

Projeto 5 - Prolog

ATE meia noite de 7/10

Compressão e descompressão de listas

dada a lista

```
[4,4,4,7,7,4,4,0,3,7,7,7,7]
```

vamos definir uma lista comprimida cujos elementos são listas de 2 elementos [item, quantidade] onde quantidade é o número de vezes que o item aparece sequencialmente na lista. Assim, a compressão dessa lista seria:

```
[[4,3],[7,2],[4,2],[0,1],[3,1],[7,4]]
```

- Implemente o predicado `comprime(ListaOriginal, ListaComprimida)`

```
?- comprime([3,3,3,4,5,6,5,5,5,5,7], X).
```

```
X = [[3,3],[4,1],[5,1],[6,1],[5,4],[7,1]]
```

- Implemente a função `descomprime(ListaComprimida, ListaExpandida)` que é o inverso de `comprime`

```
descomprime([[3,3],[4,1],[5,1],[6,1],[5,4],[7,1]], X).
```

```
X = [3,3,3,4,5,6,5,5,5,5,7]
```

Comentário

É possível que você só precise escrever um predicado `comprime(A,B)` que quando dado `A` computa em `B` a compressão, e quando dado `B` (uma lista comprimida) e `A` é uma variável sem valor, retorna em `A` a lista original. Nós já vimos pelo menos dois predicados que funcionam dessa forma bidirecional o `tam` (tamanho de uma lista) e o `append`. Em ambos, não pensamos explicitamente em implementar o predicado de forma bidirecional. Isso aconteceu sem querer. Isso pode acontecer nesse problema. Eu ainda não implementei esse predicado mas minha intuição diz que isso pode acontecer. Mesmo com a minha experiência de Prolog eu não sei como escrever predicados que deliberadamente funcionem de forma bidirecional mesmo quando isso é possível. Em suma, implementem o `comprime`, e testem se ele funciona de forma bidirecional - e assim você não precisará implementar o `descomprime`.