كلية العلوم

القسم : الرياضيات

السنة : الرابعة

المادة : ذكاء صنعي

المحاضرة : الخامسة /عملي/

{{ مكتبة A to Z }}

Facebook Group : مكتبة A to Z

يمكنكم طلب المحاضرات برسالة نصية (SMS) أو عبر (What's app-Telegram) على الرقم 0931497960

عملي ذكاء صنعي

الجلسة الخامسة

م.ريم بصل

م.ريم رقبو

القوائم Lists

تعريف القائمة:
هي عبارة عن مجموعة عناصر يمكن ان تكون هذه العناصر من أي نوع من الانواع المعرفة في Type بلغة Prolog.

ويمكن ان تكون في نفس المحموعة انواع مختلفة من العناصر فقد تحتوي المجموعة على اعداد صحيحة واعداد حقيقية وثوابت وحقائق ومجموعات اخرى وغيره.... .

يفصل بين عناصر المجموعة فاصلة عادية وتبدأ المجموعة بقوس مربع وتنتهي بقوس مربع [ ].

[apple , orange , banana]
[[1 , blue , 3] , [1,45] , [tomato]]

التعامل مع القوائم في Prolog:
1- يعبر عن القائمة الخالية ب [ ]
2- يعبر عن القائمة غير الخالية بالشكل [Head|Tail]

ال Head هو اول عنصر في المجموعة.
ال Tail هو عبارة عن قائمة تحوي بقية عناصر المجموعة ماعدا العنصر الاول.
[123 | [456 , 1011]] is the same as: [123 , 456 , 1011].
[[1] | [[a , 8] , c , [d]]] is the same as: [[1] , [a , 8] , c , [d]].

Error:
[123 | 456 , 1011].

مثال:
لو كتبنا الاستفسار التالي في شاشة التنفيذ:

1- [H | T] = [one , two , three , four].

يكون الخرج :

H = one ,
T = [two , three , four].

2- [H | T] = [a].

يكون الخرج :

H = a ,
T =[ ].

3- [H | T] = [ ].

يكون الخرج :

False.

4- [X , Y , Z | W] = [a , b , c , d , e].

يكون الخرج :

X = a ,
Y = b ,
Z = c ,
W = [d , e].

مثال2: لتكن لدينا الحقيقة التالية

lists1([ali , ahmad , basel]).

لو نفذنا:

1- lists1([H | T]).

يكون الخرج:

H = ali ,
T = [ahmad , basel].

2- lists1([_ , Y | [_]]).

يكون الخرج:

Y = ahmad.

3- lists1([_ | [Y | [_]]]).

يكون الخرج:

Y = ahmad.

لايجاد العنصر الأول نستخدم الحقيقة التالية:

```
head([H|_] , H).
```

لايجاد الذيل(بقية عناصر القائمة) نستخدم الحقيقة التالية:

```
tail([_|T] , T).
```

اكتب برنامج لحساب طول قائمة :

```
len ( [] , 0 ).
len( [H | T] , L ):- len( T , X ) , L is X+1.
```

لو استدعينا :

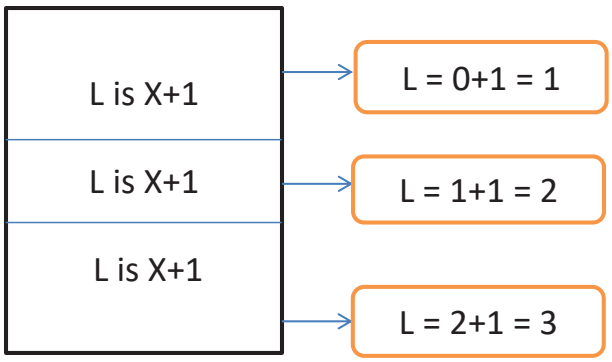
```
len([1,2,3] , L).
R1: fail.
R2: len( [ 2 , 3 ] , X) , L is X+1.

Call len ( [2,3 ] , X).
R1: fail.
R2: len( [3] , X) , L is X+1.

Call len( [3] , X).
R1: fail.
R2: len( [ ] , X) , L is X+1.

Call len ( [ ] , X).
R1: X = 0.
```

نعوض قيمة X في المكدس :



اذا L = 3.

اكتب برنامج لحذف عنصر من القائمة لمرة واحدة فقط:

```
del( X , [X | T] , T) :-!.
del(X , [H | T] , [H | L]):- del(X , T , L).
```

لو استدعينا:

```
del( 3 , [4,3,2,3] , L).
```

R1: fail.
R2: X=3 H =4 T = [3,2,3] [H | L]=[4|L] del(3 , [3,2,3] , L).

Call del(3 , [3,2,3] , L).
R1: X=3 T=[2,3]

اذا : L = [4,2,3]

اكتب قاعدة المعرفة للبحث عن عنصر ضمن القائمة:

```
search( X , [X |_ ] ).
search( X , [H| T] ) :- search( X , T ).
```

لو استدعينا:

```
search( 1 , [ 2 , 1 , 3 ] ).
```

R1: fail.
R2: X=1 H=2 T= [1,3] search(1 , [1,3]).

Call search(1 , [1,3]).
R1: X=1 [X | _] = [1 | _] True

اكتب برنامج لمعرفة اذا كانت قائمة هي جزء من قائمة اخرى

```
subset( [ H|T ] , B):- search( H , B ) , write('H= ') , writeln(H) , subset( T , B).
subset( [ ] , S).
```

لو استدعينا:

```
subset([2,1,3,5] , [1,2,3,4,5]).
R1: H=2 T=[1,3,5] B=[1,2,3,4,5]  search(2,[1,2,3,4,5])  H=2
subset([1,3,5] , [1,2,3,4,5])

 call  subset([1,3,5] , [1,2,3,4,5])
R1: H=1  T=[3,5]  B=[1,2,3,4,5]  search(1,[1,2,3,4,5])  H=1  subset([3,5] , [1,2,3,4,5])

 call  subset([3,5] , [1,2,3,4,5])
R1:  H=3  T=[5]  B=[1,2,3,4,5]  search(3,[1,2,3,4,5])  H=3  subset([5] , [1,2,3,4,5])
```

```
Call   subset([5] , [1,2,3,4,5])
R1:  H=5  T=[]  B=[1,2,3,4,5]  search(5,[1,2,3,4,5])  H=5  subset([] , [1,2,3,4,5])

 call   subset([] , [1,2,3,4,5]
R2: H= 2
     H= 1
     H= 3
     H= 5
     true.
```

اكتب برنامج لجمع عناصر قائمة:

```
sum([] , 0).
sum([ H|T ] , S):- sum(T,S1) , S is S1+H.
```

لو استدعينا:

sum([3,2,1],S).
R1:fail
R2: H=3 T=[2,1] sum([2,1] , S1) , S is S1+3

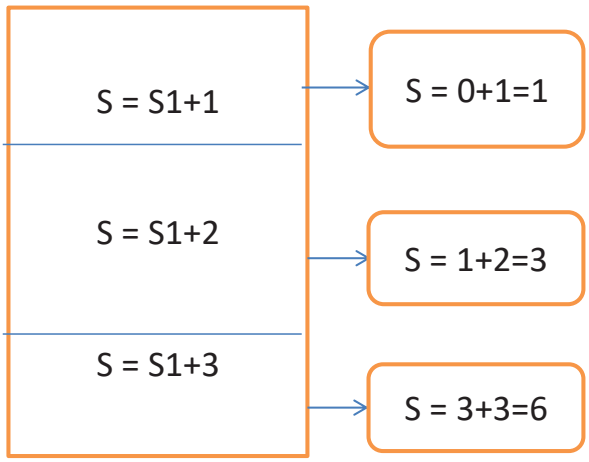
Call sum([2,1] , S1)
R1:fail
R2: H=2 T=[1] sum([1] , S1) , S is S1+2

Call sum([1] , S1)
R1: fail
R2: H=1 T=[ ] sum([ ] , S1) , S is S1+1

Call sum([ ] , S1)
R1: S1=0

نعوض قيمة S1 في المكدس

اذا S = 6



وظيفة:

١- اكتب برنامج يقوم بحذف عنصر من القائمة مع تكراراته.

٢- اكتب برنامج يقوم بحساب المتوسط الحسابي لعناصر القائمة.

مكتبة A to Z