

EP 2 – PARTE A: PROGRAMAÇÃO EM PROLOG: LISTAS E CONJUNTOS

Entrega: 21/04/2020

Motivação

O objetivo deste exercício é acelerar a familiarização dos alunos com conceitos de Programação Lógica e manipulação recursiva de listas, para que todos estejam preparados para a segunda parte do exercício.

Exercício 1

Implementar o predicado `lista_para_conjunto(Xs, Cs)` em que `Cs` é uma lista que contém os mesmos elementos de `Xs`, na mesma ordem de sua primeira ocorrência, mas cujo número de ocorrências é apenas 1. Por exemplo, a consulta

```
?- lista_para_conjunto([1,a,3,3,a,1,4], Cs).
```

sucedendo apenas para `Cs=[1,a,3,4]`. A consulta

```
?- lista_para_conjunto([1,a,3,3,a,1,4], [a,1,3,4])
```

deve falhar.

As listas `Cs` do enunciado anterior serão chamadas de *lista-conjunto*, ou seja, uma lista com apenas uma ocorrência de cada elemento.

Exercício 2

Implementar o predicado `mesmo_conjunto(Cs, Ds)` que sucede quando `Cs` e `Ds` são listas-conjuntos que representam o mesmo conjunto, ou seja, possuem os mesmos elementos ocorrendo uma única vez, em qualquer ordem. A consulta

```
?- lista_para_conjunto([1,a,3,4], Cs)
```

deve suceder quando `Cs` é uma lista que contém qualquer permutação dos elementos `a,1,3,4`.

Exercício 3

Implementar o predicado `uniao_conjunto(Cs, Ds, Es)` em que `Cs`, `Ds`, `Es` são listas-conjuntos em que o terceiro representa a união dos dois primeiros.

Exercício 4

Implementar o predicado `inter_conjunto(Cs,Ds,Es)` em que `Cs`, `Ds`, `Es` são listas-conjuntos em que o terceiro representa a intersecção dos dois primeiros.

Exercício 5

Implementar o predicado `diferenca_conjunto(Cs,Ds,Es)` em que `Cs`, `Ds`, `Es` são listas-conjuntos em que o terceiro representa o conjunto diferença do primeiro, excluindo-se os membros do segundo.

Instruções para entrega

Você está recebendo junto com este enunciado um arquivo chamado `ep2A.pro`, o tendo uma série de testes nos quais seu programa deve passar. Você deverá escrever neste mesmo arquivo as suas respostas aos itens deste enunciado

Você deve submeter via eDisciplinas apenas um arquivo chamado de `ep2A.pro` editado, contendo a sua solução. Para evitar que seu EP seja zerado, certifique-se que o arquivo foi submetido sem problemas (baixando e executando o arquivo do site) e que ele consiste em um programa executável escrito em SWI Prolog com o comando:

```
?-run_tests.
```