UNIVERSIDADE ESTADUAL VALE DO ACARAÚ

Centro de Ciências Exatas e Suas Tecnologias (CCET)

Ciência da Computação

**MAXWELL ALVES TEIXEIRA**

**LUCAS DE LIMA**

**Exercício Extra**

**SOBRAL - CE**

**2018**

1ª. Simplifique a cláusula abaixo:

pega3(E,L):-L=[B|A],A=[Y|Ys],Ys=[Z|D],Z=E

R:

pega3(E,[B|A]):- A=[Y|Ys],Ys=[Z|D],Z=E

pega3(E,[B|[Y|Ys]]):- Ys=[Z|D],Z=E

pega3(E,[B|[Y|[Z|D]]]) :- Z=E

pega3(E,[B|[Y|[E|D]]]).

2ª. Rode o trace para membro(X,[a,b,c])?

R:

[trace] ?- membro(X,[a,b,c]).

Call: (8) membro(\_2846, [a, b, c]) ? creep

Exit: (8) membro(\_2846, [a, b, c]) ? creep

true.

3ª. Escreva em linguagem natural as regras que definem o predicado último. Uma regra base e uma regra recursiva.

R:

ultimo([U], U).

ultimo([X|Xs], U):- ultimo(Xs, U).

4ª. Defina um predicado contiguos/1, que testa se uma lista tem dois elementos contíguos iguais. Com e sem o uso do append/3. Por exemplo ?-contiguos ([a,b,c,c,e]).

Yes.

R:

5ª. Defina um predicado dupl/1, que é verdadeiro, se a lista tem elementos duplicados.

Use o member.

R: dupl(L):- member(X,L),select(X,L,L2),member(Y,L2),X == Y,!.

6ª. Defina um predicado trocaPU/2 que devolve uma lista em que o primeiro e último elementos são trocados de posição. Use append.

7ª. Defina um predicado remDupl/2 que remove os elementos duplicados de uma lista, use apenas o member e recursividade. Remova as cópias iniciais dos elementos da lista.

Assim, ?-remDupl([a,b,a],X). X=[b,a]. Yes

8ª. Defina o predicado palindrome/1, que é verdadeiro se a lista é um palíndromo, por exemplo, [a,b,c,d,c,b,a]. Faça duas versões, uma usando reverse e uma recursiva usando append.

R:

reversao(L,R) :- reversao(L,[],R).

reversao([],R,R).

reversao([X|Xs], Acc, R):- reversao(Xs, [X|Acc], R).

palindrome(L):- reversao(L,R), R == L.

9ª. Escreva um predicado pertence3/3 em que o terceiro parâmetro é true ou false. Como segue.

?- pertence(a, [c,a,b],X). X=true Yes

R: pertence(X,L,Y):- member(X,L),Y == true.

10ª. Faça um predicado insOrd/3, que insere um elemento numa lista mantendo-a ordenada. Faça duas regras: uma base e uma recursiva. ?-insOrd(4,[2,3,5,7],L).

L=[2,3,4,5,7] Yes