Introduction to Prolog

5th October 2018

Rohan Sonawane

Asis Kumar Panda

What is Prolog ?

- Prolog stands for “programming in logic”
- Declarative programming language
- Useful in Artificial Intelligence, NLP, Relational Databases
- Approximates first-order logic

History

- Kowalski: late 60’s Logician who showed logical proof can support computation.

- Colmerauer: early 70’s Developed early version of Prolog for natural language processing, mainly multiple parses.

- Warren: mid 70’s First version of Prolog that was efficient.

Installation

1. Add the ppa **ppa:swi-prolog/stable** to your system's software sources:
   - Open a terminal (Ctrl+Alt+T) and type:
     ```
     sudo add-apt-repository ppa:swi-prolog/stable
     ```
   - Afterwards, update the package information:
     ```
     sudo apt-get update
     ```

2. Install SWI-Prolog through the package manager:
   - Open a terminal (Ctrl+Alt+T) and type:
     ```
     sudo apt-get install swi-prolog
     ```

LOAD FILE IN PROLOG?

- Store all your facts and rules in one file and save this file with extension “.pl” Example : kb1.pl
- Open terminal and go to the path where your file is stored
- Type swipl to start prolog
- To load the knowledge base, type name of the file without extension in square brackets

Example : [kb1].

CONSTRUCTS IN PROLOG

- There are three basic constructs in Prolog :

1) Facts
2) Rules
3) Queries

- Collection of Facts and Rules is Knowledge Base.

- How to use knowledge Base?

Answer : By asking Queries

FACTS

- *John likes Mary*
  - `like(john,mary).`
- Names of relationship and objects must begin with a lower-case letter.
- Relationship is written *first* (typically the *predicate* of the sentence).
- *Objects* are written separated by commas and are enclosed by a pair of round brackets.
- The full stop character ‘.’ must come at the end of a fact.

RULES

Statements about objects and their relationships

head :- body

- Express
  - If-then conditions
    - I use an umbrella if there is a rain
    - `use(i, umbrella) :- occur(rain).`
  - Generalizations
    - All men are mortal
    - `mortal(X) :- man(X).`
  - Definitions
    - An animal is a bird if it has feathers
    - `bird(X) :- animal(X), has_feather(X).`

EXAMPLE

KNOWLEDGE BASE 1

```
woman(mia).
woman(jody).
woman(yolanda).
playsAirGuitar(jody).
party.
```

Number of facts : 5
Number of Rules : 0

KNOWLEDGE BASE 2

```
happy(yolanda).
listens2Music(mia).
listens2Music(yolanda):-
happy(yolanda).
playsAirGuitar(mia):-
listens2Music(mia).
playsAirGuitar(yolanda):-
listens2Music(yolanda).
```

Number of facts : 2
Number of Rules : 3

HOW TO USE KB1 ?

KNOWLEDGE BASE 1
woman(mia). *woman(jody).* *woman(yolanda).* *playsAirGuitar(jody).* *party.*

```
?- woman(X).
X = mia;
X = jody;
X = Yolanda.

?- playsAirGuitar(Y).
Y = jody.
```

HOW TO USE KB2 ?

Knowledge Base 2

```
happy(yolanda).
listens2Music(mia).
listens2Music(yolanda):-
happy(yolanda).
playsAirGuitar(mia):-
listens2Music(mia).
playsAirGuitar(yolanda):-
listens2Music(yolanda).
```

Queries

```
?- playsAirGuitar(yolanda).
true.
```

What if first fact is removed from knowledge base?

CLAUSES

- The facts and rules contained in a knowledge base are called clauses.
- Another way of looking at KB2 is to say that it consists of three predicates (or procedures).
- The three predicates are:

```
listens2Music

happy

playsAirGuitar
```

Family (KNOWLEDGE BASE 3)

```
father(a,b).
father(e,d).
mother(c,b).
mother(d,f).
parent(X,Y) :- father(X,Y).
parent(X,Y) :- mother(X,Y).
grandfather(X,Y):- father(X,Z),parent(Z,Y).
```

CONJUNCTION

```
happy(vincent).
listens2Music(butch).
playsAirGuitar(vincent):-
listens2Music(vincent) , happy(vincent).
playsAirGuitar(butch):- happy(butch).
playsAirGuitar(butch):- listens2Music(butch).
```

Note : “,” means AND

DISJUNCTION

```
happy(vincent).

listens2Music(butch).

playsAirGuitar(vincent):- listens2Music(vincent),
happy(vincent).

playsAirGuitar(butch):- happy(butch).

playsAirGuitar(butch):- listens2Music(butch).
```

We can replace last two clauses by

```
playsAirGuitar(butch):- happy(butch) ; listens2Music(butch).
```

KNOWLEDGE BASE 4

```
loves(vincent,mia).
loves(marsellus,mia).
loves(pumpkin,honey_bunny).
loves(honey_bunny,pumpkin).
jealous(X,Y):- loves(X,Z),
loves(Y,Z).
```

```
?- jealous(marsellus,W).
    W = vincent.
```

- We can use variables in clauses if required.

Prolog Syntax

TERMS

- What are facts, rules, and queries built out of?
- Answer: terms

There are four kind of terms

1) Atoms
2) Numbers
3) Variables
4) Complex terms or structures

ATOMS

- A string of characters made up of upper-case letters, lower-case letters, digits, and the underscore character, that ***begins with a lower-case letter.*** Here are some
examples: butch , big_kahuna_burger ,listens2Music and playsAirGuitar .

- An arbitrary sequence of characters enclosed in single quotes. For example ’ Vincent ’, ’ The Gimp ’, ’ Five_Dollar_Shake ’, ’ &^%&#@$ &* ’, and ’ ’. The sequence of characters between the single quotes is called the atom name.

- A string of special characters. Here are some
examples: @= and ====> and ; and :- are all atoms. As we have seen, some of these atoms, such as ; and :- have a pre-defined meaning.

NUMBERS

```
?- A is 36/5.
A = 7.2.

?- A = 36/5.
A = 36/5.

?- (31 is (36 - 5)).
true.

?- (31 = (36 - 5)).
false.
```

ARITHMETIC OPERATIONS IN PROLOG

```
?- 12 is 6*2.

true.
```

```
?- X is mod(15,4).

X = 1
```

```
add_3_and_double(X,Y) :- Y is (X+3)*2.

?- add_3_and_double(1,X).

X = 8
```

VARIABLES

- A variable is a string of upper-case letters, lower-case letters, digits and underscore characters that ***starts either with an upper-case letter or with an underscore***.
  For
  example, X , Y , Variable , _tag , X_526 , List , List24, _head , Tail , _input and Output are all Prolog variables.

- The variable _ (that is, a single underscore character) is rather special. It's called the anonymous variable

Anonymous Variable

- The name of every anonymous variable is _ .
- The anonymous variable is an element of Prolog that is similar to a variable in almost every way.
- As presumably, all the occurrences of the name of an ordinary variable stand for the same variable within one clause.
- On the contrary**,** *every occurrence of _denotes a distinct variable* .

Example:

```
loves(mia, jack).
loves(jack, mia).

?- loves(X, X).
false.
?- loves(_, _).
true;
true.
```

Library DataBase Example

```
book(CatalogNo, Title, author(Family, Given)).
libmember(MemberNo, name(Family, Given), Address).
loan(CatalogNo, MemberNo, BorrowDate, DueDate).
date(Year, Month, Day).
borrowed(MemFamily, Title, CatalogNo) :-
   libmember(MemberNo, name(MemFamily, _), _),
   loan(CatalogNo, MemberNo, _, _),
   book(CatalogNo, Title, _).
later(date(Y, M, Day1), date(Y, M, Day2)) :- Day1 > Day2.
later(date(Y, Month1, _), date(Y, Month2, _)) :-
   Month1 > Month2.
later(date(Year1, _, _), date(Year2, _, _)) :- Year1 > Year2.
```

Complex terms or Structures

- Complex terms are build out of a functor followed by a sequence of arguments.

  Example : `like(a, b).`

- The arguments are put in ordinary parentheses, separated by commas, and placed after the functor.

- Note that the functor has to be directly followed by the parenthesis; you can't have a space between the functor and the parenthesis enclosing the arguments. ***The functor must be an atom***. That is, variables cannot be used as functors. On the other hand, arguments can be any kind of term.

- The number of arguments that a complex term has is called its arity.

- Nesting of Complex terms inside complex terms

```
hide(X,father(father(father(butch)))).
```

QUESTIONS

Which of the following sequences of characters are atoms, which are variables, and which are neither?

1. vINCENT
2. Footmassage
3. variable23
4. Variable2000
5. big_kahuna_burger
6. ’big kahuna burger’
7. big kahuna burger
8. ’Jules’
9. ’_Jules’

QUESTIONS Continued ..

- Which of the following sequences of characters are atoms, which are variables, which are complex terms, and which are not terms at all? Give the functor and arity of each complex term.

1. loves(Vincent,mia).
2. ’loves(Vincent,mia)’.
3. Butch(boxer).
4. boxer(Butch).
5. and(big(burger),kahuna(burger)).
6. and(big(X),kahuna(X)).
7. _and(big(X),kahuna(X)).
8. (Butch kills Vincent).
9. kills(Butch Vincent).
10. kills(Butch,Vincent).

QUESTION :

Two individuals are cousins if they have a grandparent in common, but they are not the same person, and are not siblings. Using predicate grandparent(*X*, *Y*), meaning that *X* is a grandparent of *Y*, and predicate sibling(*X*, *Y*) meaning that *X* and *Y* are siblings, write a definition of cousin(*X*, *Y*) using Prolog notation, meaning that *X* and *Y* are cousins.

Answer :

```
cousin(X,Y) :- grandparent(Z,X),
               grandparent(Z,Y),
               not(X = Y),
               not(sibling(X,Y)).
```

ASSIGNMENT

```
mother(M, C).              /* M is the mother of C */
father(F, C).              /* F is the father of C */
male(X).                   /* X is male */
female(X).                 /* X is female */
```

Write Prolog rules for

1) grandmother(G, X). /* G is a grandmother of X */

2) sister(S, X). /* S is a sister of X */

3) half_brothers(B1, B2). /* B1 and B2 are male and have the same mother or the same father, but not both. */

References

- https://wwu-pi.github.io/tutorials/lectures/lsp/010_install_swi_prolog.html
- http://www.cse.unsw.edu.au/~billw/cs9414/notes/prolog/intro.html
- http://lpn.swi-prolog.org/lpnpage.php?pagetype=html&pageid=lpn-htmlse2
- http://www.dobrev.com/help/tut/The_anonymous_variable.html
- https://www.ics.uci.edu/~kibler/ics171/Lectures/Prolog.ppt
- https://www.cse.iitb.ac.in/~cs344/.../cs344-lect22-23-24-prolog-2008-3-12-16-17.ppt