Trabalho Prático - Matemática Discreta

Victória Olívia Aráujo Vilas Boas - 2019431429

Belo Horizonte, 02 de Outubro 2020

1 Introdução

O objetivo do trabalho proposto foi aplicar os conhecimentos de relações binárias para identificar as relações encontradas em um conjunto de dados de entrada. Este relatório tem o objetivo de apresentar a descrição da resolução dos problemas propostos e a análise de complexidade por tempo e espaço das soluções aqui apresentadas. Os algoritmos foram escritos em linguagem C, e compilados em ambiente macosx. Outros detalhes a respeito do ambiente computacional e mais instruções para execução do código podem ser encontrados no arquivo leiame.txt.

2 Relações binárias

Dado um conjunto de entradas, que chamaremos de A, deve-se identificar as relações binárias existentes nesse conjunto. As relações que devem ser identificadas são:

1. R é reflexiva $\Leftrightarrow \forall x \in A, (x, x) \in R$
2. R é irreflexiva $\Leftrightarrow \forall x \in A, (x, x) \notin R$
3. R é simétrica $\Leftrightarrow \forall$ x e y $\in A,$ se (x,y) $\in R, então (y, x) \in R$
4. R é anti-simétrica $\Leftrightarrow \forall$ x e y $\in A,$ se (x,y) $\in R, e (y, x) \in R$ então x = y
5. R é assimétrica $\Leftrightarrow \forall$ x e y $\in A,$ se (x,y) $\in$ R, então (y,x) $\notin R$
6. R é transitiva $\Leftrightarrow \forall$ x, y e z $\in A,$ se (x,y) $\in$ R e (y, z) $\in R$ então (x,z) $\in R$

Além disso, é necessário verificar se a relação é de ordem parcial, ou de equivalência, e deve-se imprimir o fecho transitivo dessa relação.

2.1 Descrição da resolução

Para a resolução desse problema foram criados dois arquivos c que dividem o programa entre testes de relação, impressão de pares faltantes, e a solução para as entradas do problema.

A estrutura de pastas para a resolução do problema está organizada da seguinte forma:

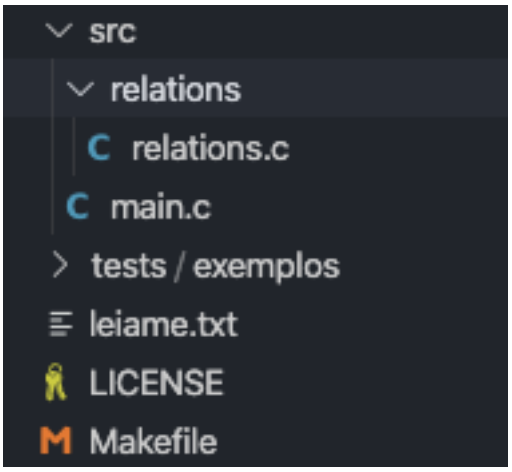

Figure 1: Estrutura das pastas

A pasta ***src*** contém a resolução do problema proposto. Além dela temos a pasta ***tests***, com exemplos de entradas e saídas; o arquivo ***leiame.txt*** com instruções para a execução do programa; o arquivo ***MakeFile*** com diretivas para compilação e execução do programa.

Para a resolução deste problema foi criada uma matriz nXn para mapear as relações e testá-las posteriormente. Além dessa matriz foi criado um vetor contendo todos os elementos que serão mapeados no conjunto de relação. O tamanho n do vetor vai de 0 ao numero de elementos na relação, e cada linha e coluna da matriz é mapeada pelo vetor de elementos.

Exemplo: O vetor com os elementos [3, 5] e a matriz $\begin{matrix} 1 & 1 \\ 0 & 0 \end{matrix}$ podem ser mapeados da seguinte forma: A posição (0,0) da matriz indica a posição zero do vetor. Nesse caso significa que existe uma relação entre (3,3). Um outro exemplo é a linha 0 da matriz e a coluna 1 da matriz, que respectivamente indicam o índice 0 do vetor e o indice 1 do mesmo vetor; ou seja, temos uma relação em (3,5)

Sempre o usuário entrar com um par ordenado do conjunto iremos marcar na matriz de adjacência que a relação existe, até que a entrada seja a string ***EOF***, ou o final de um arquivo. Depois de marcado, iremos verificar todas as relações existentes segundo as regras supracitadas. Por exemplo, para a relação reflexiva iremos verificar se todos as diagonais da matriz de adjacencia estão preenchidas. Todos esses testes estão no arquivo ***relations.c***

2.2 Complexidade

1. Complexidade de tempo

   A maioria das funções da solução possuem uma complexidade quadrática, dado que percorrem uma matrix, exceto o teste que verifica se a matrix é transitiva. Esse tem uma complexidade de ordem cúbica. Seja n o numero de elementos unicos no conjunto de relações, a complexidade de tempo do programa é:

   $$\mathcal{O}(n^3) \tag{1}$$

2. Complexidade de espaço

A complexidade de espaço desse problema, dado que para resolvê-lo usaremos uma matriz de adjacência está em função do tamanho da matriz. Seja n o numero de elementos unicos no conjunto de relações, a complexidade de espaço do programa é:

$$\mathcal{O}(n^2) \tag{2}$$