Introduction to Artificial Intelligence and Problem Formulations

Dr. Tapas Kumar Mishra

Assistant Professor
Department of Computer Science and Engineering
SRM University
Amaravati-522 240, Andhra Pradesh, India
Email: tapaskumar.m [at] srmap [dot] edu [dot] in

SWI-Prolog for MS-Windows

SWI-Prolog for MS-Windows

- Download: `http://www.swi-prolog.org/download/stable`
- The installation folder (by default C:\Program Files\swipl) contains a subfolder **demo** with the file **likes.pl**.
- Be sure to get the **quotes right** and terminate the command with a **full stop (.)**.
  - ?- [swi('demo/likes')].
  - **true.**

Executing a query

Executing a query

```
?- likes(sam, X).
X = dahl ;
X = tandoori ;
X = kurma ;
X = chow_mein ;
X = chop_suey ;
X = sweet_and_sour ;
X = pizza ;
X = spaghetti ;
X = chips.
?-
```

Some Useful Commands

consult

```
?- consult(likes).
true.
```

pwd

Print working directory (folder).

```
?- pwd.
% c:\program files\swipl\demo\
true.
```

ls

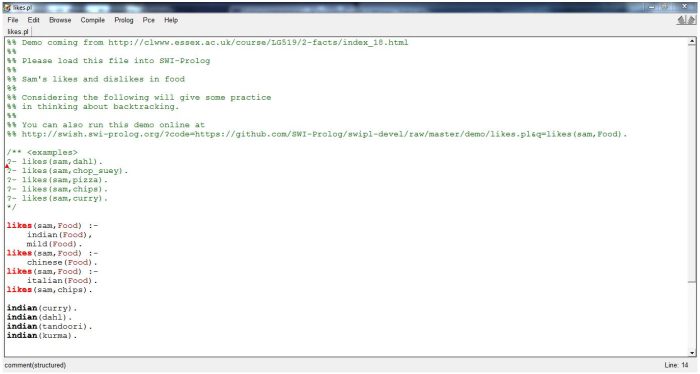
List files in current directory.

```
?- ls.
% likes.pl README.TXT
true.
```

Some Useful Commands

edit

If Prolog is started by opening a .pl file in the explorer, edit this file. Also available from the menu.
?- edit.
true.



```
%% Demo coming from http://clwww.essex.ac.uk/course/LG519/2-facts/index_18.html
%%
%% Please load this file into SWI-Prolog
%%
%% Sam's likes and dislikes in food
%%
%% Considering the following will give some practice
%% in thinking about backtracking.
%%
%% You can also run this demo online at
%% http://swish.swi-prolog.org/?code=https://github.com/SWI-Prolog/swipl-devel/raw/master/demo/likes.pl&q=likes(sam,Food).

/** <examples>
?- likes(sam,dahl).
?- likes(sam,chop_suey).
?- likes(sam,pizza).
?- likes(sam,chips).
?- likes(sam,curry).
*/

likes(sam,Food) :-
    indian(Food),
    mild(Food).
likes(sam,Food) :-
    chinese(Food).
likes(sam,Food) :-
    italian(Food).
likes(sam,chips).

indian(curry).
indian(dahl).
indian(tandoori).
indian(kurma).
```

Some Useful Commands

make

Reload all files that have been changed since they were last loaded. Normally used after editing one or more files.

```
?- make.
true.
```

Data types and Variables

Data types

Prolog's single data type is the **term**. Terms are either **atoms**, **numbers**, **variables** or **compound terms**.
Atom: general-purpose name with no inherent meaning
Number : Numbers can be floats or integers. (Length is Arbitrary for different compilers)
Variables : denoted by a string consisting of letters, numbers and underscore characters, and **beginning with an upper-case letter or underscore**
Compound term : comma-separated list of argument terms, which is contained in parentheses.

Operators

OR operator

We have to write **FACTS** / **RULES** multiple times.

Example:

```
Asian(X):-
      Indian(X).
Asian(X):-
      Pakistani(X).
Asian(X):-
      nepali(X).
```

Operators

AND operator

We have to write **FACTS** in a RULE separated by comma (,)

Example:

```
Pass(X):-
    Physics(X),
    Chemistry(X),
    Math(X).
Math(ravi).
Physics(ravi).
Chemistry(ravi).
math(ganesh).
```

Operators

Arithmatic operator

Arithmetic examples	**Prolog Notation**
$6 + 2 = 8$	8 is 6+2.
$6 * 2 = 12$	12 is 6*2.
6 - 2 = 4	4 is 6-2.
6 - 8 = - 2	-2 is 6-8.
$6 \div 2 = 3$	3 is 6/2.
$7 \div 2 = 3$	3 is 7/2.
1 is the remainder when 7 is divided by 2	1 is mod(7,2).

Program 1

This program is designed to answer questions about relationships within a given family tree. The program will tell you who the mother, father, sister, brother, aunt, uncle, grandmother, grandfather, brother in law, sister in law, mother in law, father in law, ancestor, and descendent of someone is. The standard form is predicate(someone, relation) where the relation will be the mother or father or so on.

```
male(dasarath).
male(ram).
male(laxman).
male(bharath).
male(satrughna).
male(luv).
male(kush).
female(kaikeyi).
female(kaushalya).
female(sumitra).
female(sita).
female(urmila).
```

Program 1

```
/* parent ( child, parent). */
parent(ram, dasarath).
parent(laxman, dasarath).
parent(bharath, dasarath).
parent(satrughna, dasarath).
parent(luv, ram).
parent(kush, ram).
parent(luv, sita).
parent(kush, sita).
parent(ram, kaushalya).
parent(laxman, sumitra).
parent(bharath, kaikeyi).
parent(satrughna, sumitra).
```

Program 1

Q1: find the grand parent of luv (using recursion)

```
/* parent ( child, parent). */
grandparent(X,Y):-
    parent(X,Z),
    parent(Z,Y).
grand_father(X,Y):-
    parent(X,Z),
    parent(Z,Y),
    male(Y).
```

References

1. https://www.swi-prolog.org/.

Thank You!