

IF708 - Programação Funcional

Lista de exercícios

1. O algoritmo de Luhn (https://en.wikipedia.org/wiki/Luhn_algorithm) é usado para verificar números de cartões de crédito, por exemplo, por erros de digitação. O algoritmo funciona da seguinte maneira: considera cada dígito com um número separado movendo para a esquerda, dobra número sim, número não a partir do segundo número, partindo da direita subtrai 9 de cada número que é maior que 9 soma todos os números resultantes se o total é divisível por 10, o número do cartão é válido

Defina uma função `luhnDobro :: Int -> Int` que dobra o valor de um dígito e subtrai 9 se o resultado for maior que 9. Usando a função `luhnDobro` e o inteiro resultante da função `mod`, defina a função `luhn :: Int -> Int -> Int -> Int -> Bool` que decide se um cartão de quatro dígitos é válido

2. Defina uma função recursiva `euclides :: Int -> Int -> Int` que implementa o algoritmo de Euclides para calcular o máximo divisor comum entre dois inteiros não negativos: se os dois números forem iguais, o próprio número é o resultado; do contrário, o menor número é subtraído do maior e o mesmo processo é repetido.
3. Redefina as funções `map` e `filter` usando `foldr`
4. Usando a notação de compreensão de listas, defina as funções

`subLists , subSequence :: [a] -> [[a]]`

que retornam todas as sublistas e subsequências de uma lista. Uma sublista é obtida por omitir alguns dos seus elementos; uma subsequência é um bloco contínuo de uma lista. Por exemplo, tanto `[2,4]` quanto `[3,4]` são sublistas de `[2,3,4]`, mas apenas `[3,4]` é uma subsequência.

5. Compreensão de listas podem ser traduzidas em expressões envolvendo `map`, `filter` e `concat`

```
[x | x <- xs] = xs
[f x | x <- xs] = map f xs
[e | x <- xs, p x, ...] =
  [e | x <- filter p xs, ...]
[e | x <- xs, y <- ys, ...] =
  concat [ [e | y <- ys, ...] | x <- xs ]
```

Traduza as expressões

```
[m*m | m <- [1 .. 10] ]
[m*m | m <- [1 .. 10], m*m < 50 ]
[ x+y | x <- [1 .. 4], y <- [2 .. 4] , x > y ]
[ x:p | x <- xs, p <- perms (xs\\[x]) ]
```

usando as expressões acima. Funções auxiliares podem ser necessárias como parte da tradução