A Cifra de Vigenère

Programação Funcional em Haskell

Prof. Rodrigo Ribeiro

Cifra de Vigenère — (I)

- Similar a Cifra de César porém, a chave possui o mesmo tamanho do texto.
- Idéia:
  - Replicar a chave até que esta possua o mesmo tamanho do texto.
  - Usar o mesmo algoritmo, mas usando para cada posição um deslocamento possivelmente diferente.
- Exemplo:
  - Texto: “Haskell without ask is Hell”
  - Chave: “Heaven”
  - Repl.: “HeavenHeavenHeavenHeavenHea”

Cifra de Viginère — (II)

- Primeira questão: Como gerar a chave replicada?
  - Será gerado um número infinito de repetições da chave e depois usaremos take, para obter a chave do tamanho do texto a ser encriptado.
- Mas como gerar um número infinito de repetições da chave?
  - Solução: usar a função cycle:

```
cycle :: [a] -> [a]
cycle xs = xs ++ cycle xs
```

Cifra de Viginère — (III)

- Gerando a chave

```
module Main where
import Data.Char

keys :: Int -> String -> [Int]
keys n  = map ord . take n . cycle
```

Cifra de Viginère — (IV)

- Encriptando o texto

```
shift :: Int -> Char -> Char
shift n = chr . (+ n) . ord

encode :: String -> String -> String
encode k t = zipWith shift (keys n k) t
             where
               n = length t
```

Cifra de Viginère — (V)

- Decodificando

```
decode :: String -> String -> String
decode k t = zipWith shift ks t
             where
               n = length t
               ks = map negate (keys n k)
```

Cifra de Viginère — (VI)

- Testando a implementação

```
test :: String -> String -> Bool
test k t = decode k (encode k t) == t

*Main> test "Haskell without ask is hell" "heaven"
True
```

Cifra de Viginère — (VII)

- Finalizando...
  - Assunto do próximo encontro: tipos de dados algébricos e sinônimos de tipos.
  - Leituras recomendadas:
    - Real World Haskell: Capítulos de 1 a 4
    - Learn you a Haskell: Capítulos 7 e 8 (até a seção recursive data strutures, inclusive)
- Último encontro antes do recesso...
  - Tarefa para entrega depois do recesso.