Introduction to Prolog

Dr. Muhammad Shuaib Qureshi

Goal-Oriented Programming

- It is a language in which the satisfaction of goals is the basis of program execution.

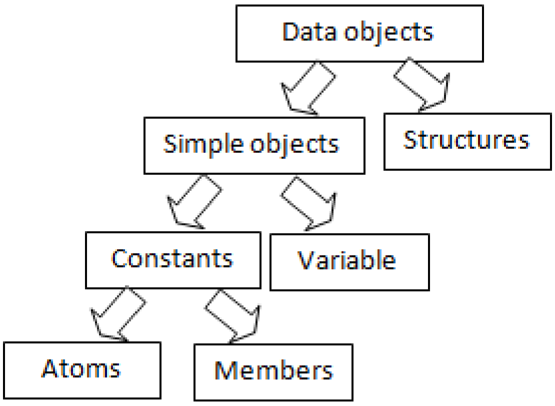
OR

- A language in which the set of goals to be satisfied is an essential part of a function or procedure body.
- Such languages include Prolog and other logic programming languages.

Introduction to PROLOG

- Programming in Logic.
- Used for symbolic and non-numerical computation.
- Has a built in intelligent search mechanism.
- Can handle complex problems in compact programs.
- Writing a program in Prolog means writing facts and rules which together comprise knowledge base.
- Facts and rules use predicates which represent relationships among data objects.

Data Objects



Atoms:

- Strings of letters, digits, underscore character starting with lower case letter:

  sarah_jones, x25, x__y, x_yAB

- String of special characters:

  <-->, ===>, ….

- Strings of characters enclosed in single quotes:

  ‘India’, ‘Tom’

- Numbers:

  Include integers and real numbers i.e. 1, 3131, -0.0035, 3.14

Variables:

- String of letters, digits and underscore characters that starts *either* with an upper-case letter *or* with an underscore:
  Y, Child, _a23, Student_List

Structures:

- Objects that have many components
- Components can themselves be structures
- *Functor* is used to combine components into single structure

date(1, jan, 2021), date(Date, Month, 2021)

date(31, cat, -4.3), segment(point(1,1),point(3,3))

Functors are recognized by:

- Name
- Number of arguments (Arity)

Predicate:

- A predicate consists of a head and a number of arguments
- Is a function which returns true/false

Example:

father(Ali, Naveed). %Ali is father of Naveed.

Clauses:

- ***Facts***
  - Declare things that are unconditionally true
  - Have a head and empty body

Examples:

brother(Akram, Saleem). %Akram is brother of Saleem

likes(Abid,dogs). % Abid likes dogs

- ***Rules***
  - Declare things (predicates) that are true, depending on a given condition
  - Have a non-empty body

Example:

mother(X,Y):-parent(X,Y), female(X).

X is mother of Y if X is parent of Y and X is female.

- ***Recursive rules***

Example:

Rule 1: predecessor(X,Z):- parent (X,Z).

Rule 2: predecessor(X,Z):- parent(X,Y),predecessor(Y,Z)

Queries:

- Asking the program what is true
- Have empty head

Example:

? parent(Nabeel, Hadi). % Is Nabeel parent of Hadi?

Example:

? parent(Nabeel, X).   % Find X such that Nabeel is parent of X.

Matching:

Two terms (eg. predicate instance) will match if:

- They are identical, or
- The variables in the terms should be instantiated such that after substitution of variables by these objects, the terms become identical.

Example:

?date(D,M,2021)=date(10,jan,Y)

D=10

M=jan

Y=2021

PROLOG execution summary:

- At each stage there is a list of goals to be satisfied.
- PROLOG selects the leftmost sub goal and searches for a rule/fact head which matches it.
- The matching process can bind variables.
- The first sub goal is then replaced by the body of the matching rule/fact, yielding a new list of sub goals.
- If there are no (more) matching rule/fact (sub goal fails!)

or if the user asks for the next solution, PROLOG backtracks.

- Unless all instantiation fails for a given rule, next rule is not explored.
- System remembers where call rests in AND/OR graph and what instantiations are there for each term.
- The most recent choice of rule/fact is undone, including any variable bindings which were made when the head was matched.
- PROLOG searches for an alternative matching rule/fact.

Program Exercise:

```
% Facts:
      parent(pam,bob).
      parent(tom,bob).
      parent(tom,liz).
      parent(bob,ann).
      parent(bob,pat).
      parent(pat,jim).
% Rules:
      predecessor(X,Z) :- parent(X,Z).        % R1
      predecessor(X,Z) :- parent(X,Y), predecessor(Y,Z).  % R2
```

Queries and Output:

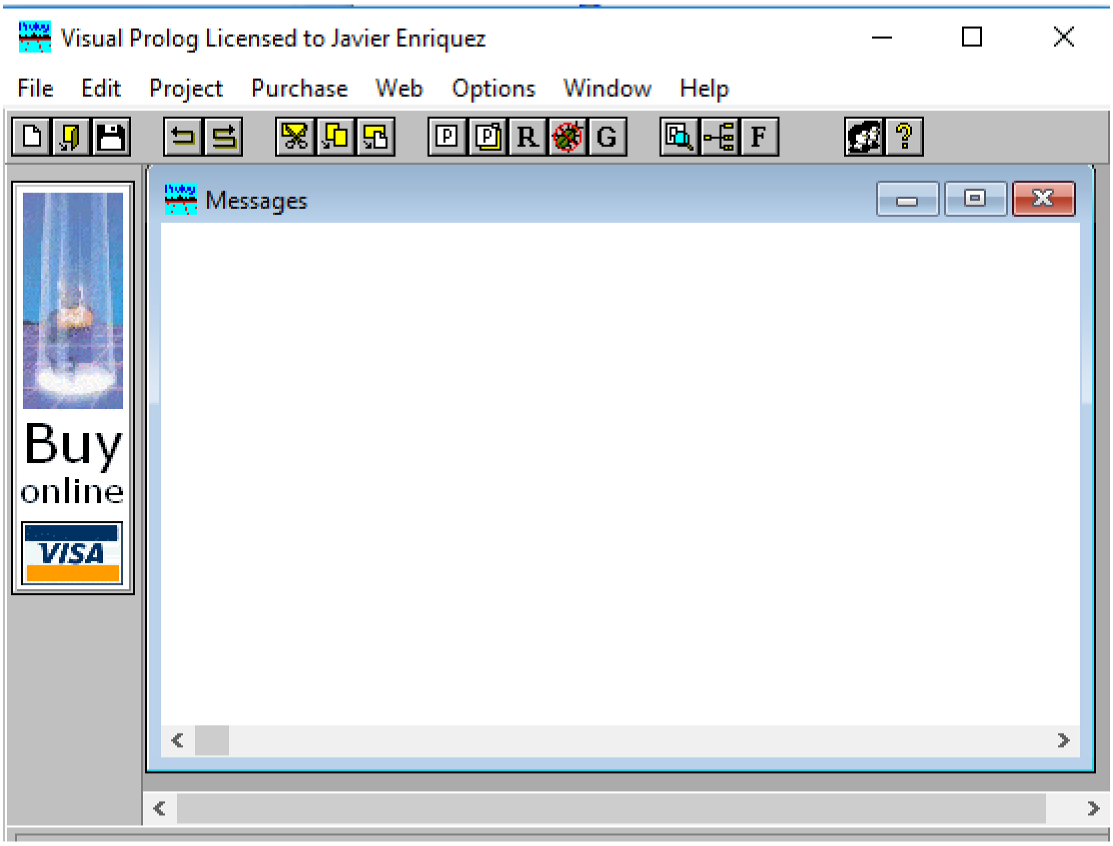
```
? predecessor(tom,pat).
Yes.

? predecessor(bob,X).
X=ann
X=pat
X=jim
No.
```

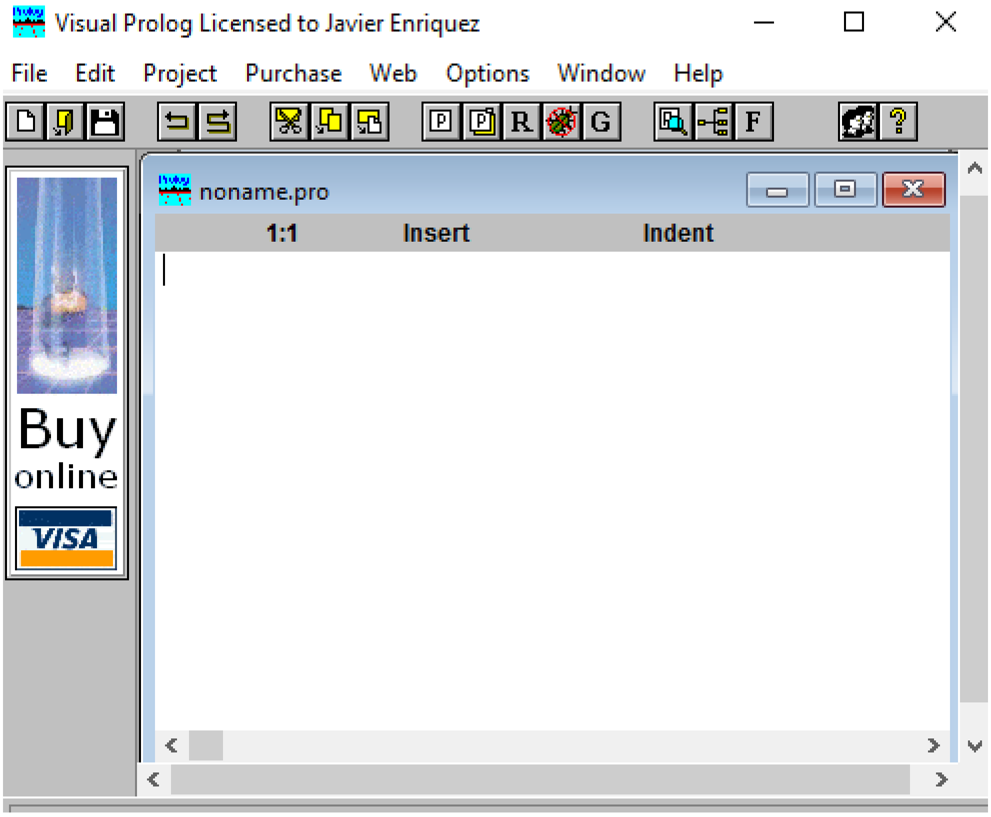
How to run Visual Prolog:

- Go to:

Start menu → **all programs** →visual prolog 5.2 personal edition →vip32.



- Click on File menu and select New



Type the program in the white space.

To represent the “likes” facts in VPROLOG:

- File➔New➔noname.pro
- Then in the predicate section write the declaration of the used predicates:

```
PREDICATES
nondeterm likes (symbol,symbol)
```

- ***In clauses section write the facts:***

```
CLAUSES
likes (ali,football).
likes (ali,tenis).
likes (ahmad,tenis).
likes (ahmad,handball).
likes (samir,handball).
likes (samir,swimming).
likes (khaled,horseriding).
```

Queries as goals in PROLOG:

- ***To supply a query in PROLOG put it in the goal section as follows:***

```
GOAL
likes (ali, football).
```

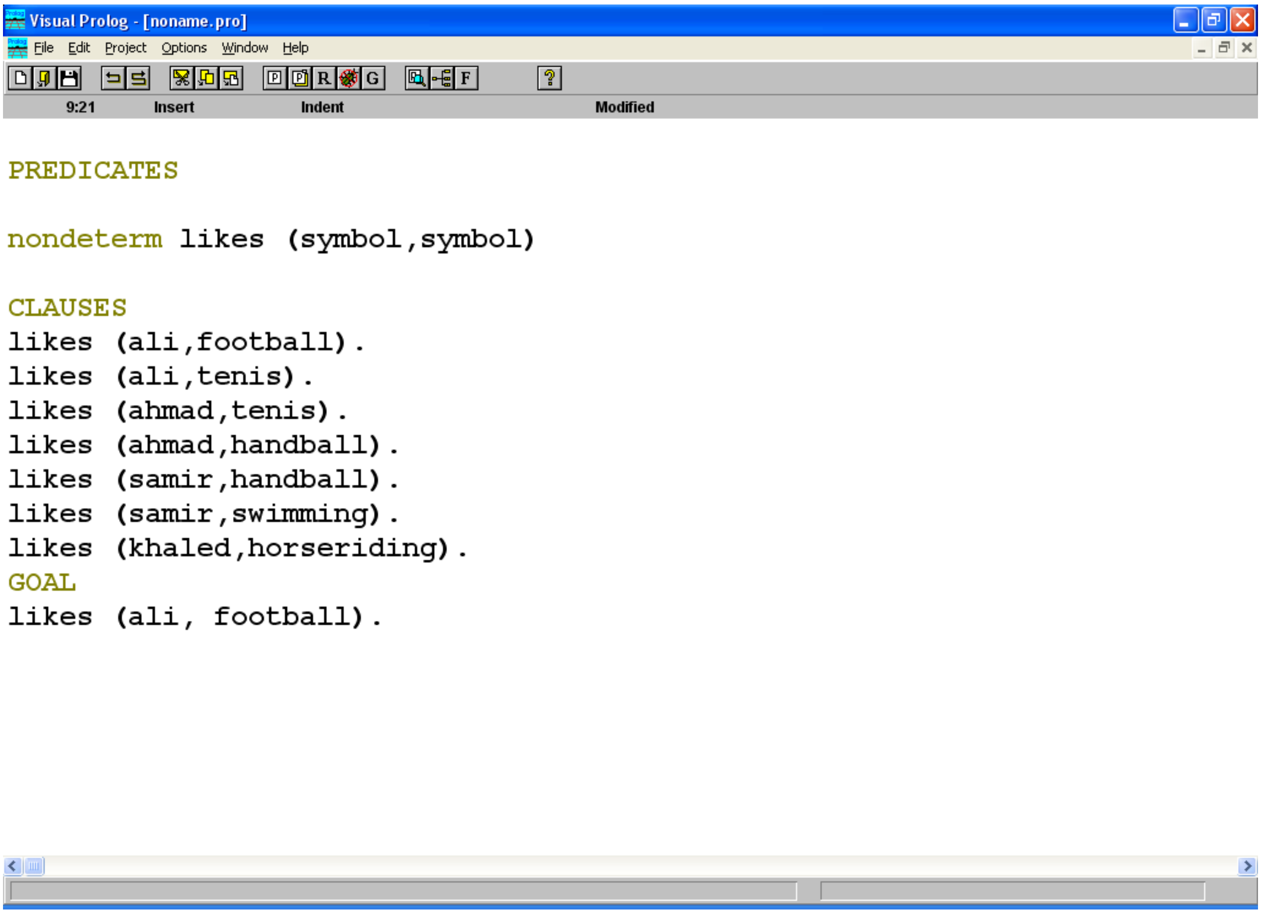
- ***Then press Cntrl+G***

  The output will be Yes or No for concrete questions.

Concrete questions: (queries without variables)

Example:

```
GOAL
likes (samir, handball).%  Yes
likes (samir, football). % No
```


Visual Prolog - [noname.pro]
File Edit Project Options Window Help
9:21
Insert
Indent
Modified

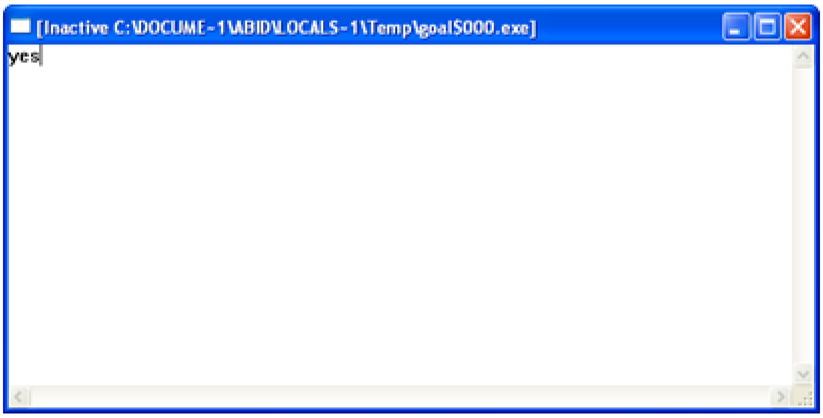
```
PREDICATES

nondeterm likes (symbol,symbol)

CLAUSES
likes (ali,football).
likes (ali,tenis).
likes (ahmad,tenis).
likes (ahmad,handball).
likes (samir,handball).
likes (samir,swimming).
likes (khaled,horseriding).
GOAL
likes (ali, football).
```

Then click on “G” to run the goal.

The result is ‘yes’ as ali likes football.



```
yes
```

Queries with variables:

- ***To know all sports that Ali likes:***

  likes (ali,What). Or

  likes (ali, X). X is a variable.

- ***To know which person likes tenis:***

  likes (Who,tenis).

- ***To know who likes what (i.e all likes facts)***

  likes (Who,What).   Or

  likes (X,Y).   where X is “Who” and Y is “What” .

  Variable must be in upper case or first letter of variable must be in upper case.

Compound queries with one variable:

- ***To list persons who like tennis and football***, the goal will be

  likes(Person, tennis), likes (Person, football).

- ***To list games liked by ali and ahmad.***

  likes (ali,Game), likes (ahmad,Game).

Compound queries with multiple variables:

To find persons who like more than one game:

```
                    likes (Person, G1), likes (Person,G2), G1<>G2.
```

To find games liked by more than one person:

```
                    likes (P1,Game),  likes(P2,Game), P1<>P2.
```

Facts containing variables:

Assume we add the ***drinks*** facts to the ***likes*** database as follows:

```
PREDICATES
   drinks (symbol, symbol)
CLAUSES
   drinks (ali, pepsi).
   drinks (samir, lemonada).
   drinks (ahmad, milk).
```

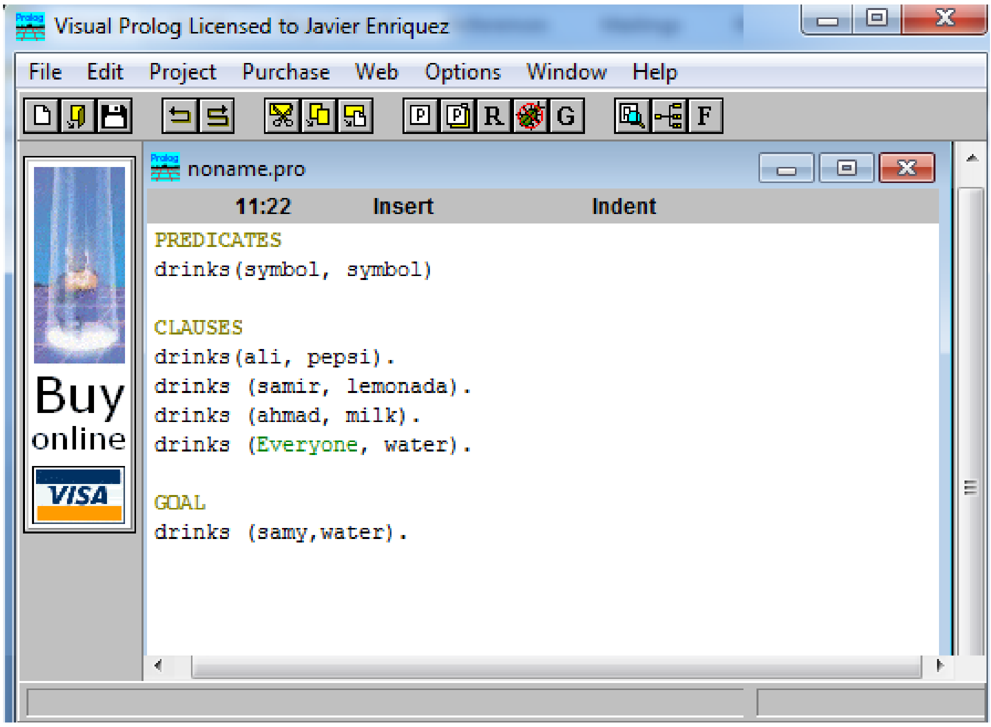
- *To add the fact that all persons drink water:*

```
    drinks (Everyone, water).
```

- *If we put a goal like:*

```
   drinks (samy, water).
   drinks (ahmad, water).
```

⇨ **The answer will be YES.**

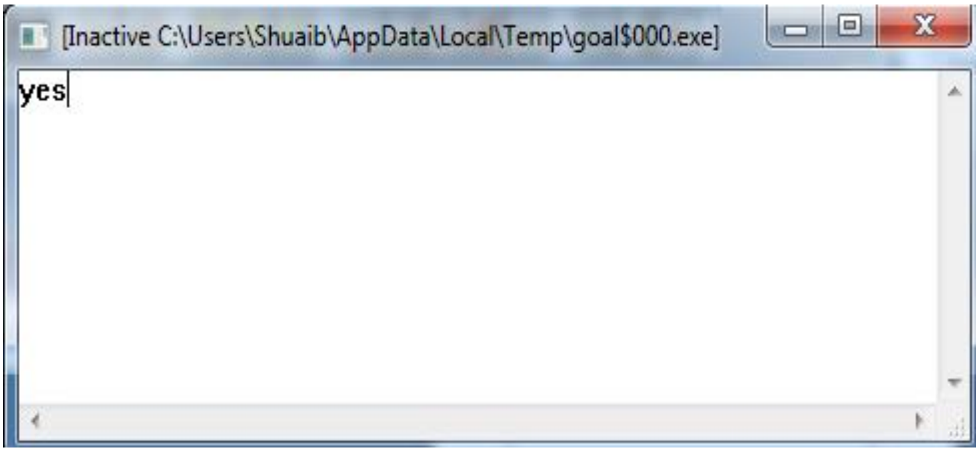

Visual Prolog Licensed to Javier Enriquez
File Edit Project Purchase Web Options Window Help
noname.pro
11:22 Insert Indent
Buy online
VISA

```
PREDICATES
drinks(symbol, symbol)

CLAUSES
drinks(ali, pepsi).
drinks (samir, lemonada).
drinks (ahmad, milk).
drinks (Everyone, water).

GOAL
drinks (samy,water).
```

Output:



```
yes
```

Rules:

- Rules are used to infer new facts from existing ones.
- A rule consists of two parts: ***head*** and ***body*** separated by the symbol **:-** .

To represent the rule that expresses the facts that two persons are friends if they both like the same game:

friends (P1,P2):- Head of the rule

likes (P1, G), likes (P2, G), P1< >P2. Body of the rule

(Here P1 and P2 are friends, due to P1 and P2 likes the same Game.)

Exercise:

1. Write a simple Prolog program for the following output

1. List all the game which ali likes.
2. Find all the persons who like cricket.
3. List all the persons and the games they like.
4. Find the games like by ali and ahmed.

2. Declare your own predicates, clauses and write a complete prolog program to implement the rule given below.

1.
```
friends (P1,P2):-
        likes (P1,G), likes (P2,G), P1<>P2.
```

(Hint: find all friends.)

2. Assume the following “likes” facts knowledge base

```
likes (ali, football).
likes (ali, tennis).
likes (ahmad, tennis).
likes (ahmad, handball).
likes (samir, handball).
likes (samir, swimming).
likes (khaled, horseriding).
```

Find all the people who like more than one game.

Extra (supplementary) Materials

Prolog Programming for Artificial Intelligence by Ivan Bratko, 4th edition.