

## Segunda lista de exercícios

Prof.: Carlos Camarão

10 de Abril de 2017

1. (2 pontos) Defina uma função *quebr2* que separe uma lista  $x$  em duas metades  $(x_1, x_2)$ , tais que  $x_1 ++ x_2 = x$  e o comprimento de  $x_1$  é igual ao de  $x_2$  (se o tamanho de  $x$  for par) e um a menos do que o de  $x_2$  caso contrário.

Por exemplo:  $\text{quebr2 } [1,2,3,4,5,6] = ([1,2,3], [4,5,6])$   
 $\text{quebr2 } [1,2,3,4,5,6,7] = ([1,2,3], [4,5,6,7])$

2. (2 pontos) Escreva função *splitWith* tal que *splitWith*  $p$   $x$  retorna a lista  $xs$  que quebra a lista  $x$  em elementos para os quais o predicado  $p$  é *False*.

Por exemplo, *words* poderia ser definida como *splitWith isSpace*.

3. (2 pontos) Usando *lines*, *words* e *map*, defina função que leia um arquivo texto de entrada e imprima em um arquivo de saída de mesmo nome mas extensão ".sai" a primeira palavra de cada linha do arquivo de entrada.

Para ler um arquivo, use: *readFile :: FilePath -> IO String*; o (primeiro) argumento de *readFile* é o nome do arquivo.

Para imprimir, use *writeFile :: FilePath -> String -> IO ()*. Nota: *writeFile* *arq* *s* escreve *s* no arquivo de nome *arq*.

Use notação **do** para escrever ações (valores construídos com o construtor de tipos *IO*).

4. (3 pontos) Considere a definição:

$\text{todosParesNats} = [(x, d-x) \mid d \leftarrow [0..], x \leftarrow [0..d]]$

Reescreva *todosParesNats* sem usar essa notação especial de Haskell para definição de listas, com geradores e filtragem.

Dica: use *map*.