



Universidade Federal de Santa Catarina

Campus Trindade

Centro Tecnológico

INE5416 - Paradigmas de Programação

Relatório Trabalho 2

Programação Funcional - Scala

Leonardo Brito (21200896)

Fernanda Müller (21202109)

Isac Martins (21203361)

Florianópolis, 11 de novembro de 2023

1. Análise do problema

Anteriormente foi realizada uma versão de um resolvidor do jogo Kojun em Haskell, a solução utilizava de recursão e de elementos específicos do Haskell para a solução do problema, usando backtracking para resolver a situação do board.

Agora nessa tarefa o objetivo seria transformar o resolvidor para Scala em uma linguagem que funde POO e Programação Funcional, o que difere essencialmente de Haskell na própria filosofia do problema.

2. Solução Adotada

O trabalho em questão é um solver para um quebra-cabeça baseado em Scala, onde o objetivo é preencher uma matriz com números, seguindo determinadas regras de região e adjacência. O pacote `solver` contém a classe principal `MyKojunSolver`, que aceita duas matrizes como entrada: uma matriz de valores (`valueMatrix`) e uma matriz de regiões (`regionMatrix`).

A classe `MyKojunSolver` possui um método privado `createRegionMapping` para mapear cada região (identificada por strings) às suas respectivas posições na matriz. O método `canInsertValue` é usado para verificar se um valor pode ser inserido em uma posição específica da matriz, considerando as regras do jogo: o valor deve ser único na sua região, não pode ser igual aos valores adjacentes e deve respeitar a ordem crescente ou decrescente dentro da região.

A solução do quebra-cabeça é encontrada através do método recursivo `solveKojun`, que tenta inserir valores válidos em cada posição da matriz. Se um valor válido é encontrado, o método prossegue para a próxima posição; caso contrário, ele tenta um valor diferente. Se nenhum valor válido for encontrado para uma posição, o método retorna `None`, indicando que não há solução.

O pacote `utils` inclui a classe `Matrix`, que fornece funcionalidades básicas para manipulação de matrizes, como obter ou definir o valor em uma posição específica, além de verificar a validade de uma posição.

O objeto `TabuleirosPre` no pacote `utils` contém exemplos pré-definidos de matrizes de valores e regiões, usados para testar o solver.

Finalmente, o objeto `Main` é o ponto de entrada do programa, onde o usuário é solicitado a inserir o tamanho da matriz. Com base no tamanho, as matrizes de valores e regiões são obtidas e usadas para criar uma instância de `MyKojunSolver`. O método `solve` é chamado para encontrar a solução do quebra-cabeça, que é então impressa na tela.

3. Organização

O trabalho foi realizado ao longo das semanas disponíveis através de reuniões do discord onde nos juntamos para discutir nossas ideias e trabalhar no projeto utilizando a ferramenta de Live Share do Visual Studio Code.

4. Dificuldades Encontradas

A dificuldade principal foi traduzir os métodos do Haskell para o Scala, utilizando a orientação a objetos de maneira correta, tal como a tipagem mais abstrata e flexível que gerou múltiplos problemas durante a compilação, os problemas de tradução foram tão grande que a maior parte do código foi evidentemente refatorada para o correto funcionamento.