Logic Programming: Prolog

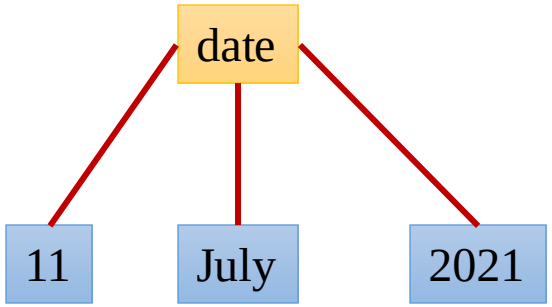
10/02/2025

Koustav Rudra

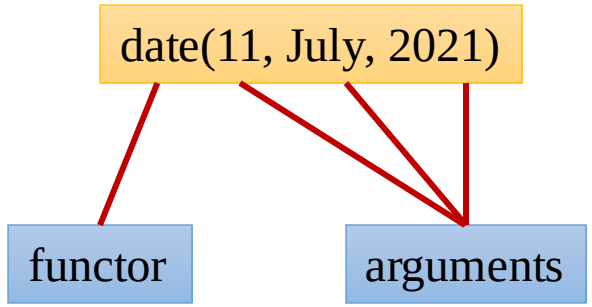
Prolog: Data Objects - Structures

- Structured Data Objects –
  - Structured data objects (or structures) are objects that have multiple components.
  - The components themselves can in turn be structures

  - e.g., date can be viewed as a structure with three components --- day, month, year
  - 11th July 2021 : date(11,July,2021)



Tree Representation



Prolog Representation

Structured Data Objects: Example

- P1: point(1,1)
- P2: point(2,3)
- S: seg(P1,P2): seg(point(1,1),point(2,3))
- T: triangle(point(4,Z),point(6,4),point(7,1))

- Structures can be naturally pictured as trees
- Prolog can be viewed as a language for processing trees

Prolog: Data Structure

- Lists
  - Lists of anything, symbolic lists
- Lists can be written as:
  - [item1, item2, …]
  - [Head|Tail]
    - Head is the first element in the list, remaining is the tail (list)
  - [item1, item2, …|Others]
    - Head consists of several items, followed by the tail which is other items [list]
- [a, b, c] = [a|[b,c]] = [a,b|[c]] = [a,b,c|[]]

- Items can be list as well
  - [[a,b], c, [d, [e,f]]]
  - The head of the above list is list [a,b]

List Examples: Membership

- member(X,Y) ➔ X is a member of list Y

- member(X,[X|Tail]).
- member(X,[Head|Tail]) :- member(X,Tail).

- a , [[b], [a,b], b]
  - Looking only at first level
  - How to find membership within sub-lists?

List Examples: Concatenation

- conc([],L,L).
- conc([X|L1],L2,[X|L3]) :- conc(L1,L2,L3).

- ?-conc([a], Z, [a,b]).
  - Z = [b]
- ?-conc([a], [b], Z).
  - Z = [a, b]

List Examples: Concatenation

- conc([],L,L)
- conc([X|L1],L2,[X|L3]) :- conc(L1,L2,L3)

- ?- conc([a,b],[c,d],[a,b,c,d])
- ?- conc([b],[c,d],[b,c,d])
- ?- conc([],[c,d],[c,d])

X = a, L1 = [b], L2 = [c,d], L3 = [b,c,d]

X = b, L1 = [], L2 = [c,d], L3 = [c,d]

- ?- conc([a],[b],[a,d])
- ?- conc([],[b],[d])

List Examples: Concatenation

- conc([],L,L)
- conc([X|L1],L2,[X|L3]) :- conc(L1,L2,L3)

- ?- conc([a],y,[a,b])
- ?- conc([],y,[b])
- y = [b]

X = a, L1 = [], L2 = y, L3 = [b]

Y = [b]

List Examples: Add and Delete

- Add(X,L1,L2)
  - Add(a,[],[a]).
  - Add(X,L1,[X|L1]).

- Del(X,[],[]).
- Del(X,[X],[]).
- Del(X,[X|Tail],Tail).
- Del(X,[Y|Tail],[Y|Tail1]) :- Del(X,Tail,Tail1).

Delete all instances of X

List Examples: Permutation

- permutation([],[]).
- permutation([X|L],P) :- permutation(L,L1), remove(X,P,L1).
- remove(X,[X],[]).
- remove(X,[X|L],L).
- remove(X,[Y|Tail],[Y|Tail1]) :- remove(X,Tail,Tail1).

List Examples: Permutation

- Permutation([],[]).
- Permutation([X|L],P) :- Permutation(L,L1), insert(X,L1,P).

- ?- Permutation([a,b,c,d],[d,c,a,b])

[X|L] = [a|b,c,d], L1 = [d,c,b] P = [d,c,a,b]

  - Permutation([b,c,d],L1), insert(a,L1,[d,c,a,b])
  - Permutation([c,d],L'), insert(b,L',L1)

  - Depth first search
  - Draw derivation tree?

Arithmetic and Logical operators

- We have +, -, *, /, mod
- The “is” operator forces evaluation
- ?- X is 3/2.
  - Will be answered X = 1.5
- We have
  - X>Y, X<Y, X>=Y, X=<Y
  - X=:=Y
    - X and Y are equal
  - X =\= Y
    - X and Y are not equal

Prolog: Inputs and Outputs

- write()
  - To write the output we can use the write() predicate
- | ?- write(56).
- 56
- yes
- | ?- write('hello’).
- hello
- yes
- | ?- write('hello'),nl,write('world’).
- hello
- world

Prolog: Inputs and Outputs

- read()
  - The read() predicate is used to read from console
- | ?- write('Write a number: ‘), read(Number).

- area :- write('Write a number: '), read(Number), process(Number).
- process(0).
- process(Number) :-
  - A is Number * Number,
  - write(‘Area of '),write(Number),write(': '),write(A),nl,
  - area.

Examples

- GCD of two numbers
  - gcd(0,X,X).
  - gcd(X,0,X).
  - gcd(X,X,X).
  - gcd(X,Y,D) :- X>=Y, Y1 is X mod Y, gcd(Y1,Y,D).
  - gcd(X,Y,D) :- X<Y, Y1 is Y mod X, gcd(Y1,X,D).

- Length of a list
  - length([],0)
  - length([_|Tail], N) :- length(Tail, N1), N is N1+1

Thank You