G52PSA
Planning, Search & AI Programming

Lab 1: Getting started

Brian Logan
School of Computer Science
bsl@cs.nott.ac.uk

G52PSA labs

- lab sessions are in B52 (CS) from 10:00-11:00 on Thursdays
- one hour should be more than enough to complete the exercises
- I'll be in the lab to answer questions

First lab: getting started

The aims of this lab are:

- to check that you can run SWI-Prolog
- to be able to load a simple Prolog program and execute some queries
- to make some simple modifications to the program, e.g., new facts and rules

Running Prolog

- (on Windows) the simplest way to run Prolog is to double click on a file with a .pl extension

- this should produce:

```
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.2.6)
Copyright (c) 1990-2012 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

?-
```

- the `?-` is the Prolog *top-level prompt*, at which you can type commands (directives) and queries

SWI-Prolog reference manual

- to read the manual, type “help.” (you don’t have to type the prompt characters but you do need the “.”)

```
?- help.
```

- the manual is also available at

http://www.swi-prolog.org/pldoc/refman/

- for the first lab, you should read at least sections 2.1, 2.6, 3.3 and 3.4

Exercise 1: getting started

- download the Prolog program for this lab, `family.pl`, from Moodle
- run Prolog and load (consult) the family relationships program, either by double clicking on `family.pl` or by typing

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

- at the Prolog prompt – Prolog should respond with (something like):

```
% family compiled 0.00 sec, 17 clauses
true.

?-
```

Exercise 2: simple queries

- try some simple queries, e.g.,

```
?- male(tom).

?- father(tom,bob).

?- father(bob,X).

?- grandparent(pam,X).

?- parent(tom,X), female(X).
```

- try checking for alternative solutions to the query with "`;`"
- where there are multiple solutions, try to understand the order in which they are produced

Exercise 3: changing the program

- modify the family relationships program to add

  a) new members of the family, e.g., `parent(alice,pam)`, `female(alice)`

  b) a new family relationship defined in terms of existing relationships, e.g., `grandfather(X,Y)`, or `sister(X,Y)`

- try some simple queries to make sure your modified program works as you expect

Exercise 3: editing files

- to do this, open the file `family.pl` in the Prolog editor by typing:

```
?- edit(family).
```

- make the changes, save the file and reload it into Prolog by typing:

```
?- make.
```

- Prolog should respond with (something like):

```
% family compiled 0.00 sec, 18 clauses
true.

?-
```