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 5 – Listas

1. Escreva um predicado em PROLOG para eliminar elementos de uma lista

a) O predicado deverá eliminar apenas a primeira ocorrência de um elemento na lista, sendo exemplificado como se segue:

?- elimina1(2, [1,2,3,2,4,2],L).
L=[1,3,2,4,2]
yes

b) O predicado deverá eliminar todas as ocorrências de um elemento na lista, sendo exemplificado como se segue:

?- elimina_todos(2, [1,2,3,2,4,2],L).
L=[1,3,4]
yes

2. Teste o seguinte predicado, pedindo soluções múltiplas. Indique o seu objectivo.

```
p([ ],[ ]).
p(L,[X|L1]):-elimina1(X,L,Li),p(Li,L1).
```

Apresenta todas as listas possíveis de serem construídas com uma dada lista passada como parâmetro, através da eliminação sucessiva de elementos.

3. Escreva um predicado em PROLOG para substituir todas as ocorrências de um dado elemento numa lista por outro elemento.

?- substitui(2,6, [1,2,3,2,4,2],L).
L=[1,6,3,6,4,6]
yes

4. Escreva um predicado em PROLOG para inserir um elemento numa dada posição numa lista.

?- insere(7,4, [1,2,3,2,4,2],L).
L=[1,2,3,7,2,4,2]
yes

5. Considere o seguinte predicado em PROLOG que faz a inversão de uma lista originando outra.

```
inverte(L,LI):- inverte1(L,[ ],LI).

inverte1([ ],L,L).
inverte1([X|L],LI,LF):- inverte1(L,[X|LI],LF).
```

a) Faça a traçagem desse predicado quando é posta a questão:
?-inverte([1,2,3], LI).

b) Qual a função do 2º parâmetro do inverte1?

6. Considere o predicado união escrito em PROLOG que efectua a união de dois conjuntos representados por listas (dentro de cada lista não há elementos repetidos):

```
união([ ],L,L).
união([X|L],L1,L2):- member(X,L1),!,união(L,L1,L2).
união([X|L],L1,[X|L2]):- união(L,L1,L2).
```

a) Faça a traçagem desse predicado quando é posta a questão:
?-união([1,2,3],[2,4,6], LI).

b) O que ocorreria se fosse retirado o “!” ?

c) Implemente um predicado que faça a intersecção entre 2 conjuntos.

d) Implemente um predicado que faça o conjunto diferença entre 2 conjuntos, ou seja, que seja capaz de gerar um conjunto com os elementos que pertencem a um dos dois conjuntos, mas não a ambos.

7. Escreva um predicado em PROLOG que dada uma lista de elementos representando uma aposta de totobola, com múltiplas representadas por listas internas, seja capaz de gerar as apostas simples correspondentes.

Por exemplo:
?- totobola ([1,x,2,1,x,2,1,x,2,1,x,[1,x],[1,x,2]],LS).
LS=[1,x,2,1,x,2,1,x,2,1,x,1,1];
LS=[1,x,2,1,x,2,1,x,2,1,x,1,x];
LS=[1,x,2,1,x,2,1,x,2,1,x,1,2];
LS=[1,x,2,1,x,2,1,x,2,1,x,x,1];
LS=[1,x,2,1,x,2,1,x,2,1,x,x,x];
LS=[1,x,2,1,x,2,1,x,2,1,x,x,2];
no

8. Escreva um predicado em PROLOG que dada uma lista de elementos representando uma aposta de totoloto, que poderá ter mais de 6 números, seja capaz de gerar as apostas simples correspondentes.

Por exemplo:
?- totoloto ([1,2,3,4,5,6,7],LS).
LS=[1,2,3,4,5,6];
LS=[1,2,3,4,5,7];
LS=[1,2,3,4,6,7];
LS=[1,2,3,5,6,7];
LS=[1,2,4,5,6,7];
LS=[1,3,4,5,6,7];
LS=[2,3,4,5,6,7];
no

9. Considere termos do tipo t(Di,Df) que representam os intervalos de datas disponíveis para uma equipa de futebol realizar jogos num dado mês. Esses intervalos são agrupados em listas ordenadas. Por exemplo, uma dada equipa poderia ter a seguinte lista de datas livres:

[t(2,7), t(12,14), t(19,19), t(25,28)]

Escreva um predicado que recebendo duas dessas listas, correspondentes a duas equipas, seja capaz de gerar outra lista com as datas possíveis dos jogos. Essa última lista também deverá vir no mesmo formato.

Por exemplo:
?- datas_jogo([t(2,7), t(12,14), t(19,19), t(25,28)], [t(1,3), t(6,13), t(17,21), t(26,27)],L).
L=[t(2,3), t(6,7), t(12,13), t(19,19), t(26,27)]
yes

10. O problema das 8 rainhas consiste em colocar 8 rainhas num tabuleiro de xadrez vazio de 8x8 posições tal que não seja possível que uma rainha ataque qualquer outra. Uma rainha ataca uma outra se esta estiver na mesma linha, coluna ou diagonal.

O tabuleiro pode ser representado por uma lista de 8 elementos em que cada elemento da lista representa a posição de uma das rainhas no tabuleiro. Uma possível solução para o problema é a seguinte:

[(1,4),(2,2),(3,7),(4,3),(5,6),(6,8),(7,5),(8,1)]

Esta lista representa o tabuleiro:

8						●		
7			●					
6					●			
5							●	
4	●							
3				●				
2		●						
1								●
	1	2	3	4	5	6	7	8

Deve desenvolver um predicado que permita obter todas as soluções para o problema das 8 rainhas.

Uma vez que as rainhas terão sempre que estar em colunas diferentes para não se atacarem, poderá partir de uma posição inicial:

pos_inicial([(1,Y1),(2,Y2),(3,Y3),(4,Y4),(5,Y5),(6,Y6),(7,Y7),(8,Y8)]).

O predicado que devolve a solução pode utilizar a lista com as posições iniciais e escolher os valores adequados para cada um dos Y obtendo uma solução válida.

Por exemplo:

?- rainhas8(P).
P = [(1,4),(2,2),(3,7),(4,3),(5,6),(6,8),(7,5),(8,1)]

11. Dado um conjunto de *n* homens e *n* mulheres, o objectivo deste exercício é juntá-los em pares após cada homem ter dado a lista de mulheres que prefere de *1* até *n* $[m_1,\ldots,m_n]$ e cada mulher ter feito o mesmo para os homens $[h_1,\ldots h_n]$.

A solução será uma lista de pares $[h_i,m_j]$ e é considerada aceitável se os casamentos forem estáveis isto é, não existirem dois pares $[h_i,m_j]$, $[h_k,m_l]$ tal que h_i prefere m_l em vez de m_j e m_l prefere h_i em vez de h_k.

A teoria de grafos demonstra que este problema tem sempre solução e adaptações da solução deste problema são utilizadas para a colocação de candidatos em lugares quando estes manifestam uma ordem de preferência.

Dadas as seguintes listas:

Homens = [a,b,c,d,e] Mulheres = [p,q,r,s,t]

Poderá utilizar a seguinte base de conhecimento:

```
preferencias(a, [q,t,p,r,s]).
preferencias(b, [p,q,r,s,t]).
preferencias(c, [q,r,t,s,p]).
preferencias(d, [p,r,q,s,t]).
preferencias(e, [t,r,q,p,s]).
preferencias(p, [e,a,d,b,c]).
preferencias(q, [d,e,b,a,c]).
preferencias(r, [a,d,b,c,e]).
preferencias(s, [c,b,d,a,e]).
preferencias(t, [d,b,c,e,a]).
```

Deverá implementar o seguinte predicado:

casamentos_estaveis(Homens, Mulheres, Casamentos).

Para a base de conhecimento anterior, a chamada ao predicado implementado deverá resultar no seguinte:

```
?- casamentos_estaveis([a,b,c,d,e], [p,q,r,s,t], C).
C = [m(a,p),m(b,s),m(c,t),m(d,r),m(e,q)]
```