

Relatório de Atividades – Projeto I

Neste trabalho foi implementado um algoritmo para a resolução de tabuleiros de Sudoku. Os tabuleiros consistem de matrizes 9 x 9 separadas em nove sub-matrizes 3 x 3. As regras do jogo são simples, um valor de 1 a 9 pode ser atribuído à uma célula se e somente se:

1. Nenhuma outra célula da coluna é igual ao valor a ser atribuído;
2. Nenhuma outra célula da linha é igual ao valor a ser atribuído;
3. Nenhuma outra célula da sub-matriz é igual ao valor a ser atribuído.

Compilação e Execução

O jogo e solução foram implementados na linguagem Java. Para compilar e executar o programa os seguintes comandos devem ser executados:

Compilação: `javac sudoku/Main.java`

Execução: `java sudoku/Main bt < ../inputs/board1.in`

Onde o parâmetro **bt** pode também receber os valores **fc** e **mvr**. Essa flag tem como objetivo indicar quais heurísticas de poda devem ser usadas no método de resolução do Sudoku:

- a) **bt**: backtracking simples
- b) **fc**: backtracking com forward checking
- c) **mvr**: backtracking com forward checking e MVR

Além disso, o símbolo **<** irá encaminhar para a entrada do código o que estiver escrito no arquivo `../inputs/board1.in`. A primeira linha deste arquivo consiste no número de casos de teste. As próximas linhas contêm números de 0 a 9 separados por espaços, onde 0 indica que a célula do Sudoku deve ser preenchida.

Cada caso de teste deve possuir 9 linhas com 9 algarismos em cada. Os casos de teste são separados por uma linha em branco.

Backtracking Simples

Esse algoritmo é o mais custoso das três opções. É um algoritmo recursivo que a cada iteração avalia se o valor que é atribuído é válido. Para isso todos os valores são testados. Caso haja algum conflito, o valor é ressetado, a recursão volta na pilha de execução e um próximo valor é testado até que todo o tabuleiro esteja preenchido ou que seja constatado que não há solução possível.

Essa versão do método de solução foi implementada com sucesso e consegue solucionar o problema em cerca de 300 mil atribuições.

Backtracking com Forward Checking

Esse algoritmo também é recursivo, porém ele não testa cada solução possível para fazer uma atribuição. Ele mantém uma lista com o domínio de cada célula a ser preenchida, e a cada iteração o algoritmo atualiza esse domínio. Se uma atribuição falhar, o domínio é revertido para a status da atribuição anterior e outro valor é testado. Há uma falha quando o domínio de uma célula fica vazio. Se não há falhas o backtracking é chamado.

Infelizmente esse algoritmo não funcionou. A codificação possui problemas na atualização do domínio das células.

Backtracking com Forward Checking e MVR

É o mesmo do algoritmo anterior, mas a política de MVR ordena o domínio das células, de forma a escolher na próxima atribuição a célula com o menor domínio.

Esse algoritmo não foi implementado.