Artificial Intelligence

Lecture 18

- **Prolog Programming for AI**

Prepared by:

Md. Mijanur Rahman, Prof. Dr.

Dept. of CSE, Jatiya Kabi Kazi Nazrul Islam University

Email: mijanjkkniu@gmail.com

Prolog Programming for AI

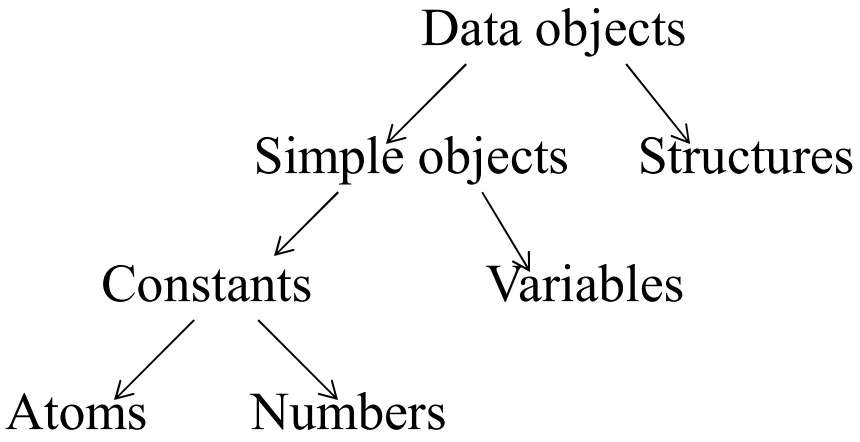
- **Outlines:**
  - Syntax of Prolog Program
  - Matching Operation
  - Input and Output
  - Meaning of Prolog Program
  - How to write a rule?
  - …

Syntax of Prolog Program…

- **Syntax of Prolog Program:**
  - Data objects
  - Atoms
  - Numbers
  - Variables
  - Structures
  - Terms

Syntax of Prolog Program…

- **Data objects:**
- The Prolog system recognizes the type of an object in the program by its syntactic form. This is possible because the syntax of Prolog specifies different forms of each type of data object.



Syntax of Prolog Program…

- **Atoms:**
- They can take more complicated forms- that is strings of the following characters:
  - **Upper case letters: A-Z**
  - **Lower case letters: a-z**
  - **Digits: 0-9**
  - **Special characters: + - * / < > = : . & _ ! ~**
- **Atoms can be formed in three ways:**

1) String of letters, digits and the underscore character '_', starting with lower-case letter:

   anna x x25 x_ x__y

2) Strings of special character:

   <--->======>… .:. ::=

3) Strings of characters enclosed in single quotes:

   'Tom' 'Sarah Jones'

Syntax of Prolog Program…

- **Numbers:**
- Numbers used in Prolog include integer numbers and real numbers. The syntax of integer is simple-
  - 1 1213 0 -95
- The simple syntax of real numbers:
  - 3.14 -0.000035 100.50

Syntax of Prolog Program…

- **Variables:**
- Variables are strings of letters, digits and underscore characters. They start with an upper-case letter or an underscore character:
  - X Result Sum _x23 Area_circle
  - hasachild(X) :- parent(X,Y).
  - hasachild(X) :- parent(X,_).
  - ?- parent(X,_).

Syntax of Prolog Program…

- **Structures:**
- Structure objects are objects that have several components. For example the **date** can be viewed as a structure with three components: day, month and year.
  - Date (1, may, 2001)
- **Geometric objects**
- A point in 2D space is defined by its two coordinates; a line segment is defined by two points; and a triangle can be defined by three points. For example-
  - P1 = point(1,1)
  - P2 = point(2,3)
  - S = seg(P1,P2) = seg(point(1,1),point(2,3))
  - T = triangle(point(4,2),point(6,4),point(7,1))

Syntax of Prolog Program

- **Terms:**
- The central data structure in Prolog is that of a **term**. In Prolog, the terms can be used to represent complex data objects.
- There are terms of four kinds:
  - atoms,
  - numbers,
  - variables, and
  - compound terms.
- Atoms and numbers are sometimes grouped together and called atomic terms.

Matching Operation

- The most important operation on terms is ***matching***. ***Matching*** is a process that takes as input two terms and checks whether they match.
- If the terms do not match we say that this process fails. If they do match then the process succeeds.
- Given two terms, we say that they ***match*** if:
  - They are identical, or
  - The variables in both terms can be instantiated to objects in such a way that after the substitution of variables by these objects the terms become identical.
- For example, the two terms date(D,M,2001) and date(D1,may,Y1).
  - D is instantiated to D1
  - M is instantiated to may
  - Y1 is instantiated to 2001

Input and Output…

- Reading data from files and outputting data to files.
- **read(X).**

  is used for reading terms from current input stream.
- **write(X).**

  is used for outputting term X on the current output file.

- **Example-6:** Read a number and write the cube of the number.

```
cube :- read(X), process(X).
process(stop) :- !.
process(N) :- C is N*N*N, write(C), cube.
```

?- cube.
2.
8
stop.

Input and Output

- Example-7: Read a number and write the cube of the number.

```
cube :-     write('Enter a number:'),
            read(X), process(X).
process(stop) :- !.
process(N) :-   C is N*N*N,
                    write('Cube of '), write(N),
                    write('is '),write(C),cube.
```

?- cube.
Enter a number: 2.
Cube of 2 is 8
Enter a number: stop.

- **LIVE Prolog program editor and execution:**
  - https://swish.swi-prolog.org/

Meaning of Prolog Program…

- Two levels of meaning of Prolog programs; namely-
  - The declarative meaning and
  - The procedural meaning
- The **declarative meaning** is concerned only with the relations defined by the program. It thus determines what will be the output of the program.
- On the other hand, the **procedural meaning** also determines how this output is obtained; that is, how the relations are actually evaluated by the Prolog system.

Meaning of Prolog Program…

- Consider a clause:
  - P :- Q,R.
  - Where P, Q and R have the syntax of terms.
- Some alternative declarative reading of this clause are:
  - P is true If Q and R are true.
  - From Q and R follows P.
- Two alternative procedural readings of this clause are:
  - To solve problem P, first solve the subproblem Q and then subproblem R.
  - To satisfy P, first Q and then R.

Meaning of Prolog Program…

- The procedural meaning specifies how Prolog answers questions. To answer a question means to try to satisfy a list of goals.
- **Example-8: Some facts and rules.**

```
PROGRAM
%Facts
big(bear).                  %Clause 1
big(elephant).              %Clause 2
small(cat).                 %Clause 3
brown(bear).                %Clause 4
black(cat).                 %Clause 5
gray(elephant).             %Clause 6

%Rules
dark(Z) :- black(Z).        %Clause 7: Anything black is dark
dark(Z) :- brown(Z).        %Clause 8: Anything brown is dark

QUESTION
?- dark(X), big(X).  %Who is dark and big?
```

Meaning of Prolog Program…

- **EXECUTION TRACE**

1) Initial goal list: **dark(X), big(X).**
2) Scan the program from top to bottom, first goal **dark(X)** match found at Clause7:

   dark(Z) :- black(Z).

   Replace first goal and the new goal is:

   black(X), big(X).
3) Scan to find a match with black(X), match found at clause5. **black(cat).** This clause has no body, so **X=cat.**

   big(cat).

PROGRAM

```
big(bear).          %Clause 1
big(elephant).      %Clause 2
small(cat).         %Clause 3
brown(bear).        %Clause 4
black(cat).         %Clause 5
gray(elephant).     %Clause 6

dark(Z) :- black(Z).    %Clause 7
dark(Z) :- brown(Z).    %Clause 8
```

QUESTION

```
?- dark(X), big(X).  %Who is dark and big?
```

Meaning of Prolog Program…

- **EXECUTION TRACE**

4) Scan the program for the goal **big(cat)**. No match found. Therefore backtrack to step(3) and undo the instantiation X=cat. Now the goal list is again:

   black(X) :- big(X).

   Continue scanning below the clause5. No clause found. Therefore backtrack to step 2 and continue scanning below clause 7. Clause 8 found.

   dark(Z) :- brown (Z).

   Replace the first goal giving:

   brown(X) , big(X).

PROGRAM

```
big(bear).              %Clause 1
big(elephant).          %Clause 2
small(cat).             %Clause 3
brown(bear).            %Clause 4
black(cat).             %Clause 5
gray(elephant).         %Clause 6

dark(Z) :- black(Z).    %Clause 7
dark(Z) :- brown(Z).    %Clause 8
```

QUESTION

```
?- dark(X), big(X).  %Who is dark and big?
```

Meaning of Prolog Program

- **EXECUTION TRACE**

5) Scan the program to match **brown(X)**. Found clause 4, **brown(bear)**. This has no body. The goal list shrinks to:

   big(bear).

6) Scan the program and find the clause **big(bear).** It has no body. So the goal list shrinks to empty. This indicates successful termination. So,

   X = bear.

PROGRAM

```
big(bear).                %Clause 1
big(elephant).            %Clause 2
small(cat).               %Clause 3
brown(bear).              %Clause 4
black(cat).               %Clause 5
gray(elephant).           %Clause 6

dark(Z) :- black(Z).      %Clause 7
dark(Z) :- brown(Z).      %Clause 8
```

QUESTION

```
?- dark(X), big(X).  %Who is dark and big?
```

Example: How to write a rule?

- **Example-9: Computing Greatest Common Divisor (GCD).**

For two integers X and Y, the GCD, D can be found as follows (with rules):

1) If X and Y are equal, then D is equal to X.

```
gcd(X, Y, D) :- X=:=Y, D is X.
```

2) If X<Y then D is equal to the GCD of X and Y-X.

```
gcd(X, Y, D) :- X<Y, Y1 is Y-X,
                gcd(X, Y1, D).
```

3) If Y<X then do the same as (2) with X and Y interchanged.

```
gcd(X, Y, D) :- Y<X, gcd(Y, X, D).
```

- **Prolog Program:**

```
gcd(X, Y, D):- X=:=Y,
               D is X.
gcd(X, Y, D):- X<Y,
               Y1 is Y-X,
               gcd(X, Y1, D).
gcd(X, Y, D):- Y<X,
               gcd(Y, X, D).
```

?- gcd(12, 27).

3

- **Practice this program with Input and output predicates.**

Prolog Programming for AI

TO BE CONTINUED…