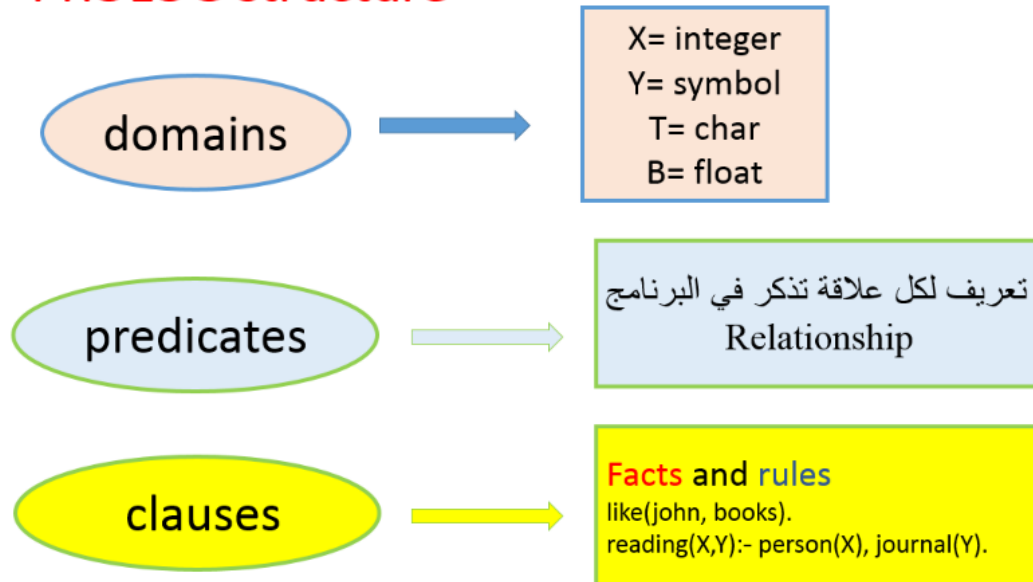


PROLOG structure



PROLOG_fact

The relationship has a specific order, johns own the book, but the book dose not owns john, and this relationship represent **fact**.

Example:

“John owns the book”
Owns (john,book)

Relationship(object1,object2)

- **rule** used to describe relationship between objects.

Example: the rule” two people are sisters if they are both female and have the same parents”

1. Tell us something about what it means to be sisters.
2. Tell us how to find if two people are sisters, simply: check to see if
They are both female have the same parents.

Sisters (X,Y)

Prolog language component _ Facts

• Syntax of fact: كيف نكتب fact في لغة prolog

1. The name of all relationship and objects must begin with a lower-case letter, for example **likes (john, mary)**.

العلاقات والحقائق تبدأ بحرف صغير

2. The relationship is written first, and the objects are written separated by **commas**, and enclosed by a pair of round **brackets**.

Like (john, mary).

العلاقة تكتب أولاً ثم الحقائق تكتب داخل اقواس يفصل بينها فارزة

3. The full stop character **!** Must come at the end of fact.

كل حقيقة تنتهي ب نقطة .

3

Relationship Representation

• Tom is John's child → `child(tom , john)`

• Ann is Tom's child → `child(ann, tom)`

• John is Mark's child → `child(john, mark)`

• Alice is John's child → `child(alice, john)`

Prolog language component _ Rules

- Rules are used when you want to say that a fact depends on a group of other facts
 1. One fact represents the head (**conclusion**).
 2. The word **if** used after the head and represented as “:-‘.
 3. One or more fact represents the requirement (condition).

If (condition) then (conclusion)

[Conclusion: - condition]

Rule Representation

- The grandchild of a person is a child of a child of this person"

For all X and Y , grandchild(X, Y) if
there exists a Z such that child(X , Z) **and** child(Z, Y)

grandchild(X , Y) :- child(X, Z) , child(Z, Y).

الثوابت والمتغيرات

1. الأعداد الصحيحة (Integers) وهي مجموعة الأعداد الصحيحة الموجبة أو السالبة المحصورة بين العددين (-32,765) و (32,765) , مثل:

2. العناصر (Atoms)

العنصر (Atom) هو عبارة عن سلسلة من الحروف أو الأعداد أو الحروف الخاصة والتي تصف اسم أي شيء (Object) أو علاقة (Relationship) ما يجب أن يتحقق فيه الشروط الثلاثة التالية :

1- ألا يبدأ بعدد صحيح أو حرف كبير Capital أو العلامة (_) (underscore).

2- ألا يحتوي على علامة (-). (hyphen).

3- إذا احتوى العنصر على أي علامة من العلامات السابقة يجب أن ينحصر بين علامتي التنصيص ("").

أمثلة العناصر ما يلي: mohmmad

chapter_10

"Abc"

و الأمثلة التالية ليست عناصر Atoms

456Vector بدا برقم

Street بدا بحرف كبير

_large_Number بدا بعلامة اندر سكور

slam-mosa يحتوي علامة -

- المثال التالي يوضح كيفية استخدام المعامل (,) اي (and)

Likes (ali , mohamad) , like (omar , mohamad)

- أما المثالين التاليين فيوضحان كيفية استخدام المعامل (or) وكلاهما يؤدي نفس المعنى :

Likes (ali , mohamad) ; like (omar , mohamad)?

Likes (ali , mohamad) | like (omar , mohamad)?

Homework

- \Every mother loves her children
- " Mary is a mother and Tom is Mary's child"

3. Questions

السؤال يكتب في ال Goal

Example:

The following fact owns (mary , book).

We can ask: does mary own the book in the following manner:

Goal:

طريقة السؤال الأولى :

Owens (mary ,book)

السؤال يكتب في ال Goal بهذه الطريقة

Prolog will search through the database you typed before, it look for facts that match the fact in the question.

للإجابة عن السؤال في لغة برولوج ... يتم البحث في كل الحقائق التي أدخلت للبرنامج للمطابقة

If prolog finds a fact that matches the question,
Then answer will be **Yes**, otherwise the answer is **No**.

Variables

طريقة السؤال الثانية: نستخدم متغيرات داخل

If we want to get more interest information about fact or rule, we can use **variable** to get more than Yes/No answer.

*variables dose not name a particular object but stand for object that we cannot name.

*variable name must begin with capital letter.

Z, X, Y, N

*using variable we can get all possible answer about a particular fact or rule.

نحصل على كل الإجابات الممكنة

Goal:

Owms (mary ,X)

السؤال بالطريقة الثانية

Answer : X= book

Example:

Fact

Like (john, mary).

Like (john, flower).

Like (ali, mary).

الحقائق

Question:

1. Like (john,X)

X= mary

X = flower

2. like(K, mary)

K=john

2. Like(X, Y)

X=john Y=flower

X=john Y=mary

X=ali Y=mary

Example:

لديك الحقائق الآتية

Age(a,10).

Age(b,20).

Age(c,30).

الأسئلة الآتية

Goal:

1.Age(a,X).

X=10

2.age(X,20).

X=b

3.age(X,Y).

X=a Y=10,

X=b Y=20,

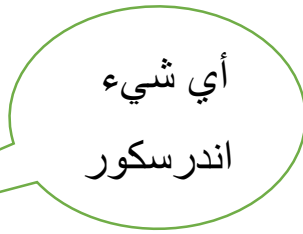
X=c Y=30.

4.Age(_,X).

X=10 , X=20, X=30. ‘_’ means don't care

5.Age(_,_).

Answer :Yes



أي شيء
اندرسكور

Q: write a program in Prolog language contains

- 1- Facts : (elen and mark likes tennis, Elin and john likes football, tom likes baseball, eric likes swimming). 6 facts
- 2- Rules: (bill likes whatever tom likes)
(mark likes Activity or tom likes the same Activity)

Solution: In the editor we have to write the following code

```
domains
name,sport=symbol
predicates
likes(name,sport)
clauses
likes(elen,tennis).
Likes("Elen",football).
likes(john,football).
likes(tom,baseball).
likes(eric,swimming).
likes(mark,tennis).
likes(bill, Activity) if likes(tom, Activity).
likes(mark, Activity) :- likes(tom, Activity).
```

Now to run the program : use Alt+R to go to Dialog window then write Likes(X,Y) and press enter to get the solutions contained all possible relations

في برنامج برولوج تقسم الشاشة الى اربع اجزاء

Editor يكتب فيه نص البرنامج

Dialog يكتب فيه تنفيذ البرنامج و نجد فيه النتائج المتوقعة

Messages تظهر فيه رسائل النظام والاطفاء الممكنة وما يقوم به الكومبايلر

Trace نافذة يظهر فيها الاختبار الذي نقوم به اثناء تنفيذ البرنامج

ملاحظات : في **Editor** نكتب نص البرنامج و الذي يكون مقسم الى ثلاث اجزاء كما هو موضح في المثال اعلاه

تعريف لكل متغير يستخدم في البرنامج 1- domains

تعريف لكل علاقة تذكر في البرنامج predicates -2

كل الحقائق التي نحتاجها في البرنامج وكل سطر في هذا الجزء يجب ان ينتهي ب
-3 clauses
(.)

Q: write a program in Prolog language contains

1- Facts : (Sara and jak reads book, robin and tomas reads magazine , alex read novel, sam read poetry). 6 facts

2- Rules: (sam read whatever alex read)
(tomas read whatever and Sara read the same)

Run the program.

Q: write a program in Prolog language contains

1- Facts : (ahmed, father of [mohammed, ali and sama] noor is mother to the same children). 6 facts

The relationship between mohammed and ali [brother]

domains

name1,name2=symbol

predicates

father(name1,name2)

mother(name1,name2)

brother(name1,name2)

clauses

father (ahmed,mohammed).

father (ahmed,ali).

father (ahmed,sama).

mother(noor, mohammed).

mother(noor, ali).

mother(noor, sama).

علامة , تعني And

علامة ; تعني Or

brother (ali,Y) :- father (ahmed, Y) , mother(noor, Y).

ماهي الأسئلة الممكن عملها في goal بلنسية لهذا البرنامج

Q: Write a program in prolog lang. find the type of animal who can swim?

Domains

X=symbol

predicates

type(X,X)

is_a(X,X)

lives(X,X)

can_swim(X)

clauses

type(ungulate, animal).

type(fish, animal).

is_a(zebra, ungulate).

is_a(herring, fish).

is_a(shark, fish).

lives(zebra, on_land).

lives(frog, on_land).

lives(frog, in_water).

lives(shark, in_water).

can_swim(Y) :-

type(X, animal) ,

is_a(Y, X) ,

lives(Y, in_water).

Goal : can_swim(N)

N=herring

N= shark

Goal : Type(X,Y)

Goal :is_a(V,B)

Q: write prolog program to read integer value and print it.

domains

I = integer

predicates

print.

clauses

print :- write ("please read integer number "), readint(X),
write("you read",X).

Goal Print.

Output:

Please read integer number 4

You read 4

Q: write prolog program that take two integer input us integer

and print the greater.

domains

i= integer

predicates

greater (i,i)

clauses

greater(X,Y):- X>Y,write("the greater is",X).

greater(X,Y):- X<Y write (" the greater is ",Y).

Goal

Greater(4,3).

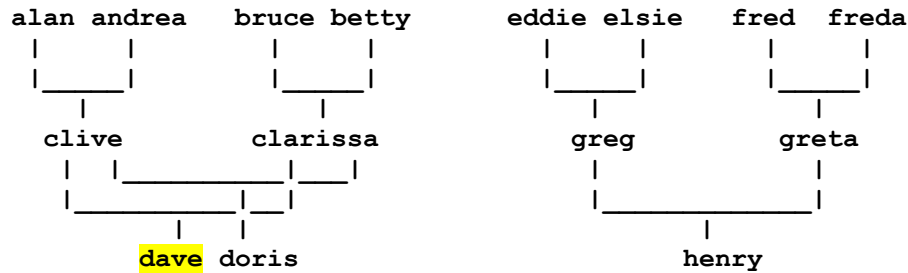
Output:

The greater is 4

المحاضرة التاسعة / ذكاء اصطناعي / عملي

Example 1: Parent and ancestor (الاب و السلف) الجد او الشخص الأعلى في شجرة العائلة)

/* Suppose we have a family tree like this :



which is defined in Prolog by the following 3 sets of predicates:

```
% parent(Parent, Child).
```

```
parent(alan, clive).
parent(andra, clive).
parent(bruce, clarissa).
parent(betty, clarissa).
parent(clive, dave).
parent(clarissa, dave).
parent(clive, doris).
parent(clarissa, doris).
parent(eddie, greg).
parent(elsie, greg).
parent(fred, greta).
parent(freda, greta).
parent(greg, henry).
parent(greta, henry).
```

```
% male(Person).
% This Person is male.
```

```
male(alan).
male(bruce).
male(clive).
male(dave).
male(eddie).
male(fred).
male(greg).
```

```
male(henry).
```

```
% female(Person).  
% This Person is female.
```

```
female(andrea).  
female(betty).  
female(clarissa).  
female(doris).  
female(elsie).  
female(freda).  
female(greta).
```

```
% married(Person1, Person2).  
% Person1 is married to Person2.
```

```
married(alan, andrea).  
married(bruce, betty).  
married(clive, clarissa).  
married(eddie, elsie).  
married(fred, freda).  
married(greg, greta).
```

```
% PROBLEM 1  
% How do you find out if someone is the ancestor of someone  
else ? الحفيد
```

```
ancestor(A, B):-  
    parent(A, B). % A is B's ancestor if  
                  % A is B's parent.
```

```
ancestor(A, B):-  
    parent(P, B),  
    ancestor(A, P). % A is B's ancestor if  
                  % some person P is B's parent and  
                  % A is P's ancestor.
```

```
% PROBLEM 2  
% How do you find out if someone is the descendant of someone  
else ? الحفيد او السليل عكس
```

```
descendant(A, B):-  
    parent(B, A). % A is B's descendant if  
                  % B is A's parent.
```

```
descendant(A, B):-  
    parent(B, P),  
    descendant(A, P). % A is B's descendant if  
                    % B is a parent of some person P and  
                    % A is P's descendant.
```

```
% PROBLEM 3
```

% How do you know if someone is related to someone else ?

```
related(A, B):- ancestor(A, B). % A is related to B if % A is B's ancestor.

related(A, B):- descendant(A, B). % A is related to B if % A is B's descendant.

related(A, B):- ancestor(X, A), ancestor(X, B), A \== B. % A is related to B if % some person X is A's ancestor and % X is also B's ancestor and % A is not the same as B.
```

% There are lots of other different ways of solving this problem.

% PROBLEM 4
% How can you decide whether two people could possibly marry, given that
% only an unrelated male and female are allowed to do this ?

possible_to_marry(A, B):- ?

واجب

```
possible_to_marry(A, B):- male(A), female(B), \+ related(A, B). % It is possible for A to marry B if % A is male and % B is female and % A is not related to B.
```

\+ means "not provable" (e.g. **\+ false** is always true).

ملاحظات

```
% FOOTNOTE
% The symbol \== means "not the same as" (e.g. foo \== baz is true).
% The symbol \+ means "not provable" (e.g. \+ false is always true).
% The symbol _ represents an "anonymous variable" (i.e. a variable
% for which there is no need for a particular name).
% Comments are on lines beginning with %
% or in sections bounded by /* and */
```

Example 1:

The symbol	means
\neq	not the same as ليس نفس
\nmid	not provable لا يمكن إثباته \nmid false is always true
$\text{read}(_ ,_)$	"anonymous variable" (i.e. a variable for which there is no need for a particular name). مثلا X

أكبر عدد بين اثنين

domains

$x = \text{integer}$

predicates

$\text{max}(x, x, x)$

clauses

$\text{max}(X, Y, \text{Max}) :- X > Y, \text{Max} = X.$

$\text{max}(X, Y, \text{Max}) :- \text{Max} = Y.$

Output :

Goal: $\text{max}(2, 3, \text{Max})$

Max=3

1 Solution

Goal: $\max(-1, -4, \text{Max})$

Max=-1

1 Solution

جمع عددين

predicates

sum.


Clauses


```
sum:-      write("inter X value "),readreal(X) ,
           write("inter Y= value "),readreal(Y) ,
           S = X+Y,
           write("Sum is = "),write(S).
```

Example:


برنامج عمل الروبوت وكيف يمكن ان يرفع اشكال معينه فوق بعض.

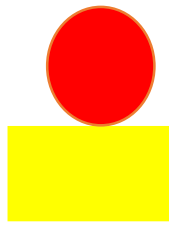
الاشكال

Circle ... C 

Rectangle ... R 

Triangle T 

Square S 



Can_be(C,R).

Can_be(C,S).

Can_be(R,S).

Can_be(T,R).

Can_be(T,S).

Can_be(S,R).

Can_not_be(R,C).

Can_not_be(T,C).

Can_not_be(S,C).

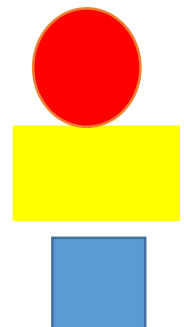
Can_not_be(C,T).

Can_not_be(R,T).

Can_not_be(S,T).

Robot_can_pickup1(X,Y):- Can_be(X,Y),!.

Robot_can_pickup2(X,Y,Z):- Can_be(X,Y), Can_be(Y,Z).



Fill the blanks

- Tom is John's child → `child(tom,)`
- Ann is Tom's child → `child(....., tom)`
- John is Mark's child → `.....(john,)`
- Alice is John's child → `.....`

- Is Ann a child of Tom? → `child(ann, tom)`
- Who is a grandchild of Ann? → `grandchild(X, ann)`
- Whose grandchild is Tom? → `?`
- Who is a grandchild of whom? → `?`

`child(ann, tom)`

- Since there are no variables in the first goal the answer is simply

yes

`grandchild(X, ann)`

- Since the program contains no information about grandchildren of Ann the answer to the second goal is

No

`grandchild(tom, X)`

- Since Tom is the grandchild of Mark the answer is

X = mark

`grandchild(X, Y)`

- The final goal yields three answers:

*X = tom Y = mark
X = alice Y = mark
X = ann Y = john
child(alice, john)*

Q: write a prolog program to define the following idea

• If someone want to buy a car how to represent this relationship, use the following facts :

- Kelly is a person
- Judy is a person
- Mercedes is a car
- Benz is a car
- Kelly like B enz
- Judy likes pizza
- pizza for sale
- Mercedes for sale
- Benz for sale

Domains

X=symbol

predicates

can_buy(X, X)

person(X)

```

car(X)
likes(X, X)
for_sale(X)
clauses
can_buy(X, Y) :- person(X), car(Y), likes(X, Y), for_sale(Y).
person(kelly).
person(judy).
car(mercedes).
car(benz).
likes(kelly, benz).
likes(judy, pizza).
for_sale(pizza).
for_sale(mercedes).
for_sale(benz).

```

من الامتحانات

- Q1: write a complete Prolog program contains the following facts
{ dgree_exam1(alex, 59), dgree_exam1(sara, 91), dgree_exam1(john, 42), }

predicates

```
dgree_exam1(symbol,integer)
```

clauses

```
dgree_exam1(alex, 59).
```

```
dgree_exam1(sara, 91).
```

```
dgree_exam1(john, 42).
```

- 1: convert the following sentence to prolog language " Anyone like to read magazine "
- ```
Like_to_read(X,magazen).
```
- 2: convert the following sentence from prolog language to human language { grandchild("Alexendar" , ann) }
  - Alexendar is a grandchild of ann
- 3: possible\_to\_marry(A, B):- male(A), female(B), \+ related(A, B).
- this rule is >>>
  - Correct
  - Incorrect