UNIVERSIDADE FEDERAL DE UBERLÂNDIA

GABRIEL RIBEIRO BERNARDI - 11821BCC036

AARE - Avaliação Parcial da Aula 4

UBERLÂNDIA, MG

2020

Link com repositório do github para a disciplina: https://github.com/gabrielrbernardi/Prolog
Observação: Alguns comentários para maior entendimento dos predicados estão juntos com os códigos

Exercício 1

```
ehVazia(Lista) :- not(member(_,Lista)).

pulaSapo(N, [A, B|C]) :-
    X is B-A, %X recebera o valor da diferenca entre os dois proximos canos
    (   X =< N -> ( pulaSapo(N,[B|C]) -> nl; %se o valor da diferenca dos proximos canos for maior que o salto que o sapo consegue dar, o predicado retornara falso, o que indica que o sapo nao consegue pular os canos. caso contrario, prossegue com a execucao da recursao
                    ( ehVazia(C) -> true) %se chegarmos nos ultimos dois canos, o valor de C sera vazio, o que faria com que o predicado retornasse falso, o que faria com que, mesmo que os saltos sejam possiveis, a funcao retornaria falso. para isso foi criada a funcao ehVazio que verificara se C e uma lista vazia. Se isso ocorrer, indica que o sapo conseguiu fazer os pulos e chegamos no final da lista.
                  );
        false
    ).
```

Casos de teste:

Caso teste	Pergunta	Resposta
1	pulaSapo(1, [1,2,3]).	true
2	pulaSapo(1, [1,2]).	true
3	pulaSapo(1, [1,3]).	false
4	pulaSapo(2, [2,4,6,8,10]).	true
5	pulaSapo(1, [2,4,6,8,10]).	false
6	pulaSapo(1, [1,2,4]).	false
7	pulaSapo(2, [1,2,4]).	true

Exercício 2

```
ehMembro(A, [A|_]).
ehMembro(A, [_|B]) :-
    ehMembro(A, B).

somaPar([Cabeca|Rabo], C) :-
    D is C - Cabeca,
    ehMembro(D, Rabo), !; %verifica se D e membro da lista "Rabo"
    somaPar(C, Rabo).
```

Casos teste	Pergunta	Resposta
1	somaPar([1,3,4,1], 1).	false
2	somaPar([1,3,4,1], 2).	true
3	somaPar([1,3,4,1], 3).	false
4	somaPar([1,3,4,1], 4).	true
5	somaPar([1,2,3,4,1], 3).	true
6	somaPar([1,2,3,4,1], 6).	true
7	somaPar([1,2,3,4,5,6,7,8,9], 14).	true
8	somaPar([1,2,3,4,5,6,7,8,9], 18).	false

Exercício 3

Exercício 4

```
intercala([],B,B). %se primeira lista vazia, intercala com os proprios elementos de B
intercala(A,[],A). %se segunda lista vazia, intercala com os proprios elementos de A

%para os seguintes predicados temos que o algoritmo ira escolher o menor valor entre o primeiro elemento da primeira lista e o primeiro elemento da segunda lista. feito isso, esse elemento será colocado no inicio da lista obtida pela recursão
intercala([D|A],[E|B],[E|C]) :-
    (   D > E -> intercala([D|A],B,C), !).
```

```
intercala([D|A],[E|B],[D|C]) :-
    (   D =< E ->   intercala(A,[E|B],C), !).
```

Casos teste:

Caso teste	Pergunta	Resposta
1	intercala([1,2,5],[3,4,6],R).	R = [1, 2, 3, 4, 5, 6]
2	intercala([1,2,5,7,9,11],[3,4,6],R).	R = [1, 2, 3, 4, 5, 6, 7, 9, 11]
3	intercala([1,3,5],[2,4,6],R).	R = [1, 2, 3, 4, 5, 6]
4	intercala([1,3,5],[2,4,6],[1,2,3,4,5,6]).	true
5	intercala([1,3,5],[2,4,6],[1,2,3,4,5]).	false
6	intercala([9,8,7],[6,5,4],R).	R = [6, 5, 4, 9, 8, 7]