

Exercícios de programação em PROLOG – 24/03/2018

Listas

1ª. Simplifique a cláusula abaixo:

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

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

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

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.

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

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.

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

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