

TEX⁴L⁰G: Simple Prolog Documentation in L^AT_EX

Tim Menzies¹, Claude A. Clause²

¹ Lane Department of Computer Science, University of West Virginia, PO Box 6109, Morgantown, WV, 26506-6109, USA;

<http://menzies.us>; e-mail: tim@menzies.us

² Prolog Programming International

WP ref: tim/src/pl/timlog/tex4log.tex November 10, 2002

Abstract Simple macros written in L^AT_EXallow for a simple documentation scheme for Prolog.

1 Introduction

Gauguin once said that “The ugly may be beautiful, the pretty never.” Here, we strive to turn the ugly facts (and rules) of Prolog into beautiful and attractive typeset prose.

We are not the first to attempt this goal (e.g., see Figure 1). Standard L^AT_EXmacros for pretty-printing Prolog source code (e.g. `lgrind.sty`) focus on just making the source code look pretty. TEX⁴L⁰G is a system for generating descriptions of Prolog source code; i.e. not just source code all the source code *and* explanatory text.

Services offered by TEX⁴L⁰G are:

- A framework for organizing Prolog applications into lots of easily explainable units.
- Via L^AT_EX, the ability to generate postscripts and acrobat files from Prolog source ‘code’.
- Very simple inclusion of Prolog source code files.

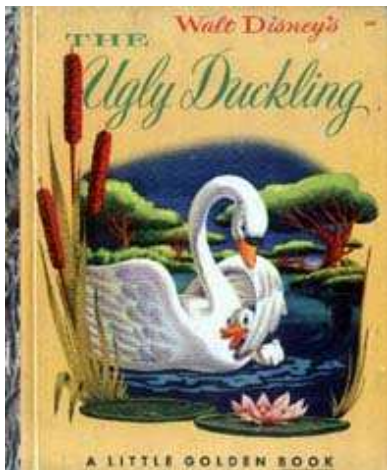


Fig. 1

- A simple protocol for including outputs from Prolog programs into a document.
- Line numbers on source code.
- The ability to add “marks” to source code and refer to these marks symbolically.
- Cross references between source code.
- Auto-inclusion of the GNU Public License into your system.
- Various useful little L^AT_EXtricks.

The above is implemented via the *least* number of extensions to standard L^AT_EX. The dream is that such a minimal implementation will be easy to extend and maintain. We’ll see!

2 Installation

Find an existing TEX⁴L⁰G installation and copy the entire directory tree into a new directory, say `dir`. Create a new sub-directory, say `dir/myapp`. The main file of your new application will be stored in `dir/myapp.pl`, will be documented in `dir/myapp.tex` and the source code will be stored in `dir/myapp/`.

3 Style notes

3.1 Prolog Source Code Idioms

3.1.1 Preparing Include Files Divide your Prolog code up into lots of little files. If it can’t fit into one column, it is too big. In practice, this means no file is more than 55 characters wide and 80 lines long.

Predicates can’t have long argument lists. Practically speaking, this often implies that you need DCGs to handle carrying round updatable state.

3.1.2 Including Source Code Source code should be included into the L^AT_EX source using the `\code{app/file}{Caption}` command. This will generate a L^AT_EX reference `fig:file` which can be accessed in the usual way using `\fig{app/file}`. Included source code gets rendered in a box which includes the name of the file.

3.1.3 Marking Lines in Source Code Within a source code file, a line can be marked with `\MARK{X}` where `X` can be referenced in the \LaTeX source using any of the following macros.

- `\Line{X}` renders as (e.g.) “line 23”.
- `\Where{X}` renders as (e.g.) “line 23 in Figure 5”.

If these references are at start of sentence, they need a leading capital letter. So, two more macros are defined:

- `\LINE{X}` renders as (e.g.) “Line 23”.
- `\WHERE{X}` renders as (e.g.) “Line 23 in Figure 5”.

3.1.4 Referring to Included Files Anytime a Prolog program loads a file, it should be followed by a \LaTeX figure reference. For example:

```
:- [abc]. % see \fig{xyz/abc.pl}
```

This way, if you forget to include `abc.pl`, you’ll get one of those “?” warnings that \LaTeX generates (so you’ll know if you’ve missed anything).

3.1.5 Including Output From Demonstration Code A Prolog idiom in \TeX4Log is that some source code comes in sets of threes:

1. `X.pl` source code. `Xeg.pl` demonstration code that exercises `X.pl`.
2. `Xeg.spy` contains the output generated when `Xeg.pl` was executed.

The macro `\peektell{X}` will include `Xeg.pl` and `Xeg.spy`, if those files exist.

The macro `\showtell{X}` extends `\peektell{X}`. This macro includes `X.pl` and `Xeg.pl` and `Xeg.spy`, if those files exist.

3.2 \LaTeX Idioms

\TeX4Log includes some useful and succinct \LaTeX macros. This section describes those macros.

The silliest macro is `\TeX4Log` which renders as \TeX4Log .

Also:

- Itemized lists can be succinctly started and stopped using `\bi\item...\ei`.
- Enumerated lists can be succinctly started and stopped using `\be\item...\ee`.
- Definition lists can be succinctly started and stopped using `\bd\item...\ed`.

A simple code listing environment with small font size is available via `\begin{LISTING}...\end{LISTING}` or `\bl...\el`.

In-line verbatim text can be typeset using pairs of exclamation marks; e.g. `!text!` is set as `text`. If you need an exclamation mark in your text, use the `\exclaim` macro which expands to “!”.

Cited numeric references are sorted and ranges are shortened to dashes; e.g. `\cite{men01c,men01b,men01a}` could be rendered as [1-3].

Long URLs can be typeset using `\url{...}`. The `\url` macro renders URLs in verbatim font and long strings may be broken at the “/” character (if that improves their rendering).

If figures are labelled with `\label{fig:abc}`, then they can be referenced using (e.g.) `\fig{abc}`. This is typeset as (e.g.) Figure 11.

If equations are labelled with `\label{eq:def}`, then they can be referenced using (e.g.) `\eq{def}`. This is typeset as (e.g.) Equation 11.

If sections are labelled with `\label{sec:ghi}`, then they can be referenced using (e.g.) `\tion{ghi}`. This is typeset as (e.g.) § 2.

3.3 Prolog Idioms

Your application will comprise one main file that loads everything else. Usually this list is loaded silently:

```
:- load_files(
    [myapp/file1 % see \fig{myapp/file1.pl}
    ,myapp/file2 % see \fig{myapp/file2.pl}
    ],
    [silent(true) % no stray text on screen
    ,if(changed) % load files only once
    ]).
```

The load order can vary by often includes:

operators which tells the Prolog parser how to handle the pseudo-English constructs in your code.

hooks that define your `term_expansion` and `goal_expansion` rules.

hacks that define some patches to our Prolog system

- *Many library files* that set things up;
- *Some core files* that combine the libraries to perform some interesting task;
- *Some example files* that show off the core code.

For every core file `core.pl`, there should be a `coreeg.pl` example file containing a `coreeg` predicate that does the showing off. `coreeg.pl` should have the following structure:

```
:- [lib1 % see \fig{lib1.pl}
    ,lib2 % see \fig{lib2.pl}
    ].
...
ccode :- doSomething.
:- demos(coreeg). % call to the core code that traps output
```

A file with this structure will generate a file `coreeg.out` as a side effect of loading `coreeg.pl`. In order to include the core, example, and output files, the macro `\showtell{core}` is useful. The call `\showtell{core}` generates three figures: one for `coreeg.pl`, one for `coreeg.out` and one for `core.pl`.

It is good practice to reference the core, example, and output files in the main file; e.g.

```
:- [abc]. % see \fig{abc.pl}, \fig{abceg.pl} & \fig{abceg.out}
```

Never use a backslash such as `\X` so:

- Use `not(X)` instead of `\+ X`
- Use `nl` instead of `\n`.

Never use `{X}` since \LaTeX can’t print that. So:

- Define `goal_expansion/2` to repair over-zealous DCG expansions (see Figure ??).
 - Use `code(X)` to denote code that should be not be DCG-expanded. Then add the following item to `hooks.pl`
- ```
goal_expansion(call(A,B,B),A).
```

## 3.4 Other Requirements

**3.4.1 Peekers** A *peeker* is a set of predicate sthat “looks before it leaps”. The idea is that before doing anything on `X`, the code peeks at `X` and then uses what it sees to determine what actions to perform. Peekers, as used in this code, are filters that convert `X` to `Y`. Their general form is three predicates: `p`, `p0`, `p1`:

## license/license.pl

```
hello :-
 '[myName,myVersion,myYear,myMantra,myAuthor]=
 [Name, Ver, Year, Mantra, Creator],
 format('~n~w version ~w',[Name,Ver]),nl,
 format('Copyright (C) ~w by ~w',[Year,Creator]),
 nl,format('~w',[Mantra]),
 nl,nl,format('~w',[Name]),
 chars('license/nowarranty.txt'). % see §A.1

warranty :-
 '[myName,myYear,myAuthor]=[Name, Year,Creator],
 format('~w by ~w',[Name,Creator]),nl,
 format('Copyright (C) ~w',[Year]),nl,
 chars('license/warranty.txt'),nl. % see §A.2

conditions :-
 chars('license/conditions.txt'),nl. % see §A.3

wave :-
 '[myName,myVersion]=[Name,Ver],
 format('~w version ~w',[Name,Ver]),nl.
```

Fig. 2

```
p(X,Y) :- once(p0(X,Z)), p1(Z,Y).

p0(X, barph(X)) :- var(X). % usual top peeker
p0(X, barph(number(X))) :- number(X). % e.g.
p0(X,Y), (X,Y) ...
p0(X, standard(X)). % some final default action

p1(barph(X), _) :- print(bad(X)), fail.
p1(standard(X),Y) :- Y is X % e.g.
p1((X0,Y0), (X,Y)) :- p(X0,X), p(Y0,Y) ...
```

The `p0` predicate classifies the incoming structure into one of  $N$  terms. One `p1` predicate is written for each such term. If `p1` recurses, then the recursive call is back to `p`.

Peekers have several useful features:

- Fewer/no cuts in the code.
- Stepping through the `spy` outputs over for a peeker is usually easier since only one choice for `p1` is ever shown and the `p0` processing is often simple enough that the programmer can just skip it.
- Often there are less `p1` predicates than `p0` predicates since `p0` might map different incoming terms to the same functor (e.g., in the above, while there are two situations where we want to `barph`, we only needed to write `barph` once.
- If `p0` is constrained to key just on features available when the source code is loaded for the first time, then the peeker can be optimized as follows. At load time, pre-compute and cache all the `p0` results on the loaded code. Then, at runtime, replace all the `p0` with `p0(X,X)` and all that lookup computation disappears.

## A License

This software is distributed under the GNU General Public License. Figure 2 shows routines to display that license.

### A.1 nowarranty.txt

NB 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 modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term 'modification'.) Each licensee is addressed as 'you'.

Activities other than copying, distribution and modification 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 modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files 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 modified 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.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and 'any later version', you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM 'AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

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