This document is a minimal example of
           using the {\PROD} Prolog documentation and
           delivery  system.
}
```

Fig. 3 A sample PROD preamble from `prod1.pl`. The results of this preamble can be viewed at `http://tim.menzies.com/pdf/prod1.pdf`.

- There can be no characters to the left of the comment characters.
- If the line ends in `%<`, then the heading is declared and verbatim text starts straight after the heading.

Currently, PROD does not support headings levels greater than 3.

```
@Book{bratko01,
  Author =    "I. Bratko",
  Title =     "Prolog Programming for Artificial
                Intelligence. (third edition)",
  Publisher =     "Addison-Wesley",
  Year =      2001
}

@article{me89zb,
  author =    "T.J. Menzies",
  title =     "Domain-Specific Knowledge Representations",
  month =     "Summer",
  journal =   "AI Expert",
  year =      "1989",
}

@InProceedings{menz91,
  AUTHOR =    "T.J. Menzies",
  YEAR =      " 1991",
  TITLE =     "{ISA} {O}bject {PARTOF}
               {K}nowledge {R}epresentation (Part Two)?",
  BOOKTITLE =     " Tools Pacific 4",
  EDITOR =    " B. Meyer",
  Note =      "Available from
     \url{http://tim.menzies.com/pdf/tools91.pdf}"
}

@PhdThesis{me95,
  AUTHOR =    " T.J. Menzies",
  YEAR =      " 1995",
  TITLE =     " Principles for Generalised
                Testing of Knowledge  Bases",
  School =    " University of New South Wales",
  Note =      "Ph.D. thesis. Available from
     \url{http://tim.menzies.com/pdf/95thesis.pdf}"
}

@TechReport{me96c,
  Author =    "T. Menzies and P. Haynes",
  Title =     "Empirical Observations of Class-level
               Encapsulation and Inheritance",
  Institution =  "Department of Software Development,
                  Monash University",
  Year =      1996,
  Note =      "Available from
        \url{http://tim.menzies.com/pdf/96encap.pdf}"
}

@InCollection{mich90,
  author =    {R.S. Michalski},
  editor =    {B.G. Buchanan and D.C. Wilkins},
  booktitle =     {Reading in Knowledge
                  Acquisition and Learning},
  title =     {Toward a Unified Theory of Learning},
  publisher =     {Morgan Kaufmann},
  year =      1993,
  pages =     {7-38}
}

@unpublished{spinmanual,
  author = "{G}erard {J}. {H}olzmann",
  title = "{B}asic {SPIN} {M}anual",
  note = "{A}vailable at
  \url{http://cm.bell-labs.com/cm/cs/what/spin/Man/Manual.htm}
"}

@Manual{swiprolog,
  Title =     "SWI-Prolog",
  Author =    "Jan Wielemaker",
  Note =      "Available from
    \url{http://swi.psy.uva.nl/projects/xpce/SWI-Prolog.html}."
}
```

Fig. 4 A sample LaTeX citation database.

2 How?

2.1 Installing

PROD comes as one ¤at directory with lots of included £les. Email me at `tim@menzies.us` for that zip £le. Just unzip it into a fresh directory.

If you just want to run a PROD application, all you need is a Prolog interpreter. A PROD £le is a syntactically valid Prolog program that can be loaded into a Prolog interpreter without modi£cation.

On the other hand, if you want to use PROD to document your code, they you'll need a working LaTeX, Prolog and Perl installation. Most UNIX installations have all three. But if you need to get your own system going under Windows, then the software shown in Figure 5 might be useful.

2.2 How to load a PROD system

PROD assumes that £les come in a PROD-compatible format.

2.2.1 PROD-compatible applications A PROD-compatible Prolog system comprises several £les:

1. A main load £le called, say, `myfile.pl`. This main load £le loads up to three other £les.
2. `myfile0.pl`: a small set of pre-load actions.
3. `myfile1.pl`: the bulk of the code.
4. `myfile2.pl`: start-up actions to be performed after the the code is loaded.
5. A documentation £le called `myfile.pdf` auto-generated from `myfile.pl`.

2.2.2 Sample pre-load actions in `myfile0.pl`

- *Loads of other Prolog systems*: In the case where you are loading other PROD-compatible £les, then you'll have to carefully inspect the pre-load and start-up actions of the systems you are loading. In the best case, you can just load the main £les of the other PRODs. However, sometimes you have to skip loading those pre-load and start-up £les, but weave their actions in with your own pre-loads and start-ups.
- *Operator de£nitions*.
- *Flags* such as what predicates are `dynamic`.
- *Hooks into the Prolog reader*: such as 'goal_expansion/2' and 'term_expansion/2'.
- *Hacks*: those shameful things we can't avoid. So we keep separate from the rest of our beautiful code in a separate section. And we don't talk too much about them. So lets go to a new section.

2.2.3 Start-up actions in `myfile2.pl` These are application-speci£c and may include actions like loading con£guration £les, then some domain-speci£c assertions, then calling the main processing predicate of the system.

2.3 How to document a PROD system

2.3.1 Starting from scratch To start writing PROD code, copy the `template.pl` (which comes with the standard PROD distribution) and rename it to (e.g.) `yourfile.pl`. Once that is done, then two programs are required to convert your code into a PDF format.

$$yourfile.pl \overset{prep}{\longrightarrow} yourfile.tex \overset{\text{LaTeX}}{\longrightarrow} yourfilepdf$$

The `prep` and LaTeX programs are described below.

2.3.2 `Prep`*: converting* `*.pl` *to* `*.tex` The pre-processor `prep` converts the £le (e.g.) `prep0.pl` to `prep0.tex`. It is convenient to create a £le `preps` that lists all your £les that will need `prepping`. For example:

```
perl prep file1
prep prep file2
```

When executed, this script looks for (e.g.) a `file.pl` and `file2.pl` and generates the £les `file1.tex`, `file2.pl`. Note that during that translation,

- The characters `\*` and `*\` are deleted. Hence, the characters on (e.g.) line 47 and 51 of Figure 2 are deleted.
- A line starting with `%\command` is converted to `\command` (e.g. line 24 of Figure 2).

The resulting `*.tex` £le can then be converted to PDF using some LaTeX system.

2.3.3 LaTeX: converting `*.tex` *to* `*.pdf` On my UNIX system, the script `mytex` generates PDF £les from LaTeX £les, then copies it to my web site. The command line

$$mytex\ prod0\ 03prod0$$

takes Figure 2 and generates the £le that can be viewed at `http://tim.menzies.com/pdf/03prod0.pdf`.

Incidently, this £le is `http://tim.menzies.com/pdf/03prod.pdf` and was generated using the command line

$$mytex\ prod\ 03prod$$

2.3.4 Load order and documentation order Sometimes, the order in which you load £les into Prolog is *not* the order in which you want to explain an application. For example, consider an application containing some low-level support code. The support code may have to be loaded *£rst*, before the rest of the application can be loaded. However, in terms of motivating and explaining the application, you want to explain that support code *last*.

The solution to this problem is to separate the Prolog loads from the LaTeX loads. This technique is used in the `lib.pl` as follows. Note in the following code segment, the use of `\input{libX}` *after* the call to the Prolog `load_files`:

PERL:

–✓ Perl can be downloaded from many sources. For example, it comes as part of the *CYGWIN* distribution from `http://xfree86.cygwin.com`.

PROLOG:

–✓ Interpreters: SWI-Prolog `http://www.swi-prolog.org`.
–✓ Editors:
  – Some of my students speak highly of the *Prolog IDE* editor `http://www.bildung.hessen.de/abereich/inform/skii/material/swing/indexe.htm`.
  – I prefer *EMACS*, a Windows version of which can be found at `http://www.gnu.org/software/emacs/windows`
  –$ An excellent alternative to *EMACS* is *TEXTPAD*: `http://www.textpad.com/download/`. It has ignorable nag screens which can be removed for $27 (ish).
  – A simpler editor, that is free, and has a smaller footprint, is *PFE*. Its a very useful editor and it can be installed without super users. `http://www.lancs.ac.uk/people/cpaap/pfe/`.

LaTeX:

– Postscript processing
  –✓ *GHOSTSCRIPT* and *GSVIEW* are the core postscript processing utilities: `http://www.cs.wisc.edu/~ghost`.
– A LaTeX compiler:
  –✓ *MIKTEX* is a good Windows-based LaTeX distribution: `http://www.miktex.org`
  – LaTeX training material can be found in many places including `http://www.ling.upenn.edu/advice/latex.html`. For this page you can £nd the very excellent:
    • Quick start directions: `http://www.ling.upenn.edu/advice/latex/starting.html`
    • The Not So Short Introduction to LaTeX (highly recommended): `ftp://ftp.tex.ac.uk/tex-archive/info/lshort/english/lshort.pdf`. This document may also be found with the standard PROD distribution.
    • Guide to Including Graphics `http://www.ling.upenn.edu/advice/latex/grfguide.pdf`
– Editing LaTeX:
  –✓$ Under Windows, *WINEDT* is the recommended LaTeX editor: `http://www.winedt.com`. It has some ignorable nag screens which can be removed for $30 (ish).
– Viewing the output. LaTeX generates DVI £les, postscript £les, and Acrobat £les.
  –✓ DVI £les can be viewed using the *YAP* viewer that comes with *MIKTEX*.
  –✓ Postscript £les can be viewed using the *GSVIEW* program from `http://www.cs.wisc.edu/~ghost`.
  –✓ The Acrobat reader can be downloaded from `http://www.adobe.com/products/acrobat/readstep2.html`.
– Plotting scienti£c data:
  – The *GNUplot* utility from `http://www.gnuplot.vt.edu/` can generate postscript plots of scienti£c data.
– Drawing packages:
  –$ *MAYURA DRAW* is a vector drawing utility for creating SVG and EPS illustrations: `http://www.mayura.com/`. It can be used for free for 30 days (ish) then a registration must be bought for $30 (ish).
  – The amazingly useful, and very small, *jpeg2ps* converts any JPEG £le to an eps: `http://www.pdflib.com/jpeg2ps/`. Now, any graphic that can be converted to a JPEG can be EPS-ed and included into a LaTeX document.
  – And to convert anything to JPEG, use *IRFANVIEW*: `http://www.irfanview.com`
  – Finally, if you can't import it any other way, get it on the screen, screen sieze it with *SCREENSIZE* (`http://www.pcmag.com/article2/0,4149,10206,00.asp`, copy and paste it to *IRFANVIEW* then *jpeg2ps* it.
– Auto-layout of directed and undirected graphs:
  – *DOT*: The GRAPHVIZ distribution from Bell Labs contains the *DOT* graph layout and visualization tool: `www.research.att.com/sw/tools/graphviz`. *DOT* can generate EPS £les.
– Spell checking LaTeX:
  – The *ISPELL* checker is a good UNIX-based spell checker. Most UNIX installations integrate it with *EMACS*.
  – *WINDEDT* has a good editor.

Fig. 5 Support code for PROD, under Windows. For a minimal installation, only get the items marked with ✓. This software is freeware, except the items marked with $.

```
 1 latex $1 > /dev/null
 2 grep "Warning:" $1.log
 3 bibtex $1 > /dev/null
 4 grep "Warning:" $1.blg
 5 latex $1 > /dev/null
 6 latex $1 >/dev/null
 7 dvips -q $1.dvi -o $1.ps
 8 ps2pdf $1.ps $1.pdf
 9 rm $1.ps              # save space- zap postscript file
10 cp $1.pdf $HOME/public_html/pdf/$2.pdf
11 chmod a+r
$HOME/public_html/pdf/$2.pdf
```

Fig. 6 `mytex`: generating pdf £les from LaTeX, then copying the result to a web-enable directory so it can be browsed. Assumes that the directory `$HOME/public_html/pdf/` has already been generated. The call to `bibtex` on line 3 generates the bibliography. The multiple passes through LaTeX on lines 5 and 6 £x up all the bibliography and £gure references.

```
%%%% Installation %<
:- load_files([lib0  % pre-load actions
              ,lib1  % predicates
              ,lib2  % start-up commands
              ],[silent(yes),if(changed)]).
%>
%\input{lib0}
%\input{lib1}
%\input{lib2}
```

One nice side-effect of this technique is that the PROD source code can be divided up into simple chunks. The £les `lib0.pl`, `lib1.pl`, and `lib2.pl` only contain PROD body content since the PROD header, preamble and footer is only needed once in the `lib.pl` container £le.

3 Known Bugs

1. Fonts are a problem. If I use the standard Computer Modern font, the preview looks bad on the web. But if I use Times, then sometimes I get funny **£** characters in the font. So, for now, I use Times in order to get decent web previews.

Acknowledgements This research was conducted at West Virginia University under NASA contract NCC2-0979. The work was sponsored by the NASA Of£ce of Safety and Mission Assurance under the Software Assurance Research Program led by the NASA IV&V Facility. Reference herein to any speci£c commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government.

References

1. I. Bratko. *Prolog Programming for Arti£cial Intelligence. (third edition)*. Addison-Wesley, 2001.
2. C. Date. *An Introduction to Database Systems*, volume 6. Addison-Wesley, 1995.
3. P. Deransart, A. Ed-Dbali, and L. Cervoni. *Prolog: The Standard*. Sprunger, 1996.
4. T. Menzies. Domain-speci£c knowledge representations. *AI Expert*, Summer 1989.
5. R. O'Keefe. *The Craft of Prolog*. MIT Press, 1990.
6. D. Poole, A. Mackworth, and R. Goebel. *Computational Intelligence: A Logical Approach*. 1998.
7. L. Sterling and E. Shapiro. *The Art of Prolog (second edition)*. MIT Press, 1994.

A License

This software is distributed under the GNU General Public License.

A.1 nowarranty.txt

PROD comes with ABSOLUTELY NO WARRANTY: for more details type 'warranty'.

This is free software, and you are welcome to redistribute it under certain conditions: for more details, type 'conditions'.

A.2 warranty.txt

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 (see http://www.gnu.org/copyleft/gpl.html or type 'conditions').

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, US.

A.3 conditions.txt

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The 'Program', below, refers to any such program or work, and a 'work based on the Program' means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modi£cations and/or translated into another language. (Hereinafter, translation is included without limitation in the term 'modi£cation'.) Each licensee is addressed as 'you'.

Activities other than copying, distribution and modi£cation are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modi£cations or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modi£ed £les to carry prominent notices stating that you changed the £les and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modi£ed program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

1. prod1.pl : “An example of the PROD Prolog delivery and documentation system.”
Available from `http://tim.menzies.com/pdf/03prod1.pdf`.
2. prod.pl : “A PROlog Documentation, and Delivery Tool”.
Available from `http://tim.menzies.com/pdf/03prod.pdf`.
3. prod0.pl : “TITLE”: a bare-bones minimal example of PROD.
Available from `http://tim.menzies.com/pdf/03prod0.pdf`.
4. prodabout.pl : “Motivations”: the why and who of PROD.
Available from Available from `http://tim.menzies.com/pdf/03prodabout.pdf`.
5. family.pl : “A family database”: documentation of a very simple Prolog family database.
Available from `http://tim.menzies.com/pdf/03family.pdf`.
6. lib.pl : “Commonly used predicates”:
Available from `http://tim.menzies.com/pdf/03lib.pdf`.

Fig. 7 This document is part of the PROD delivery and documentation tool for Prolog applications. To £nd out more about PROD, the best place to start is memo #2.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS