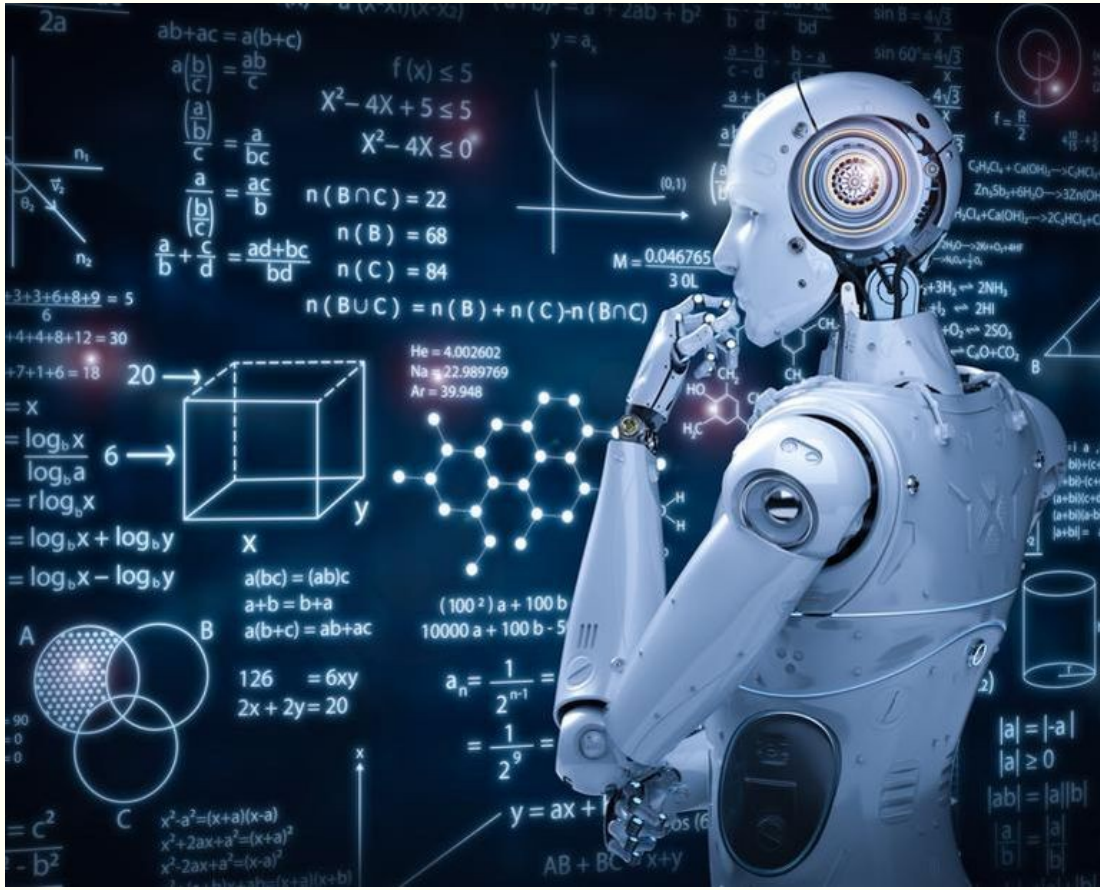
PROGRAMMING IN PROLOG

Ahana Patra



Agenda

- Proog in AI
  - Fact
  - Rule
  - Query
- Syntax of Clause
- Prolog Installation
- Prolog Operator
- Basic Programs
- Minimum, Maximum, Factorial, GCD, LCM
- List
  - Insertion, Deletion, Union, Intersection

Prolog in AI

- Prolog is a logic programming language that is used to create artificial intelligence.
- In order to come up with a query, or end goal, an artificial intelligence written in Prolog will analyze the relationship between a fact, a statement that is true, and a rule, which is a conditional statement.

Facts in Prolog

Facts –

- Facts are properties of objects, or relationships between objects;
- "Agnibha has phone number 1122334455", is written in Prolog as:

  phoneno(agnibha, 1122334455).

- It should be noted that:
  - ✓ Names of properties/relationships begin with lower case letters.
  - ✓ The relationship name appears as the first term
  - ✓ Objects appear as comma-separated arguments within parentheses.
  - ✓ A period "." must end a fact.
  - ✓ Objects also begin with lower case letters. They also can begin with digits (like 1234), and can be strings of characters enclosed in quotes e.g. color(penink, 'red').
  - ✓ phoneno(agnibha, 1122334455). is also called a predicate or clause.

Facts in Prolog

Facts about a hypothetical Department of Physics –

```
% teaches(X, Y): person X teaches in course Y
        teaches(sudhir, course001).
        teaches(tapas, course002).
        teaches(pranab, course003).
        teaches(joydeb, course001).

% student(X, Y): student X studies in course Y
        studies(suparna, course001).
        studies(santanu, course001).
        studies(sudip, course002).
        studies(srobona, course003).
        studies(subir, course003).
        studies(swarup, course003).
```

Together, these facts will form Prolog's database.

Rules in Prolog

Rules –

- Consider the following case which produces a general rule –
  - ✓ One teacher will guide a student if that student studies that very course id on which the teacher teaches
  - ✓ In Prolog this will be written as:

```
guide(Teacher, Student) :-
        teaches(Teacher, Courseid),
        studies(Student, Courseid).
```

- Facts are unit clauses and rules are non-unit clauses.
- Variable names will start with a capital letter.

Query in Prolog

Goal or query –

- Queries will be based on facts and rules. We can ask questions based on the stored information.

- Suppose we want to know if sudhir lectures in course001 or not then we can ask:

  ?- teaches(sudhir, course001).
  Yes

- To answer this query, Prolog consults its database to see if this is a known fact or not.

- We can also ask –

  ?- teaches(sudhir, X).
  X = course001

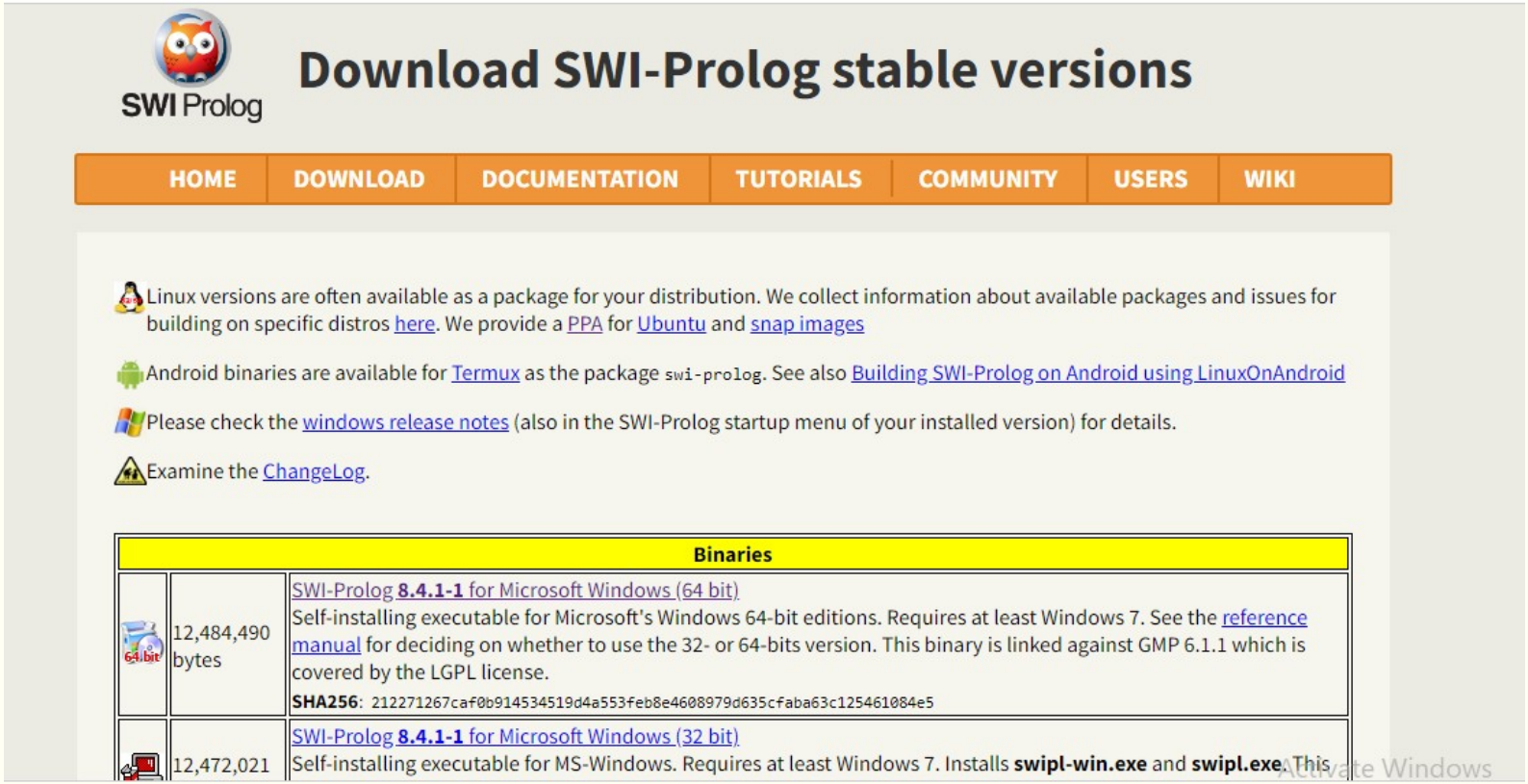
- if answer is true or yes, then the query succeeded
- if answer is false or no, then the query failed.

Syntax of Clause

- ":-" means "if" or "is implied by". Also called the neck symbol.
- The left hand side of the neck is called the head.
- The right hand side of the neck is called the body.
- The comma, "," stands for and / conjunction.
- The semi-colon ";" stands for or / disjunction.

A program consists of clauses. These are of three types: **facts, rules and questions**. A procedure is a set of clauses about the same relation.

Prolog Installation



SWI Prolog

Download SWI-Prolog stable versions

HOME | DOWNLOAD | DOCUMENTATION | TUTORIALS | COMMUNITY | USERS | WIKI

Linux versions are often available as a package for your distribution. We collect information about available packages and issues for building on specific distros here. We provide a PPA for Ubuntu and snap images

Android binaries are available for Termux as the package `swi-prolog`. See also Building SWI-Prolog on Android using LinuxOnAndroid

Please check the windows release notes (also in the SWI-Prolog startup menu of your installed version) for details.

Examine the ChangeLog.

Binaries		
64-bit	12,484,490 bytes	SWI-Prolog **8.4.1-1** for Microsoft Windows (64 bit) Self-installing executable for Microsoft's Windows 64-bit editions. Requires at least Windows 7. See the reference manual for deciding on whether to use the 32- or 64-bits version. This binary is linked against GMP 6.1.1 which is covered by the LGPL license. **SHA256**: `212271267caf0b914534519d4a553feb8e4608979d635cfaba63c125461084e5`
	12,472,021	SWI-Prolog **8.4.1-1** for Microsoft Windows (32 bit) Self-installing executable for MS-Windows. Requires at least Windows 7. Installs **swipl-win.exe** and **swipl.exe**. This

Operators in Prolog

Arithmetic Operators

+	addition
-	subtraction
*	multiplication
/	real division
//	integer division
mod	modulus
**	power

Logical Operators

```
a : - b.        % a if b
a : - b , c.    % a if b and c.
a : - b ; c.    % a if b or c.
a : - not b.    % a if b fails
```

Comparison Operators

X > Y	X is greater than Y.
X < Y	X is less than Y.
X >= Y	X is greater than or equal to Y.
X =< Y	X is less than or equal to Y.
X =:= Y	the values of X and Y are equal.
X =\= Y	the values of X and Y are not equal.

Basic Programs

File Name	Program Code	Load	Run
hello.pl	/*Prolog Hello World Program*/ hello :- write('Hello World, This is my first program in Prolog!!!').	['hello.pl']. Or [hello]. Or consult(hello). Or consult('hello.pl').	hello.
equal.pl	equal(X,Y):-X is Y.	[equal].	equal(3,6).

Minimum

Question	Program Code	Input	Output
Minimum of two numbers	minimum(M,N,M):- M =< N. minimum(M,N,N):- N=< M.	minimum(10,20,X).	X = 10.
		minimum(90,20,30).	false.

Maximum

Question	Program Code	Input	Output
Maximum of three numbers	max(A,B,C):- A>B,A>C,write("Max is "),write(A). max(A,B,C):- C>A,C>B,write("Max is "),write(C). max(A,B,C):- B>A,B>C,write("Max is "),write(B).	max(20,30,50).	Max is 50 true.
		max(30,30,30).	false.

Factorial

Factorial of a number.	fact(0,1). fact(N,R):- N>0, N1 is N-1, fact(N1,F1), R is N*F1.	?- fact(5,120). true . ?- fact(5,150). false . ?-fact(5,R). R=120

GCD

GCD of two numbers.	gcd(X,Y,G):-X=:=Y,G=X. gcd(X,Y,G):- X<Y, Y1 is Y-X, gcd(X,Y1,G). gcd(X,Y,G):-X>Y, gcd(Y,X,G).	gcd(3,8,GCD).	GCD = 1
		gcd(12,18,GCD).	GCD = 6

LCM

LCM of two numbers.	gcd(X,Y,G):-X=:=Y,G=X. gcd(X,Y,G):- X<Y,Y1 is Y-X, gcd(X,Y1,G). gcd(X,Y,G):-X>Y, gcd(Y,X,G). lcm(X,Y,LCM):-gcd(X,Y,GCD), LCM is X*Y//GCD.	lcm(3,8,LCM). lcm(12,18,LCM).	LCM =24 LCM =36

Lists

- The Lists is a data structure that can be used in different cases for non-numeric programming. Lists are used to store the atoms as a collection.
- Representation of lists in Prolog
- Basic operations on prolog such as Insert, delete, update, append.
- Set operations like set union, set intersection, etc.

Representation of lists

- A list can be either **empty** or **non-empty**. In the first case, the list is simply written as a Prolog atom, []. In the second case, the list consists of two things as given below −
- The first item, called the **head** of the list;
- The remaining part of the list, called the **tail**.
- we have a list, L = [a, b, c]. If we write Tail = [b, c] then we can also write the list L as L = [a | Tail]. Here the vertical bar (|) separates the head and tail parts.
- So the following list representations are also valid −
- [a, b, c] = [x | [b, c]]
- [a, b, c] = [a, b | [c]]
- [a, b, c] = [a, b, c | [ ]]

Insert an element in a List

Question	Program Code	Input	Output
Insert an element in a list.	insert(X,L,[X|L]).	insert(k,[a],L).	L = [k, a].
		insert(5,[a,r,3],L).	L = [5, a, r, 3].
Insert an element in a last position of a list.	insert(X,[],[X]). insert(X,[Y|Tail],[Y|Tail1]):- insert(X,Tail,Tail1).	insert(k,[a],L). insert(5,[a,r,3],L).	L = [a,k]. L = [a, r, 3,5].

Delete an element in a List

Delete an element in a list.	del(X,[X|Tail],Tail). del(X,[Y|Tail],[Y|Tail1]):- del(X,Tail,Tail1).	del(a,[a,b,a,a],L).	L = [b, a, a] ; L = [a, b, a] ; L = [a, b, a] ; false.
		del(6,[4,6,7],L).	L = [4, 7] ; false.

Union

Union of two lists	list_member(X,[X|_]). list_member(X,[_|TAIL]) :- list_member(X,TAIL). list_union([X|Y],Z,W) :- list_member(X,Z),list_union(Y,Z,W) . list_union([X|Y],Z,[X|W]) :- \+ list_member(X,Z), list_union(Y,Z,W). list_union([],Z,Z).	list_union([a,b,c,d,e], [a,e,i,o,u],L3).	L3 = [b, c, d, a, e, i, o, u] .

Intersection

Intersection of two lists	list_member(X,[X|_]). list_member(X,[_|TAIL]) :- list_member(X,TAIL). list_intersect([X|Y],Z,[X|W]) :- list_member(X,Z), list_intersect(Y,Z,W). list_intersect([X|Y],Z,W) :- \+ list_member(X,Z), list_intersect(Y,Z,W). list_intersect([],Z,[]).	list_intersect([a,b,c,d, e],[a,e,i,o,u],L3).	L3 = [a, e] .


THANK YOU