

Resolução de Problemas em Prolog

¹ Marcelo Ruan Moura Araújo e ² André Ribeiro Brito

^{1,2} Universidade Federal de São Carlos - UFSCar

^{1,2} Departamento de Computação

¹ Biomedical Image Processing Group - BIP Group

marcelo.araujo@ufscar.br e andrebrito@ufscar.br

1. Problemas

1. Dada uma lista *Lin* que contém elementos de qualquer tipo, possivelmente com repetições, construir outra lista *Lout* que mostre quantas vezes cada elemento atômico (átomo, número, lista vazia) aparece na lista dada, inclusive nas sublistas. A lista *Lout* deve conter pares de elementos sendo o primeiro elemento do par um elemento atômico que aparece na lista dada e o segundo elemento do par, o número de vezes que esse elemento aparece na lista. Variáveis e estruturas devem ser descartadas e não serão usadas na contagem.

Por exemplo, dada a lista:

```
Lin = [a,b,Z,x,4.6,[a,x],[],[5,z,x],[]]
```

Deve ser construída a lista

```
Lout=[ [a,2], [b,1], [4.6,1], [x,3], [ [],2], [5,1], [z,1] ]
```

2. Dada uma lista *Lin* com elementos de qualquer tipo, construir a codificação *Lout* dessa lista em que repetições consecutivas de elementos devem ser substituídas por pares da forma [N,E], onde N é o número de repetições consecutivas do elemento E.

```
Lin = [a,a,a,a,b,c,c,a,a,d,e,e,e,e]
```

Deve ser construída a lista

```
Lout = [ [4,a], [1,b], [2,c], [2,a], [1,d], [4,e] ]
```

3. Dada uma lista *Lin* que seja a codificação de outra lista no formato do exercício anterior, construir a decodificação dessa lista codificada.

Por exemplo, dada a lista:

```
Lin = [ [4,a], [1,b], [2,c], [2,a], [1,d], [4,e] ]
```

Deve ser construída a lista:

```
Lout = [a,a,a,a,b,c,c,a,a,d,e,e,e,e]
```

2. Soluções

2.1. Questão 1

transforma(L1 ,L):- a função transforma, tem como objetivo transformar a lista L1 onde contém sub listas em uma lista de primeiro nível, facilitando os próximos passos.

conta_ocorr(_,L,N):-! a função conta_ocorr, tem como objetivo contar as ocorrências dos elementos em uma lista no primeiro nível. Sendo passado um elemento “_”, L se referindo a lista e o N o numero de repetição.

A função **del_todas(_,L1,L2)** e **retira_elem(L2,L3)**, tem como objetivo de retirar todos os elementos repetidos em uma lista.

cria_lista(L3,L4):- a função cria lista, recebe uma lista com elementos repetido ou não, lista vazia e sublistas com elementos. Sendo a lista L4 contendo **[[X,N]|L]**, onde contém os elementos e o numero de repetições do elemento.

```
transforma([H],X) :- is_list(H),transforma(H,X).
transforma([H],[H]):-!.
transforma([H|T],X) :- is_list(H),transforma(H,Y), transforma(T,Z), append(Y,Z,X),!.
transforma([H|T],[H|X]):- transforma(T,X).
```

```
conta_ocorr(_,[],0):-!.
conta_ocorr(E,[X|Y],S):- E==X, conta_ocorr(E,Y,S1), S is S1 + 1, !.
conta_ocorr(E,[_|Z], S):- conta_ocorr(E,Z,S).
```

```
del_todas(_,[_],[_]).
del_todas(Elem, [Elem|Y], Z) :- del_todas(Elem,Y,Z).
del_todas(Elem,[Elem1|Y], [Elem1|Z]) :- Elem \== Elem1,del_todas(Elem,Y,Z).
```

```
retira_elem([], []).:-!.
retira_elem([E|Y],[E|X]):- del_todas(E,Y,Lista),retira_elem(Lista,X).
```

```
cria_lista([],[]).
cria_lista([X|Y],[[X,N]|L]):- transforma([X|Y],Y),conta_ocorr(X,Y,N1), N is N1 + 1,
retira_elem([X|Y],Z), cria_lista(Z,L).
```

```
?- [cria_lista].
true.
?- cria_lista([a,b,x,4.6,[a,x],[],[5,z,x],[[]],A).
A = [[a, 2], [b, 1], [x, 3], [4.6, 1], [[], 2], [5, 1], [z, 1]].

?- cria_lista([a,a,b,d,w,q,[a,a,v,5.8],[[]],A).
A = [[a, 4], [b, 1], [d, 1], [w, 1], [q, 1], [v, 1], [5.8, 1], [[]|...]].
```

2.2. Questão 2

junta_rep (L1, L2):- a lista L2 é obtida a partir da lista L1, agrupando ocorrências repetidas de elementos em sub-listas separadas.

empacota (X, Xs, Ys, Z):- Ys é a lista que permanece da lista Xs quando todas as cópias iniciais de X são removidas e transferidas para Z.

conta_rep (L, L2):- é chamado no predicado **cria_lista (L1, L2)**, de forma que irá verificar o tamanho dessas sublistas.

cria_lista(L1, L2):- tem o papel de chamar os dois predicados, **junta_rep** e **conta_rep**, sendo o primeiro irá agrupar as repetições em sublistas e o segundo irá verificar o tamanho dessas sub-listas, sendo esses valores inseridos nesse predicado.

Abaixo, com fundo azul, está o código utilizado, seguido da saída do programa para a mesma entrada sugerida na lista.

```
junta_rep([],[]).
junta_rep([X|Xs],[Z|Zs]) :- empacota(X,Xs,Ys,Z), junta_rep(Ys,Zs).

empacota(X,[],[],[X]).
empacota(X,[Y|Ys],[Y|Ys],[X]) :- X \= Y.
empacota(X,[X|Xs],Ys,[X|Zs]) :- empacota(X,Xs,Ys,Zs).

cria_lista(L1,L2) :- junta_rep(L1,L), conta_rep(L,L2).

conta_rep([],[]).
conta_rep([X|Xs],[[N,X]|Zs]) :- length([X|Xs],N), conta_rep(Xs,Zs).

?- cria_lista([a,a,a,a,b,c,c,a,a,d,e,e,e,e], L).
L = [[4, a], [1, b], [2, c], [2, a], [1, d], [4, e]] .

?- cria_lista([a,a,a,a,a,b,c,c,q,q,d,a],A).
A = [[5, a], [1, b], [2, c], [2, q], [1, d], [1, a]]
```

2.3. Questão 3

O exercício pede, dada uma lista como entrada Llin([N,Elem]), como exemplo Llin=([4,a],[1,b],[2,c],[1,d],[4,e])), onde contém N se referindo a quantidade de repetição do elemento e Elem os elementos que contém na lista.

O código trata-se de :

exercicio3([],[]). -> verifica se a lista é vazia, no caso, seria o caso base.

exercicio3([1,X|Ys],[X|Zs]):- exercicio3(Ys,Zs),!. -> dado o elemento, se ele repetir somente 1 vez, ele já entra na lista de saída, no caso, o primeiro elemento da lista de entrada , será o primeiro elemento de saída. Assim, chamando a cauda da lista.

exercicio3([N,X|Ys],[X|Zs]) -> no caso o N se refere a quantidade de repetição do elemento na lista, [X|Ys] é X a cabeça e Ys cauda da lista, assim a lista retorna X sendo o primeiro elemento da lista e Zs a cauda da lista. Assim o próximo passo é

decrementar a quantidade de repetição do elemento na lista, colocando a cada passo de decremento na lista de saída.

```
exercicio3([],[]).
exercicio3([[1,X]|Ys],[X|Zs]):- exercicio3(Ys,Zs),!.
exercicio3([[N,X]|Ys],[X|Zs]):- N>1,N1 is N-1,exercicio3([[N1,X]|Ys],Zs).

?- [exercicio3].
true.

?- exercicio3([[4,a],[1,b],[2,c],[1,d],[4,e]],A).
A = [a, a, a, a, b, c, c, d, e|...].

?- exercicio3([[1,a],[3,m],[1,q],[2,4.5],[1,[]]],A).
A = [a, m, m, m, q, 4.5, 4.5, []].
```