

Universidade Federal de São Carlos

Relatório do Trabalho 1

Paradigmas de Linguagens de Programação

Nome: Julia de Moura Caetano – R.A.: 619655 – Turma A
Profª Drª Heloísa de Arruda Camargo
São Carlos, 29 de Junho de 2017

1. Código-fonte

```
/*Concatenação*/
conc([ ], L, L).
conc([X|L1], L2, [X|L3]) :- conc(L1, L2, L3).

/*Apaga todas as ocorrências de um elemento em uma lista*/
rem_todas(_, [], []).:-!.
rem_todas(X, [X|L], L1):- rem_todas(X,L,L1),!.
rem_todas(X, [Y|L], [Y|L1]):- rem_todas(X,L,L1).

/*Desparentizar elementos de uma lista*/
desparentize([], []).:-!.
desparentize([X|Y], [X|Z]) :- atomic(X), desparentize(Y,Z), !.
desparentize([X|Y], Z) :- desparentize(X, X1), desparentize(Y, Y1),
conc(X1, Y1,Z).

/*Intersecção*/
inter([ ], _, [ ]):- !.
inter([X|Y], Z, [X|W]) :- member(X,Z), inter(Y,Z,W), !.
inter([X|Y], Z, W):- inter(Y,Z,W).

/*Cria uma lista só com repetições*/
tira_ao_comuns([],[],[]) :- !.
tira_ao_comuns(L1,L2,Lout) :- inter(L1,L2,La), inter(L2,L1,Lb),
conc(La,Lb,Lout).

/*Conta as ocorrências de um elemento em uma lista*/
conta(Elem, [], 1) :- !.
conta(Elem, [X|Y], N) :- X=Elem, conta(Elem, Y, Ne) , N is Ne + 1, !.
conta(Elem, [X|Y], N) :- conta(Elem,Y,N).

/*Dada uma lista Lin, retorna o elemento e quantas vezes ele aparece*/
monta_pares([],[]):-!.
monta_pares([X|Y], [[X,N]|Lout]) :- conta(X, Y, N), rem_todas(X, Y, Ls),
monta_pares(Ls,Lout).

/*Conta atomos: mostra quantas vezes elementos atômicos
(átomos + lista vazia) aparecem,
somente se repetidas nas duas listas*/
conta_atomos([],[],[]) :- !.
/*Nesta ordem: L1 é "desparentizado, L2 é desparentizado;
é feita a junção das duas listas resultantes do processo anterior, e,
depois,
a lista de pares é criada*/
conta_atomos(L1,L2,Lout) :- desparentize(L1,L1desp),
desparentize(L2,L2desp),
tira_ao_comuns(L1desp,L2desp,Lj),monta_pares(Lj,Lout).
```

Os predicados “conc” e “inter” foram retirados de exemplos mostrados em classe. Os predicados “rem_todas” e “conta” foram baseados em exemplos mostrados em classe, mas foram modificados para se adequarem ao que foi pedido no trabalho.

2. Explicação dos predicados

O funcionamento dos predicados “conc” e “inter” constam nos slides 2 e 3 da sobre Prolog.

2.1 - Predicado “rem_todas”

Este predicado recebe um elemento E, uma lista Lin e outra lista Lout. Sua função é remover todas as ocorrências do elemento E da lista Lin e devolver o resultado em Lout.

`rem_todas(X, [X|L], L1) :- rem_todas(X, L, L1), !.` : se o primeiro elemento de Lin é igual ao elemento, o predicado é chamado novamente utilizando a cauda da lista de entrada como Lin.

`rem_todas(X, [Y|L], [Y|L1]) :- rem_todas(X, L, L1).` : se a cabeça de Lin é diferente do elemento X, esta é agregada à lista de saída e o predicado é chamado utilizando as caudas das listas como entrada e saída.

2.2 - Predicado “desparentize”

`desparentize([X|Y], [X|Z]) :- atomic(X), desparentize(Y, Z), !.` : Se X for um elemento atômico (átomos ou lista vazia) é agregado à lista e o predicado é chamado novamente com o resto das listas como argumentos.

`desparentize([X|Y], Z) :- desparentize(X, X1), desparentize(Y, Y1), conc(X1, Y1, Z).` : esta parte do predicado será executada somente se o elemento for uma lista. Neste caso o predicado desparentize é chamado utilizando o elemento como lista de entrada.

2.3 – Predicado “tira_não_comuns”

Este predicado, nada mais é, que a aplicação de intersecção da primeira lista fornecida como argumento com a segunda, e vice-versa. Isto garante que todos os elementos comuns pertencentes às listas sejam retornados na lista de saída.

2.4 – Predicado “conta”

É verificado se o elemento da cabeça da lista de entrada é igual ao elemento fornecido. Se for, o contador é incrementado. Senão, a execução continua fornecendo, sempre, a cauda da lista como entrada. Essa verificação é feita toda vez que o predicado é chamado.

3. Exemplos de execução do programa

```
?- conta_atomos([1,2,3,[a,b,c,[],[c,d,c,[]]]],[[],c,a,2,[3,2,1]],L).
L = [[1, 2], [2, 3], [3, 2], [a, 2], [c, 4], [[], 3]].
```

```
?- conta_atomos([A,B,c,d,e,[[a,b],B,2,3],l,m,[3,4]],[m,n,o,[o,p,q],A,B],Lb).
A = B, B = [],
Lb = [[[], 3], [m, 2]].
```