

INE5416 - Paradigmas da Programação (2015/2)

Relatório 10: Mônadas
Caique Rodrigues Marques 13204303

Parte 1

- Homologia algébrica envolve definições abstratas utilizadas para o estudo desta área. Um espaço topológico pode ser definido como um conjunto de pontos e seus vizinhos, que todos satisfazem uma série de axiomas; permitem formalizar conceitos de continuidade e convergência. A homologia algébrica é a área de matemática que estuda estes espaços em uma visão algébrica.
- Functor, em teoria de categorias (área da matemática responsável em estudar as estruturas abstratas, os morfismos), é um tipo de mapeamento entre duas categorias, preservando a estrutura. Functor pode ser considerado como um homomorfismo de categorias.
- Mônadas, em programação funcional, são estruturas que permitem a realização de tarefas numa sequência de passos, como programação imperativa, essas estruturas permitem ao programador adicionar tarefas quando achar necessário. Um functor funciona como um mapeador (`map`) de quaisquer tipos de variáveis passados como parâmetro.

Parte 2

Facilmente percebe-se o uso de mônadas no código em Haskell, outro ponto a notar é a semelhança ao paradigma imperativo: atribuições em sequência de condições, onde nesta não faz nada se a divisão for por zero ou realiza apenas a divisão em si. A seguir, uma simples definição de soma de dois elementos. Na função `resist` é onde ocorre o cálculo da resistência, nota-se a vantagem é que o programador não precisa se preocupar com o caso da divisão por zero, pois o próprio programa já trata disso conforme especificado na função `//`.