Using SWI-Prolog

Contents

1 Running a Prolog program

Running a Prolog program involves

1. creating a file containing the program
2. saving the file
3. loading the file into Prolog (called *compiling*, *consulting*, or *reconsulting* the file in Prolog jargon)
4. finally, calling some goal defined in the program

The version of Prolog we will be using is one which runs on a variety of different platforms and is called SWI-Prolog. At York, it is accessible both on the machines in the elab in the Language Centre and on the university's Windows network. (Information about how to get your own copy is provided at the end of this documentation.)

1.1 Getting started

1.1.1 In the elab

1. Log in to your account with your username and password. If you have not previously used one of these machines, the password will be 'landls'; you will then be prompted for a new password.
2. Double-clicking the icon for `plwin.exe` on your desktop will launch the Prolog program.

You should be rewarded by the appearance of a new window containing some welcome messages and the Prolog prompt `|?-` , which indicates that Prolog is ready for input.

1.1.2 On a networked Windows machine

1. Log in to your account with your username and password.
2. from the Start menu follow the path `Accessories|Windows Explorer`. Click with the mouse.
3. In the window that opens use the panel on the left to navigate your way down the path (by clicking on the + icon) `My Computer | Teaching on cserv ... | Applications | Departments | Language | Prolog 5 | files| bin` and double-click on the last item to open it.
4. Find the file named `plwin.exe` and right-click on it
5. In the box that opens, select `Create shortcut` and agree with the suggestion to put the shortcut on your desktop
6. Double-clicking the icon for `plwin.exe` on your desktop will launch the program.

You should be rewarded by the appearance of a new window containing some welcome messages and the Prolog prompt `|?-` , which indicates that Prolog is ready for input.

1.2 Creating a Prolog program.

To create a Prolog program, you must use an editor to create a file. There are a number of editors available and you can use whatever you prefer. If you have no preference, I suggest using the emacs editor that comes with SWI-prolog. It is Prolog-aware and will format and colour-code Prolog programs for you, as well as providing a number of other Prolog-related conveniences. To use emacs on a networked Windows machine, use the following instructions. (In the elab, this has already been set up.)

1. Start Prolog (by double-clicking the shortcut to `plwin.exe` on your desktop).
2. From the *Menu* bar select `Settings|User init file`
3. Scroll down the file until you see `%:- set_prolog_flag(editor, pce_emacs).`
4. Delete the percentage sign at the beginning of the line.
5. Save the file (Either *Save* from the *File* menu, or use the key combination *Ctrl-x Ctrl-s* — that is, hold down the Control key and type *xs*).
6. Quit Prolog. Next time you start Prolog, the emacs editor will be available.

Here is how you create a Prolog program.

1. Double-click the Prolog program icon.

2. Use the File menu of the resulting window to create a new file and give it a name, with the extension `.pl`, making sure you save it in a location to which you are allowed to write files.

   - If you are using a networked Windows machine, this will mean somewhere on your your H: drive.

3. If you are in the elab, you have two choices

   (a) Save it to the elab server: `<username> on 'jigsawxp1' (U:)`. (Where `<username>` is your username.) The drawback to this is that your files are then only accessible from the elab computers.

   (b) Save it on your networked filestore; it will then be accessible both from the elab and elsewhere on campu. To do this follow these instructions

   - From the Start menu, Choose `My Network Places` and then select `Add Network Place` from the pane on the left
   - This opens the `Add Network Place Wizard`; respond by clicking `Next` until you get the invitation to enter an Internet or network address.
   - Type `\\userfs.york.ac.uk\<username>` (where `<username>` is your username).
   - Your username should be entered in the form `<username>@york.ac.uk`
   - You will be able to access your network filestore via `My Network Places`.

4. Enter the text of your file in the file window.

5. You save your file by using the Save option from the File menu on the menu bar at the top of the editor's window.

6. Make sure that you give your Prolog files the extension `.pl` .

7. You can copy text in the usual manner by using Copy and Paste from the Edit menu, or by selecting the text with the cursor (this copies automatically) and using using the key combination Ctrl-y to paste.

8. The same procedures work in the SWI-Prolog window; select the text to copy and use Ctrl-y to paste.

1.3 The Web

All these course notes are available over the web by using a browser such as Internet Explorer. The URL is:

http://www-users.york.ac.uk/ sjh1/

1.4 An Example

Use the emacs editor to create a file called `family.pl`.

When you have created a new file window, type the following, being particularly careful to type exactly what appears here, including full stops and commas:

```
male(harry).
female(liz).

parent(phil, chas).
parent(liz, chas).
parent(chas,harry).
parent(chas,wills).


grandmother(GM, C):-
    mother(GM, P),
```

```
    parent(P, C).

mother(M,C):-
    female(M),
    parent(M, C).
```

1.4.1 Saving a file

Before you can do anything with this, you must save your file. Use Save from the File menu (or *Ctrl-x Ctrl-s*).

1.4.2 Consulting a file

Now that you have created a file, you need to inform Prolog of its contents. You do this by *consulting* the file. There are two ways to do this.

First, use the mouse to select the SWI-Prolog window in which Prolog is running. Then, type either

```
|?- consult(family).
```

or

```
|?- [family].
```

Note: It is conventional to give Prolog source files the extension `.pl`. SWI-Prolog knows about this, so you do not need to give the full filename and extension when consulting it. In fact, if you do type `consult(test.pl).` or `[family.pl]`, you will receive an error message for your pains!

This is the standard procedure in all Prolog systems.

Alternatively, put the cursor in your emacs file window and use the key combination *Ctrl-c Ctrl-b* (For 'consult buffer' — windows are called 'buffers' in emacs-speak).

If your file contains no errors, you should receive a message in the SWI-prolog window telling you that the file has been consulted successfully. You are now in a position to get Prolog to do some work.

1.4.3 Posing a goal

Once you have consulted your file, you are in a position to get Prolog to do some work.To get Prolog to do some computation, you need to issue a goal. Click on the Prolog window and type the following (`|?` is the Prolog prompt):

```
|?- grandmother(liz, Who).
```

Type this exactly, including the capital W and the final full-stop, and, hit `<return>`. Prolog will respond:[1]

```
 Who = harry
```

Next, type a semi-colon followed by `<return>`. Prolog will respond

```
Who = wills
```

[1] If nothing happens, 1) check your typing, 2) make sure you haven't forgotten the full-stop.

Again, type a semi-colon followed by `<return>`. Prolog will respond:

```
no
```

This means that you have now received all the responses to the goal that Prolog can compute on the basis of the information supplied in your program file.

1.4.4 Tracing

You will frequently want to see what Prolog does while it is evaluating your programs. This is often necessary when you are trying to track down bugs in your program. Unfortunately, Prolog's debugging environment is not very good. To trace the execution of a goal (such as `grandmother(liz, Who).`), type

```
|?-trace.
```

and then a goal, such as

```
|?- grandmother(liz, Who).
```

Prolog will then output to the screen the sequence of steps it carries out in executing this goal

```
  ?- grandmother(liz, Who).
   Call: (6) grandmother(liz, _G396) ? creep
   Call: (7) mother(liz, _G450) ? creep
   Call: (8) female(liz) ? creep
   Exit: (8) female(liz) ? creep
   Call: (8) parent(liz, _G450) ? creep
   Exit: (8) parent(liz, chas) ? creep
   Exit: (7) mother(liz, chas) ? creep
   Call: (7) parent(chas, _G396) ? creep
   Exit: (7) parent(chas, harry) ? creep
   Exit: (6) grandmother(liz, harry) ? creep


Who = harry


Yes

[debug]  ?-
```

At each point where it calls a goal, Prolog pauses and requires a carriage return from you before proceeding.

To turn off tracing, type

```
| ?- nodebug. <return>
```

(Note the full stop!)

1.5 A note on filenames and directories in Prolog

You may be in the (good) habit of organising your work by putting material related to a particular topic into a (sub-)directory. If you decide to do this with your Prolog files, you need to be aware of difference in the way you specify the filename using consult.

If the file is in the root directory of your M drive, and the file has the form `<filename>.pl`, you simply type

```
| ?- consult(<filename>). <return>
```

If, however, you have created a directory called, say `myprolog` and your file `<filename>.pl` lies inside that directory, the above instruction will not work and you will receive an error message.

What you have to do is:

1. to supply the full filename, including the .pl extension
2. the path from your `$HOME` directory to the file
3. enclose the whole expression in single quotes, like this:

   ```
   | ?- consult('myprolog/<filename>.pl'). <return>
   ```

Alternatively, start Prolog and select `Consult` or `Edit` from the `File` menu and navigate to the relevant folder.

2 Quitting Prolog

When you want to finish your Prolog session, just click the close box in the top right-hand corner of the SWI-Prolog window, as normal.

Do not forget to log off before leaving your machine.

3 Further information

3.1 Free Prologs

If you have a computer of your own, there are numerous free versions of Prolog available on the web.

SWI Prolog is available for download over the Web. More information is available at :
http://www.swi-prolog.org/

For Macintoshes running System X, a version of SWI-prolog is available from the same address.

3.2 Prolog Manuals

The manual for SWI-prolog can be accessed from the SWI-prolog *Help* menu.

3.3 Prolog courses on the web

There is a Prolog course (at Birmingham University) available over the web at
http://www.cs.bham.ac.uk/~pjh/prolog_course/se207.html
This site also has information about other public domain Prologs.

And another by Patrick Blackburn, Johan Bos and Kristina Striegnitzat called 'Learn Prolog Now!' at
http://www.coli.uni-saarland.de/ kris/learn-prolog-now/

Amzi!'s site provides several Prolog learning aids:
http://www.amzi.com/products/learning_products.htm

There is also a guide to Prolog programming at

http://kti.ms.mff.cuni.cz/~bartak/prolog/index.html

Finally, the WWW Virtual Logic Programming Library contains links to all kinds of logic programming resources:
http://archive.comlab.ox.ac.uk/logic-prog.html