Universidade Federal de Pernambuco
Centro de Informática

Paradigmas de Linguagens de Programação - Primeira Prova

André Santos
20 de dezembro de 2023

Nome: _____ CPF: _____

1. (2,5 pontos) 1. A sequência de Fibonacci é uma sequência de números que começa com os números zero e um e os números seguintes são dados pela soma dos dois números anteriores. Ela é infinita. Seus primeiros números são 0,1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, …
   Escreva o código necessário para gerar uma lista infinita contendo os números de Fibonacci, a partir da lista de entrada contendo zero e um.

   ```
   fibonacci :: [Int]
   ```

   Para testar, selecione os 20 primeiros números da lista e confira com a lista acima.

2. (2,5 pontos) Escreva uma função que recebe duas listas já ordenadas e faz o *merge* (combina) as duas listas, dando como resultado uma lista também ordenada. O algoritmo deve levar em conta que as duas listas de entrada já estão ordenadas.

   ```
   merge :: Ord t => [t] -> [t] -> [t]
   ```

   Teste dando como entrada duas listas ordenadas de inteiros ou de letras.

3. (2,5 pontos) Uma estrutura de dados **Pilha** pode ser representada por uma lista onde a cabeça da lista é o topo da pilha. Por exemplo, uma Pilha com dois elementos sendo o elemento 3 no topo, e abaixo dele o elemento 5 seria representada como 3:5:[] (ou [3,5]).
   Dado o tipo de dados **`Elemento`** abaixo, que representa valores (números inteiros) ou operações aritméticas, escreva uma função que recebe uma Pilha (lista) de Elementos e gere a String da expressão correspondente:

   ```
   type Pilha t = [t]

   exemploPilhaElem :: Pilha Elemento
   exemploPilhaElem = [Valor 10, Valor 20, Soma, Valor 30, Multiplica]

   gera_string :: Pilha Elemento -> String
   -- exemplo de uso: gera_string exemploPilhaElem ⟶ "((10+20)*30)"

   data Elemento = Valor Int | Soma | Multiplica deriving (Show)
   ```

4. (2,5 pontos) Escreva uma função que calcula o resultado da expressão representada em uma Pilha de Elementos. Assuma que a Pilha só contém expressões válidas.

   ```
   calcula :: Pilha Elemento -> Int
   -- exemplo de uso: calcula exemploPilhaElem ⟶ 900
   ```

   Sugestão: sempre que encontrar na Pilha (lista) dois valores seguidos de uma operação, remova esses 3 Elementos e insira o Valor resultante do cálculo no topo da pilha. Repita essa operação até que a pilha tenha apenas um Valor.