Paradigmas de Programação

Fabrício Olivetti de França

06 de Junho de 2018

Lista de exercícios 01

1. Coloque parênteses nas seguintes expressões:

```
2^3*4
2*3+4*5
2+3*4^5
2+3/4-5^6*7
```

2. Sem utilizar qualquer ajuda, determine o valor e o tipo retornado por essas expressões. Em seguida, utilize o *ghci* para confirmar a resposta:

```
(* 9) 6

head [(0,"doge"),(1,"kitteh")]

head [(0 :: Integer ,"doge"),(1,"kitteh")]

if False then True else False

length [1, 2, 3, 4, 5]

length [1, 2, 3, 4] > length "TACOCAT"
```

3. Defina uma função para seguinte assinatura:

```
f :: (a, b) -> (c, d) -> ((b, d), (a, c))
```

4. Defina uma função

```
palindromo :: (Eq a) => [a] -> Bool
```

que verifica se uma string (ou lista) é palíndroma, utilizando a função `reverse`.

5. Mostre que a seguinte função curried pode ser formalizada em termos de expressões lambda:

```
mult :: Int -> Int -> Int -> Int
mult x y z = x*y*z
```

6. Mostre como o operador || pode ser definido de quatro modos diferentes usando pattern matching.

7. Sem usar outras bibliotecas, funções ou operadores, mostre que a definição por pattern matching de &&

```
True && True = True
_    && _    = False
```

pode ser formalizada utilizando duas expressões condicionais (*if*) aninhadas.

8. Faça o mesmo do exercício anterior para essa definição alternativa de &&:

```
True  && b = b
False && _ = False
```

usando dessa vez uma única expressão condicional.

9. Defina a única função possível para a assinatura

```
c :: a -> b -> a
```

10. Defina a única função possível para a assinatura

```
co :: (b -> c) -> (a -> b) -> a -> c
```