Documentação do Trabalho Prático

Matemática Discreta

Victoria Rocha Saliba

2018079845

Professor: Antonio Alfredo Ferreira Loureiro
Universidade Federal de Minas Gerais - UFMG

1. Introdução

O trabalho foi criado para retornar a validade das propriedades de uma matriz. Sendo fornecido um conjunto de pares ordenados, o programa analisa os casos de reflexividade, simetria e transitividade e retorna se são válidos ou não. Ele também analisa a relação(equivalência ou ordem parcial) e apresenta o fecho transitivo da relação.

2. Implementação

Para a implementação do trabalho foi criado a variável ‘numNos’, que receberá o número de nós presentes na matriz. Os valores dos nós serão inseridos na mesma linha da entrada e salvos no vetor ‘valorNos’. A partir da segunda linha os pares inseridos representam os pares ordenados, salvos em ‘x’ e ‘y’, a serem observados.

Os valores salvos em ‘maior’ e ‘menor’ serão o maior e o menor valor dos nós inseridos, salvos para serem utilizados posteriormente. A matriz ‘ligacoes’ criada, composta apenas por 0s, será a responsável por dar a visibilidade para os pares inseridos e a partir dela serão realizados os procedimentos de análise.

A função while representa o loop para receber as entradas dos pares ordenados até que o usuário decida parar. Comporta até 50 pares para que o loop não seja infinito, mas como definido na descrição do trabalho não tem a necessidade de comportar casos com mais de 50 pares. Dentro do loop, além de receber os pares, o valor 1 é alocado na posição correspondente na matriz ‘ligacoes’ para indicar a presença do mesmo.

Em seguida, temos as funções que analisam a validade das características a serem retornadas:

Reflexiva: inicialmente falsa. Temos dois ‘for’ que percorrem a matriz e, caso a posição seja na diagonal principal e o valor seja igual a 1, a variável auxiliar é acrescida de 1. No final, se ‘aux’ for igual ao tamanho da diagonal principal a variável ‘reflexiva’ passa a ser verdadeira.

Irreflexiva: inicialmente falsa. Temos dois ‘for’ que percorrem a matriz e, caso a posição seja na matriz principal e o valor seja igual a 0, a variável auxiliar é acrescida de 1. No final, se ‘aux2’ for igual ao tamanho da diagonal principal a variável ‘irreflexiva’ passa a ser verdadeira.

Simétrica: inicialmente falsa. Temos dois ‘for’ que percorrem a matriz e, caso o valor na posição seja igual ao valor na matriz invertida, a variável auxiliar é acrescida de 1. No final, se ‘aux3’ for igual ao tamanho de posições na matriz, a variável ‘simetrica’ passa a ser verdadeira.

Anti-simétrica: inicialmente verdadeira. Temos dois ‘for’ que percorrem a matriz e, caso o valor na posição seja igual ao valor na matriz invertida, fora da diagonal principal e com valor igual a 1, a variável ‘antisimetrica’ passa a ser falsa.

Assimétrica: inicialmente falsa. Temos dois ‘for’ que percorrem a matriz e, caso o valor na posição seja diferente ao valor na matriz invertida e fora da diagonal principal, a variável auxiliar é acrescida de 1. No final, se ‘aux5’ for igual ao tamanho de posições na matriz menos as da diagonal principal, a variável ‘assimetrica’ passa a ser verdadeira.

Transitiva: inicialmente verdadeira. Temos três ‘for’ que percorrem a matriz e, caso o valor na posição seja igual ao valor nas duas matrizes simuladas e diferente na terceira, a variável ‘transitiva’ passa a ser falsa.

Para retornar se a relação de equivalência ou de ordem parcial é feito apenas um if e, caso as condições sejam válidas, retorna verdadeiro.

Em seguida, o resultado das propriedades será mostrado na tela e, nos casos relatados na definição do trabalho prático, teremos os pontos presentes ou ausentes que satisfazem a relação.

Ausentes para não ser reflexiva: caso não seja reflexiva, temos dois ‘for’ que percorrem a matriz e, a posição na diagonal principal com valor diferente de 1(ou seja, 0), será impressa.

Presentes para ser irreflexiva: caso seja irreflexiva, temos dois ‘for’ que percorrem a matriz e, a posição na diagonal principal com valor igual a 1 será impressa.

Ausentes para não ser simétrica: caso não seja simétrica, temos dois ‘for’ que percorrem a matriz e, caso o valor da posição não seja igual ao valor na matriz invertida, e igual a 1, a posição será impressa.

Presentes para ser anti-simétrica: caso seja anti-simétrica, temos dois ‘for’ que percorrem a matriz e, caso o valor da posição seja igual ao valor na matriz invertida, fora da diagonal principal e igual a 1, a posição será impressa.

Presentes para ser transitiva: caso seja transitiva, temos três ‘for’ que percorrem a matriz e, caso o valor na posição seja igual ao valor nas duas matrizes simuladas e diferente na terceira, a posição será impressa.

Fecho transitivo da relação: temos três ‘for’ que percorrem a matriz e, caso o valor na posição seja igual ao valor nas duas matrizes simuladas, a posição será impressa. Tratamos também os outros casos possíveis: a posição é igual apenas em duas matrizes.

2.1 Organização do código, decisões de implementação e detalhes técnicos

O código se encontra em um arquivo chamado “relacao.c” para atender as especificações do trabalho. Foi implementado na linguagem C em uma máquina Windows usando o GCC 7.4.0 (funciona no Linux).

3. Análise de Complexidade - piores casos

Reflexiva:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: $O(1)$, pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Irreflexiva:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: $O(1)$, pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Simétrica:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: $O(1)$, pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Anti-simétrica:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: $O(1)$, pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Assimétrica:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: $O(1)$, pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Transitiva:

- Tempo: $O(n^3)$, pois passa por uma matriz n por n utilizando três ‘for’ para a percorrer e analisar suas posições;
- Espaço: $O(1)$, pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Ausentes para não ser reflexiva:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: $O(1)$, pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Presentes para ser irreflexiva:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: O(1), pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Ausentes para não ser simétrica:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: O(1), pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Presentes para ser anti-simétrica:

- Tempo: $O(n^2)$, pois passa por uma matriz n por n utilizando dois ‘for’ para a percorrer e analisar suas posições;
- Espaço: O(1), pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Presentes para ser transitiva:

- Tempo: $O(n^3)$, pois passa por uma matriz n por n utilizando três ‘for’ para a percorrer e analisar suas posições;
- Espaço: O(1), pois a matriz é alocada antes da função, portanto ela apenas a percorre.

Fecho transitivo da relação:

- Tempo: $O(n^3)$, pois passa por uma matriz n por n utilizando três ‘for’ para a percorrer e analisar suas posições;
- Espaço: O(1), pois a matriz é alocada antes da função, portanto ela apenas a percorre.

4. Referências

Para a implementação do trabalho foram consultados sites que detalhavam a definição matemática dos casos pedidos e os materiais didáticos disponibilizados pelo professor e pelo monitor da matéria.