

Nota: todos os excertos de código foram executados com `ghci ine5416_r10.hs` e chamados no interpretador.

Parte 1

- O conceito de homologia algébrica envolve outras definições abstratas utilizadas para construir este estudo. Um espaço topológico pode ser definido como um conjunto de pontos e suas vizinhanças, que satisfazem um conjunto de axiomas relacionando estes. Permite formalizar conceitos como convergência e continuidade. A homologia algébrica é o ramo da matemática que estuda estes espaços em um ambiente de caráter algébrico.
- Um functor, em teoria das categorias (ramo da matemática que trata de formas abstratas estruturas matemáticas), é um mapeamento entre categorias que preserva estruturas. Pode-se pensar que um functor é um homomorfismo entre categorias.
- As mônadas, em programação funcional, são estruturas que representam computações definidas como sequências de passos (operações em cadeia, por exemplo), permitindo ao programador processar a informação passo a passo, adicionando mais regras ao processamento quando necessário. Um functor porta-se, de grosso modo, como uma generalização de métodos de mapeamento a quaisquer tipos passados como argumento.

Parte 2

É possível perceber claramente o uso de uma abordagem monádica no código produzido em Haskell, embora ainda mostre-se necessário o uso de estruturas condicionais em uma parte crítica do código. Esquiva-se de uma divisão com zero verificando se algum dos termos é zero, retornando o tipo **Nothing**; a definição explícita de uma função soma segue logo abaixo; e por fim, a resistência de dois resistores é obtida aplicando, em cadeia, todas as mônadas implementadas. O tratamento de divisões por zero, neste caso, torna-se muito mais intuitivo, protegendo o código de tais erros.