

**Ficha de Problemas da disciplina de
Algoritmia Avançada
do 3º ano da
Licenciatura em Engenharia Informática da
Instituto Superior de Engenharia do Porto**

Ficha nº 4 – Introdução às Listas

1. Considere o seguinte predicado escrito em PROLOG:

```
membro(X,[X|_]).  
membro(X,_[L]):-membro(X,L).
```

a) Faça a traçagem desse predicado quando é posta a questão:

?-membro(3, [1,2,3,4]).

b) Faça a traçagem desse predicado quando é posta a questão:

?-membro(5, [1,2,3]).

c) Faça a traçagem desse predicado quando é posta a seguinte questão e são pedidas soluções múltiplas:

?-membro(X, [1,2,3]).

d) Faça a traçagem desse predicado quando é posta a seguinte questão e são pedidas soluções múltiplas:

?-membro(3, L).

e) Que consequências teria a substituição da primeira linha do predicado por:

`membro(X,[X|_]):-!`

A substituição da primeira linha pelo predicado aqui patente (com o cut) alterou a apresentação de resultados, sendo que limitou a selecção de possíveis elementos de uma lista, bem como a presença de um dado elemento numa única lista.

f) Que consequências teria a troca das duas linhas do predicado, ou seja:

```
membro(X,_[L]):-membro(X,L).  
membro(X,[X|_]).
```

A substituição da primeira linha pelo predicado aqui patente (invertido) passa a dar mais prioridade à cauda de uma lista, no entanto, a sua eficiência é menos eficiente, como por exemplo, com `membro(1,[1,2,3,4,5,6])`, ele terá de visitar todos os elementos da lista até ocorrer uma falha e voltar atrás. No retrocesso, aí verifica quais são os elementos que estão à cabeça, falhando sempre até chegar ao 1º Elemento, quando ocorre sucesso.

Na execução da instrução `membro(3,L)` conduz a chamadas recursivas infinitas, causando a stack “estourar”.

2. Considere o seguinte predicado em PROLOG

```
concatena([ ],L,L).  
concatena([A|B],C,[A|D]):-concatena(B,C,D).
```

- a) Faça a traçagem desse predicado quando é posta a questão:
?-concatena([1,2,3], [4,5],L).
- b) Relativamente à traçagem anterior indique como foi obtida a lista resultante, ou seja, L.
- c) Faça a traçagem desse predicado quando é posta a seguinte questão e são pedidas soluções múltiplas:
?-concatena(L1,L2,[1,2,3]).
- d) Que consequências teria a substituição da primeira linha do predicado por:

```
concatena([ ],L,L):-!.
```

- e) Que consequências teria a troca das duas linhas do predicado, ou seja:

```
concatena([A|B],C,[A|D]):-concatena(B,C,D).  
concatena([ ],L,L).
```

3. Implemente predicados em PROLOG que efectuem as seguintes tarefas:

- a) Contar os elementos de uma lista
- b) Somar os elementos de uma lista de números
- c) Obter a média dos elementos de uma lista de números
- d) Obter o menor valor de entre os elementos de uma lista de números
- e) Contar o nº de elementos da lista que se encontram dentro de uma determinada de valores (supõe-se a lista composta unicamente por valores numéricos)

4. Implemente um predicado *lista(L)* que quando for chamado terá sucesso se L for uma lista.

Exercícios de Consolidação

- 5) Construa um predicado *repetidos(L)* que verifica se uma lista tem ou não elementos repetidos.
- 6) Crie um predicado *prodvec(L1,L2,LV)* que multiplique dois vectores contidos em L1 e L2, retornando o resultado em LV.