INE5416 - Paradigmas da Programação (2015/2)

Relatório 5: Análise Léxica: Sintaxe

Caique Rodrigues Marques 13204303

Parte 1

Há duas formas de representar funções em Haskell, da forma indentada ou da forma pontuada. Abaixo, dois exemplos nas respectivas formas:

1. Esta função foi montada de forma indentada, como se nota nas linhas que compõem a sua saída. Seu funcionamento é simples: dado um número de entrada x, se ele for zero, a função retorna 1; se x for 1, retorna 5; se x for 2, retorna 2 e, por fim, se x for qualquer outro número, a função retorna 1.

```
f x = case x of
    0 -> 1
    1 -> 5
    2 -> 2
    _ -> 1
```

2. A função a seguir foi montada de forma pontuada, como se percebe em sua última linha. Quicksort representa o famoso algoritmo de ordenação de mesmo nome, dado um vetor de números como entrada, a função ordena o vetor e o retorna com os elementos ordenados de forma crescente.

```
quicksort :: (Ord a) => [a] -> [a]
quicksort [] = []
quicksort (x:xs) = quicksort lt ++ [x] ++ quicksort ge where
{ lt = [y | y <- xs, y < x]; ge = [y | y <- xs, y >= x] }
```

Parte 2

1. **Lista com elementos de 1 até 1000**

```
list = [1..1000]
lambda1 = map(\x->x) [1..1000] --Usando calculo lambda
```

2. **Progressão aritmética de 1 a 99 de razão 3**: Em Haskell é possível criar listas seguindo um padrão básico pré-estabelecido, portanto, para uma progressão aritmética, basta apenas somar o primeiro elemento com três, o resultado disto também é somado com três e assim sucessivamente.

```
ap = [1,4..99]
lambda2 = map(\x->x) [1,4..99] --Usando calculo lambda
```

3. **Progressão geométrica de 50 termos de razão 2**: Numa progressão geométrica, dado um primeiro termo, o segundo termo é o primeiro termo vezes a razão, o terceiro termo é o segundo termo vezes a razão e assim sucessivamente. A partir do primeiro elemento $x = 3$ é possível alcançar os posteriores, sendo razão igual a 2:

$$a_1 = x.2^0 = x.1 = 3$$
$$a_2 = x.2^1 = x.2 = 6$$
$$a_3 = x.2^2 = x.4 = 12$$
$$a_n = x.2^n$$

Portanto, em Haskell:

```
gp n = [n*(2**(x-1)) | x <- [1..50]]
```

4. **O n-ésimo elemento de uma lista de fatoriais**: O fatorial de um número n é determinado por $n! = n.(n-1).(n-2).....1$, portanto, basta multiplicar os números de 1 até n e ter-se-á o n-ésimo elemento de uma lista cujos elementos são os fatoriais de 1, 2, 3, etc..

```
fat n = product [1..n]
```