East West University

Department of Computer Science and Engineering

1. Introduction:

Logic programming is a programming paradigm based on mathematical logic. In this paradigm the programmer specifies relationships among data values (this constitutes a logic program) and then poses queries to the execution environment (usually an interactive interpreter) in order to see whether certain relationships hold. Putting this in another way, a logic program, through explicit facts and rules, defines a base of knowledge from which implicit knowledge can be extracted. This style of programming is popular for data base interfaces, expert systems, and mathematical theorem provers. In this tutorial you will be introduced to Prolog, the primary logic programming language, through the interactive SWI-Prolog system (interpreter).

2. Expressing fact:

Prolog permits you to describe facts as symbolic relationships. For example, if the right speaker in your stereo system is not emitting sound (is dead), you can express this in English as

The right speaker is dead.

This same fact can be expressed in Prolog as

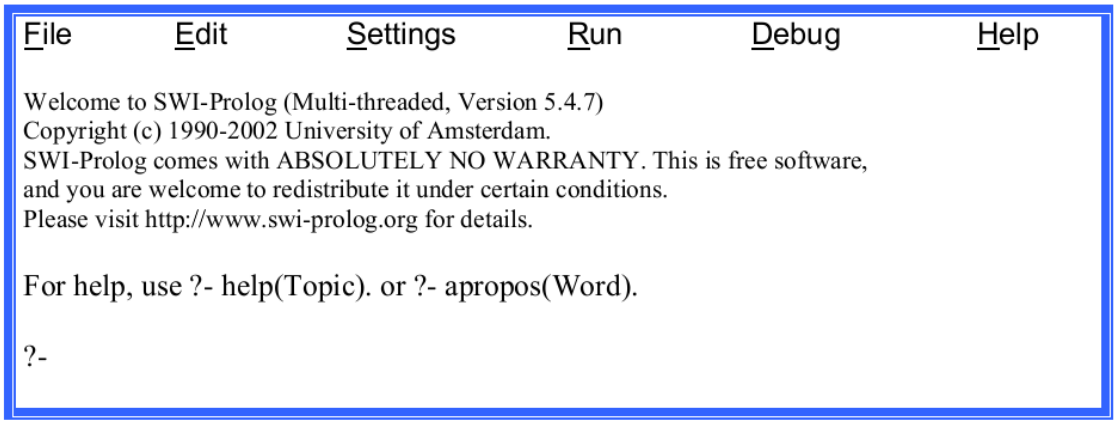
is(right_speaker,dead).

This factual expression in prolog is called a clause. Notice the period at the end of the clause.

In this example *right_speaker* and *dead* are *objects*. An object is the name of an element of certain type. It represents an entity or a property of an entity in the real world.

Fact	Relation
is(right_speaker,dead).	**Is**

3. **Using Prolog:**



3.1. Menu commands.

The SWI-Prolog console has a menu for accessing the most commonly used commands (see the first line in Figure 1). In particular, if you click on the button File of the menu you will see the following entries:
Consult, Edit, New, Reload modified files, Navigator, Exit.

We will consider the first three commands in the next section. The command Reload modified files is to reload all loaded source-files that have been modified using the make command (section 2.2). The command Navigator opens an explorer-like view on Prolog files and the predicates they contain: to view the predicates in a file you should find the file by Navigator and then click on the file name. Also Navigator provides an option to edit the selected file.
The menu button Settings allows you to change the font of the console as well as providing some additional options. The button Run provides two options: Interrupt and New thread. Interrupt is to interrupt the running Prolog process. New thread creates a new window running in a separate thread of execution. The button Help provides access to the online Prolog manual. In particular, it contains links to pages of the SWI-Prolog website.

2.2. Some useful commands.
This section contains a brief overview of important Prolog commands.
Consult(+File)
Load a source-file from the current folder. The file-extension pl can be omitted (see Figure 2)

```
?- consult(family).
% family compiled 0.00 sec, 8,016 bytes

Yes
?- consult(likes).
ERROR: source_sink `likes' does not exist
?- ['C:/Program Files/pl/demo/likes'].
% C:/Program Files/pl/demo/likes compiled 0.00 sec, 2,300 bytes

Yes
```

Figure 2

The first line in Figure 2 gives an example of a successful execution of the command consult: family.pl has been loaded from the current folder. The file likes.pl is not in the current folder, therefore to load it we should use absolute path (see the sixth line in Figure 2).

pwd

Print working directory in the console.

edit (+File)

Edit file with the given name. If Prolog is started by opening a .pl file then the file name in the command edit can be omitted.

Make:

Reload all files that have been changed since they were last loaded. This command is normally used after editing one or more files.

help(+Spec) :

Gives help on Spec. For example, help (make) gives help on the command make, and help (2) (respectively, help (2-1)) gives access to Chapter 2 of the manual (respectively to Section 1 of Chapter 2). Alternatively, you can use access to the manual via the menu.

apropos (+Keyword) :

Search for all predicates that contain Keyword in their name or short description in the manual.

Control-C:

Try to interrupt the running Prolog process (see also Runin the menu). After execution, you will see the following line in the console:

Action (h for help)?

If you really want to exit Prolog, type e. Else you can choose one of the following options.

a: abort	b: break
c: continue	e: exit
g: goals	t: trace
h (?): help	

4. Example:

Figure 3 shows the family tree of Smith's family.

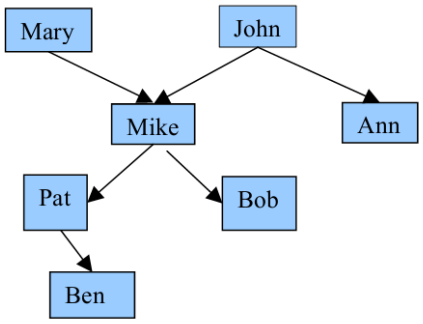


Figure 3

To write a program which defines a family relation click the button File of the menu and choose the option New. Let us call the new file smith.pl. Type the text in the file and save it (see Figure 4).

```
parent(john, ann).      /*the fact that John is a parent of Ann*/
parent(john, mike).     %the fact that John is a parent of Mike
parent(mary, mike).
parent(mike, pat).
parent(mike, bob).
parent(pat, ben).
```

Figure 4

Here parentis the name of a family relation. Note that the text in /* */ is a comment as well as the text between the percent character % and the end of the line (see the first and, respectively, the second lines in Figure 4). Consult the file (see section 2.2); for example, use the command

?- consult(smith).

Now we can pose Prolog questions about the relation parent (see Figure 5).

```
?- parent(john, mike).          /* Is John a parent of Mike? */

Yes
?- parent(mike, ann).           /* Is Mike a parent of Ann? */

No
?- parent(mike, X).            / *Who is a child of Mike? */

X = pat

Yes
?- parent(mike, _).          /* Is Mike a parent? */

Yes
```

Figure 5

If we want to get a list of all pairs parent-child, type parent(X, Y), and then press repeatedly semicolon (;) (see Figure 6). To abort the process, press return.

```
?- parent(X, Y).
X = john
Y = ann ;

X = john
Y = mike ;

X = mary
Y = mike ;

X = mike
Y = pat ;

X = mike
Y = bob ;

X = pat
Y = ben ;
No                                  /*indicates that the list is full*/
```

Figure 6

To extend the program by adding new facts or rules, edit the file smith.pl. You can use, for example, option File/Edit of the menu, which will open the file smith.pl for editing. For example, let us add information on the gender and define predicates mother, father (see Figure 7).

```
parent(john, ann).                          %the fact that John is a parent of Ann
parent(john, mike).                         %the fact that John is a parent of Mike
parent(mary, mike).
parent(mike, pat).
parent(mike, bob).
parent(pat, ben).

female(mary).                               %the fact that Mary is a female
female(ann).
female(pat).
male(john).                                 %the fact that John is a male
male(mike).
male(bob).
male(ben).

mother(X,Y) :- parent(X, Y), female(X).     %definition of the predicate mother
father(X,Y) :- parent(X, Y), male(X).       %definition of the predicate father
```

Figure 8

Consult the file smith.pl. Now you can ask new questions. For example, try the following:

?- female(X).

?- mother(X, mike).

?- mother(pat, Y).

?- father(X, Y).

Problem 1

Write a Prolog program which describes a directed graph with the following structure.