

MAC0121 relatório EP1

Victor Seiji Hariki

1 A implementação

A especificação foi bem estrita em relação ao método, assim foi usado o *backtracking*, e por isso discutirei detalhes da implementação. Existem quatro seções distintas do código, criadas para maior facilidade no desenvolvimento do algoritmo, situado no arquivo *restaUm.c*:

1. **Types:** Composto apenas pelo arquivo *types.h*, contém as definições dos tipos personalizados utilizados e *typedefs* para eles, como a *stack*, *board* e *move*. Esse arquivo é incluso por todas as outras headers do programa.
2. **Board:** Composto pelos arquivos *board.h* e *board.c*. Contém funções para interagir com a mesa, movimentar peças e lidar com movimentos. A mesa em si é implementada como um único vetor, para facilitar alocação e dealocação. De preferência, qualquer mudança ou leitura da mesa é feita por uso das funções dessa seção.
3. **Stack:** Composto pelos arquivos *stack.h* e *stack.c*. Contém funções para interagir com a pilha. De preferência, qualquer mudança ou leitura da pilha é feita por uso das funções dessa seção. Uma exceção é a função *printStack* do *restaUm.c*.
4. **restaUm:** Composto apenas pelo arquivo *restaUm.c*. Contém o algoritmo e a função *main*, que lê a entrada e a passa pro algoritmo.

O algoritmo funciona por busca exaustiva, e testa todas as possibilidades até chegar à matriz alvo ou esgotarem-se as possibilidades. O tabuleiro é comparado com o tabuleiro alvo quando o número de peças atinge o número de buracos no tabuleiro original, e, caso diferente, ignora todas as possibilidades de movimento a partir da possibilidade atual, pois quaisquer movimentos a mais fariam o número de peças descer, e, tendo menos peças que haviam buracos originalmente, é impossível conseguir o tabuleiro alvo.

2 Resultados

O algoritmo consegue resolver sem muita dificuldade boa parte dos casos até o tamanho 6x6 e a mesa original de resta um em tempo razoável. Há casos que o programa pode demorar algumas horas pra resolver, mas resolve, se possível.

3 Conclusões

Apesar de não utilizar métodos de corte de possibilidades ou heurística, o algoritmo implementado com força bruta é razoavelmente rápido, e seu código é bem simples, principalmente abstraindo boa parte das operações, como foi feito nesse EP.

4 Arquivos enviados

Foram enviados 3 arquivos compactados:

- *EP1.c* : Arquivo para avaliação. Compile-o com as flags descritas no fórum do paca.
- *types.h* : Arquivo contendo as *structs* e *typedefs*
- *board.** : Arquivos do código de manipulação da mesa.
- *stack.** : Arquivos do código de manipulação da pilha.
- *Makefile* : *Makefile* para a compilação da fonte e vinculação dos objetos. Use *make clean* para excluir os arquivos de objeto (**.o*)
- *Relatorio.pdf* : Este relatório.