

João Vitor Azevedo Marciano 743554
Vitor Hugo Chaves Cambui 744358

May 2019

Conteúdo

1	Objetivo	2
2	Código-Fonte	2
2.1	Desparentiza	2
2.2	Tira Duplicatas	2
2.3	Remove Comuns	3
2.4	Remove	3
2.5	Contar Repetições	3
2.6	Montar os Pares	3
2.7	Contar os Átomos e montar Pares	3
2.8	Retornar L em L2 após contar pares e remover repetições	3
2.9	Função Principal	4
3	Casos de Teste	4
3.1	Caso 1(Informado no Problema)	4
3.2	Caso 2	5



1 Obietivo

Escreva um programa Prolog para, dadas duas listas L1 e L2 que contém elementos de qualquer tipo, possivelmente com repetições e com sublistas, construir outra lista que mostre quantas vezes cada elemento atômico (átomo, número, lista vazia) ou estrutura, que não seja lista, aparece apenas em uma das duas listas dadas, inclusive nas sublistas. A lista resultante deve conter pares de elementos (lista de dois elementos) sendo o primeiro elemento do par um elemento que só aparece em uma das duas listas dadas e o segundo elemento do par, o número de vezes que esse elemento aparece na lista. Os elementos que aparecem nas duas listas não devem ser contados e não aparecem na lista final (ver exemplo dos átomos b, número 5, e estrutura par(c,d) nas listas abaixo). Variáveis também devem ser descartadas (ver exemplo das variáveis Z e F nas listas abaixo). Os átomos devem aparecer na lista resultante na mesma sequência em que aparecem nas listas originais: primeiro os da primeira lista e depois os da segunda lista. Por exemplo, dadas as listas:

```
L1 = [a, b, Z, [a, x], [5,x], [x], par(c,d)] e
L2 = [4.6, 5, w, [], F, b, [], par(c,d), [], par(1,2)]
```

Deve ser construída a lista

```
Lout = [[a, 2], [x, 3], [4.6, 1], [w,1], [[, 3], [par(1,2), 1]].
```

2 Codigo-Fonte

Para código mais "copiável" visitar [esse link](#)

2.1 Desparentiza

```
%Remove os parenteses de tudo da lista, deixando tudo no mesmo nivel.
desparentiza([],[]):-!.
desparentiza([X|Y],[X|Z]):- atomic(X), desparentiza(Y,Z).
desparentiza([X|Y],Z):- is_list(X),desparentiza(X,L),desparentiza(Y,L1),append(L,L1,Z),!.
desparentiza([X|Y],Z):- var(X), desparentiza(Y,Z).
desparentiza([X|Y],[X|Z]):- desparentiza(Y,Z).
```

2.2 Tira Duplicatas

```
%Tira elementos duplicados em uma lista.
tira_duplicatas([],[]).
tira_duplicatas([X|Y],Lout):-member(X,Y),tira_duplicatas(Y,Lout).
tira_duplicatas([X|Y],[X|Lout]):-not(member(X,Y)),tira_duplicatas(Y,Lout).
```

2.3 Remove Comuns

%Pega duas listas, e salva em uma terceira todos os elementos da primeira que
nao estao na segunda.
remove_comuns([],-,[]):-!.
remove_comuns([X|Y],L2,Lout):-member(X,L2),remove_comuns(Y,L2,Lout),!.
remove_comuns([X|Y],L2,[X|Lout]):-remove_comuns(Y,L2,Lout),!.

2.4 Remove

%Pega duas listas, e coloca em uma terceira os elementos que nao estao na
interseccao das duas primeiras.
remove(L1,L2,L3):-remove_comuns(L1,L2,L4),remove_comuns(L2,L1,L5), append(L4,L5,L3).

2.5 Contar Repetições

%Conta quantas vezes o elemento X se repete na lista L e retorna em N
contador(-,[],0):-!.
contador(X,[X|L],N):-contador(X,L,NewN), N is NewN+1,!.
contador(X,[_|L],N):-contador(X,L,N).

2.6 Montar os Pares

%Pega um elemento X e um numero N e monta um par da forma [X,N]
montar_pares(X,N,[X,N]).

2.7 Contar os Atomos e montar Pares

%Pega uma lista L, conta os átomos na lista, e monta uma lista com o par de
[atomo, Numero de ocorrências]
conta_atomos([],-,[]):-!.
conta_atomos([X|Y],L1,[L2|Lout]):- contador(X,L1,N), montar_pares(X,N,L2), conta_atomos(Y,L1,Lout).

2.8 Retornar L em L2 apos contar pares e remover repetições

%Pega uma Lista e retorna em L2 a lista após contar os pares e remover as
repeticoes
monta_pares(Lin,Lout):-conta_atomos(Lin,Lin,Lt),tira_duplicatas(Lt,Lout).

2.9 Função Principal

%Lê duas listas de entrada, faz o nivelamento nas duas, remove os elementos comuns às duas listas, e monta os pares conforme o enunciado.
main(Lout):-read(L1),read(L2),desparentiza(L1,L3),desparentiza(L2,L4),remove(L3,L4,L5),
monta_pares(L5,Lout).

3 Casos de Teste

Como utilizar o programa: após carregar a base de fatos/regras PLP-T1.pl, digitar "main(Lout)." e dar enter.

As próximas duas linhas devem ser as 2 listas de entrada, e cada uma delas deve ser terminada por '.'

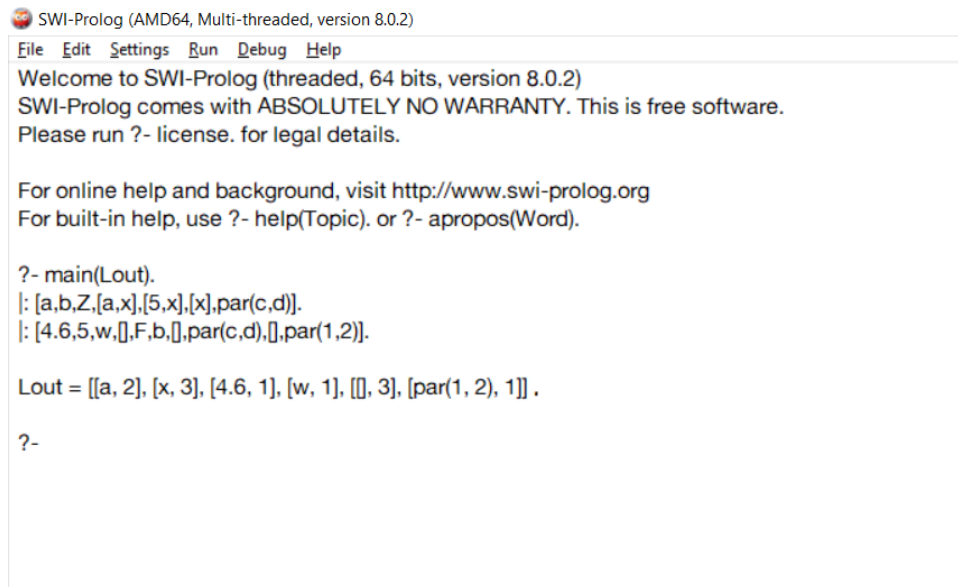
3.1 Caso 1(Informado no Problema)

L1 = [a, b, Z, [a, x], [5,x], [x], par(c,d)]

L2 = [4.6, 5, w, [], F, b, [], par(c,d), [], par(1,2)]

Deve ser construída a lista:

Lout = [[a, 2], [x, 3], [4.6, 1], [w,1], [[], 3], [par(1,2), 1]].



3.2 Caso 2

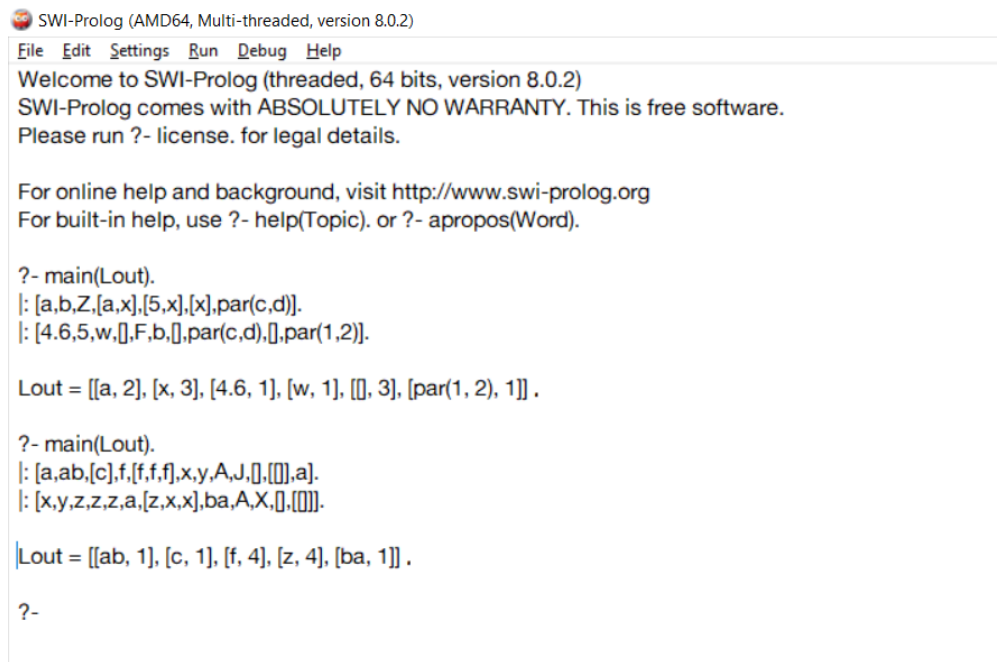
Para as listas:

L1 = [a,ab,[c],f,[f,f,f],x,y,A,J,[],[],a]

L2 = [x,y,z,z,z,a,[z,x,x],ba,A,X,[],[]]

É esperada a resposta

Lout = [[ab, 1], [c, 1], [f, 4], [z, 4], [ba, 1]] .



```
SWI-Prolog (AMD64, Multi-threaded, version 8.0.2)
File Edit Settings Run Debug Help
Welcome to SWI-Prolog (threaded, 64 bits, version 8.0.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit http://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- main(Lout).
|: [a,b,Z,[a,x],[5,x],[x],par(c,d)].
|: [4.6,5,w,[],F,b,[],par(c,d),[],par(1,2)].

Lout = [[a, 2], [x, 3], [4.6, 1], [w, 1], [], 3], [par(1, 2), 1]] .

?- main(Lout).
|: [a,ab,[c],f,[f,f,f],x,y,A,J,[],[],a].
|: [x,y,z,z,z,a,[z,x,x],ba,A,X,[],[]].

Lout = [[ab, 1], [c, 1], [f, 4], [z, 4], [ba, 1]] .

?-
```

Referências

Robert W. Sebesta *Conceitos de Linguagens de Programacao.*