

Disciplina: Paradigmas de Programacao
Professor: Maicon Rafael Zatelli
Entrega: Moodle

Atividade X - Haskell

Atencao: Faça um ZIP com todos os arquivos de solucao. Use o nome do arquivo de maneira a entender qual problema você esta resolvendo. Por exemplo, problema1.hs, problema2.hs e assim por diante.

Resolva os seguintes problemas na linguagem Haskell:

1. Altere o exemplo visto em sala sobre a operacao `Div Expr Expr` para suportar tambem as operacoes `Mul Expr Expr` (para multiplicacao), `Add Expr Expr` (para soma) e `Sub Expr Expr` (para subtracao). Assim, sua solucao deve suportar expressoes como `(Mul (Div (Add (Val 28) (Val 2)) (Sub (Val 6) (Val 1)))) (Val 3))`. Faça as modificacoes necessarias nas funcoes de avaliacao criadas e crie ao menos três exemplos de expressoes para testar sua solucao. As modificacoes devem ser feitas nas três formas de criar esta funcao vistas em sala de aula (sem usar monads, usando `>>=` e usando a notacao `do`). Não preoque-se com precedência de operadores. Qual forma foi mais facil de dar manutencao? E mais difcil?
2. Alem das operacoes inclu das no problema anterior, inclua tambem uma operacao para raiz quadrada, na forma `Sqrt Expr`. Trate a situacao de raiz quadrada de numero negativo como erro (utilize o `Monad Maybe` para isso). Faça as modificacoes necessarias nas funcoes de avaliacao criadas e crie ao menos três exemplos de expressoes para testar sua solucao. As modificacoes devem ser feitas nas três formas de criar esta funcao vistas em sala de aula (sem usar monads, usando `>>=` e usando a notacao `do`). Não preoque-se com precedência de operadores. Qual forma foi mais facil de dar manutencao? E mais difcil?
3. Crie uma funcao com a assinatura `formaTriangulo :: (Float, Float) -> (Float, Float) -> (Float, Float) -> Maybe Float`, a qual recebe três pontos 2D como parâmetro e retorna a area do triângulo ou `Nothing`, caso os três pontos nao formem um triângulo. Utilize o conceito de `Monad` para esta questao.
4. Considere a existência de um tabuleiro de Xadrez 8x8, onde temos um cavalo e desejamos efetuar movimentos. Para um determinado movimento, devemos qual casa gostar amos de colocar o cavalo, porem o cavalo jamais podera efetuar um movimento invalido, caso contrario o resultado da sequência de operacoes deve ser um erro, assim como ocorria nas questoes anteriores. Um movimento invalido e o cavalo ir para uma casa inexistente do tabuleiro ou mesmo alcancar uma casa invalida para as regras do cavalo. Uma sequência de movimentos de um cavalo pode ser representada por meio de uma lista de duplas `[(Int, Int)]`, onde cada duplas indica a posicao desejada do cavalo apos a realizacao do suposto movimento. Assim, crie uma funcao `calcPosicaoFinal :: [(Int, Int)] -> Maybe (Int, Int)` a qual deve retornar o ultimo elemento da lista, caso todos os movimentos do cavalo forem validos, ou deve retornar `Nothing`, caso algum movimento foi invalido. Assuma que o primeiro elemento da lista e a posicao inicial do cavalo. Assuma tambem que as posicoes do tabuleiro comecam em 0 e terminam em 7, sendo o canto esquerdo inferior a posicao (0,0) e o canto direito superior a posicao (7,7). Por exemplo, a sequência `[(1,0), (2,2), (0,3), (2,4)]` e uma sequência de movimentos valida, mas a sequência `[(1,0), (3,2), (0,3), (2,4)]` nao. Crie funcoes auxiliares, caso necessario. Utilize o conceito de `Monad` para esta questao.
5. Pesquise sobre a classe `Monad` do Haskell. Que outras operacoes, alem de `return` e `>>=` existem nela? Qual a diferenca entre `>>` e `>>=`?
6. Pesquise sobre as classes `Applicative` e `Functor`. O que sao? Para que servem? Faça um pequeno exemplo utilizando cada uma delas.
7. Crie uma funcao `escreva :: String -> Int -> IO ()`, a qual recebe como parâmetros uma string e um inteiro `n` e imprime na tela os `n` primeiros caracteres da string. Para imprimir cada caracter, utilize a funcao `putChar`. Não crie nenhuma funcao auxiliar e nem altere os parâmetros da funcao `escreva`. Crie uma versao usando `>>=` e outra usando a notacao `do`.
8. Crie uma funcao `echo :: IO ()`, a qual lê um caracter e simplesmente o imprime na tela. Não crie nenhuma funcao auxiliar e nem altere os parâmetros da funcao `escreva`. Crie uma versao usando `>>=` e outra usando a notacao `do`.

9. Pesquise sobre o `Monad Either` disponível no Haskell. Faça um pequeno exemplo e explique o seu funcionamento.
10. Listas, em Haskell, também são Monads. Faça um pequeno exemplo utilizando listas (e seus conceitos de Monads) e explique o seu funcionamento.