Assignment 3 – Prolog

Student Name: Nicole Colgan

Student ID: 19345826

Graphical user interface, text, application, email

Description automatically generated

1.

Format:

Teaches(mary,bob) :-

Takes(bob,ct331),

Teaches(mary,ct331).

# Code

teaches(X,Y) :-

takes(Y,Z),

instructs(X,Z).

# output

Text

Description automatically generated with medium confidence

2.

Teaches(bob,X).

A query is just a question you ask the script so we don’t need to edit the script in this case, were just asking a question

When querying, un-instantiated variables have to start with a capital letter

For some reason only tom is showing up

Text, letter

Description automatically generated

3.

?- teaches(X,mary).

Text

Description automatically generated

4.

Text

Description automatically generated

False because first is checks WHAT Ann instructs => result is ct331 so ct331 is assigned to Z.

Then it checks if Joe tajes ct345 and he doesn’t so since not both of the expressions are true, the head is not true.

* Answer = false

5.

classmates(STUDENTONE,STUDENTTWO):-

takes(STUDENTONE,Z),

takes(STUDENTTWO,Z).

Text

Description automatically generated

Above, it first checks what tom takes and it sees he takes ct331, then it assigns ct331 to Z, then it checks if mary takes ct331 and she does so the head is true

* Result = true

Text

Description automatically generated

Above, it checks to see what bob takes but bob is a teacher so he doesn’t take anything so since not both expressions are true, the head is false

* Result = false. Graphical user interface, text, application

  Description automatically generated
* Unification is when the head of a clause matched a goal
* If they have the same function name (functor) and arguments e.g. rainy(X) = rainy(seattle)
* The argument is bound do that variavle

1. Displayhead and tail of [1,2,3]

unify head with first element and tail with other elements

?- [H|T]=[1,2,3].

H = 1,

T = [2, 3].

2.

?- [H|[H2|T2]]=[1,2,3,4,5].

H = 1,

H2 = 2,

T2 = [3, 4, 5].

3.

Use “=:=” for variable equality check

Don’t need to put in “T” because we don’t use it in the rule

contains1(X,[H|\_]):-

X=:=H.

Text, letter

Description automatically generated

4.

Have to use unification (“=”) for lists

contains2(X,[\_|T]):-

X=T.

Text

Description automatically generated

contains2(X,[H|T]):-

X=T.

5.

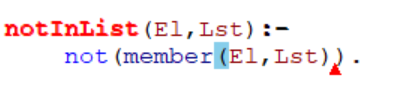
Can unify elements again but don’t need to worry about what is entered in the first case because were not using them

contains1(\_,[X|\_])=contains1(1,[1,2,3]).

X = 1. Graphical user interface, text, application

Description automatically generated

In hind sight, I could have done it like this



%if el = H, false

isNotElementInList(El,[El|T]):-

!, %discard all choice points created.

false.

%if El != H, keep traversing.

isNotElementInList(El,[H|T]):-

!,

isNotElementInList(El,T).

%if list is empty, its not in list - true

isNotElementInList(EL,[]).

Note: “!” discards all choice points since entering the predicate so commit to the clause where the cut appears.

Explanation of outputs:

Since list is empty, corresponding fact is called which is true.

Text

Description automatically generated

Rule isNotElementInList(El,[El|T]) is called because the element 1 == the head of the list (1)

Then all other choices are discard and its false.

Text

Description automatically generated

isNotElementInList(El,[H|T]) is called because the element 1 is not equal to the head of the list (2)

now the fact isNotElementInList(EL,[]). Is checked because the tail of the list is empty. This results in true.

Text

Description automatically generated

isNotElementInList(El,[H|T]) is called because the element 2 is not equal to the head of the list (1)

Rule isNotElementInList(El,[El|T]) is called because the element 2 == the head of the list (2)

Then all other choices are discard and its false.

Text

Description automatically generated

isNotElementInList(El,[H|T]) is called because the element 7 is not equal to the head of the list (1)

this continues until we get an empty list which chcecks the fact isNotElementInList(EL,[]). This results in true.

Text

Description automatically generated with low confidence

Graphical user interface, text, application

Description automatically generated

mergeLists(L1,L2,L3,Merged):-

conc(L1,L2,L1L2),%Merge L1 & L2

conc(L1L2,L3,Merged).%final case

conc([],X,X). %base case

conc([H|T],X,[H|Z]):-%keep adding to final

conc(T,X,Z).

Text, letter

Description automatically generated

Explanation

mergeLists(L1,L2,L3,Merged)is called and the first predicate conc(L1,L2,L1L2) is called

conc(T,X,Z). is called with the tail of list 1(2), all of list 2 (3,4) and some unknown value “?”

conc([H|T],X,[H|Z]) is called again and conc([],[3,4],?) is called

This gives us the base case conc([],X,X) so now “?” is [3,4]

We exit back to conc(T,X,Z) which is now conc([],[3,4],[3,4]) then exit to conc([H|T],X,[H|Z]) which will now be conc([2],[3,4],[2,3,4]).

This continues until we get back to the original conc and conc(L1L2,L3,Merged) is called.

This is now conc([1,2,3,4], [5,6], Merged)

We do the same as for the concatenation of L1 and L2 until we get a final answer for merged Text

Description automatically generated Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

%keep calling function till we get to end

%then appendreverse([],Z,Z).

reverseList(L,X):-

reverse(L,X,[]).

reverse([H|T],Final,Empty) :-

reverse(T,Final,[H|Empty]).

reverse([],Z,Z).

# Explanation of reverseList([1,2,3],X).

Firstly “reverseList(L,X)” is called as reverseList([1,2,3],X).

The predicate for this is “reverse(L,X,[]).” So “reverse([1,2,3],?,[]).” Is called

Reverse is recursively called until the tail of the original list is empty.

So the next call (the predicate of reverse([H|T],Final,Empty)) is reverse(T,Final,[H|Empty]). Which in terms of this example is reverse([2,3], ?, [1|empty]).

This corrosponds to reverse([H|T],Final,Empty).

The predicate reverse(T,Final,[H|Empty]) is called as reverse([3] , ?, [2, 1]).

This corrosponds to reverse([H|T],Final,Empty).

The predicate reverse(T,Final,[H|Empty]) is called as reverse([ ] , ?, [3, 2, 1]).

This corrosponds to reverse([ ], Z , Z).

So now Z is [3,2,1].

This returns to the previous call and now reverse(T,Final,[H|Empty]). Is reverse([], [3, 2, 1], [3, 2, 1])

The exit process continuos until we get back to the orginal reverse call where X is now [3,2,1].

Final result is X= 3,2,1.

A picture containing text, receipt

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated %if searched through all list

insertInOrder(El,[],[El]).

%if El<H we stop here and final list is [newEl, old head, old tail]

insertInOrder(El,[H|T],[El,H|T]):-

El<H,

!.

%El>!H so recursive call

insertInOrder(El,[H|T],[H|New]):-

insertInOrder(El,T,New).

# Explanation for output 1

insertInOrder(El,[H|T],[El,H|T]) matches the format “insertInOrder(7,[1,2,3], ?).” so its checked first

is 7<1 no so check next Insert in order insertInOrder(El,[H|T],[H| ?])

this is now insertInOrder(7, [1 | 2, 3], [?]). Because 7 is not greater than one

The predicate insertInOrder(El,T,New). Which will be insertInOrder(7, [2,3], ?) is checked

insertInOrder(El,[H|T],[El,H|T]) matches the format “insertInOrder(7,[2,3],?).” so its checked first

is 7<2 no so check next Insert in order insertInOrder(El,[H|T],[H| ?])

this is now insertInOrder(7, [2| 3], [?]). Because 7 is not greater than one

The predicate insertInOrder(El,T,New). Which will be insertInOrder(7, [3], ?) is checked

insertInOrder(El,[H|T],[El,H|T]) matches the format “insertInOrder(7,[3],?).” so its checked first

is 7<3? No so check next Insert in order insertInOrder(El,[H|T],[H| ?])

The predicate insertInOrder(El,T,New). Which will be insertInOrder(7, [], ?) is checked

This matched the fact insertInOrder(El,[],El). So now new list is [7]

Exit and return to insertInOrder(El,T,New). Which will be insertInOrder(7, [3], [3,7]).

Continue exiting until reach initial check where new list is now [1,2,3,7]

Text, letter

Description automatically generated Text, letter

Description automatically generated