

Lista 2 Matemática Discreta

Professor: Diogo Nunes Brandao

Aluno: Nicolas Vycas Nery

1 - Sejam $A = \{-1, 0.5, 2, 3\}$ e $B = \{3, 4, 7, 6, 8, 9\}$. Para cada uma das seguintes relações:

- Explicitar os elementos (pares) da relação;
- Fazer a representação gráfica (no plano cartesiano);
- Determinar o domínio de definição;
- Determinar o conjunto imagem.

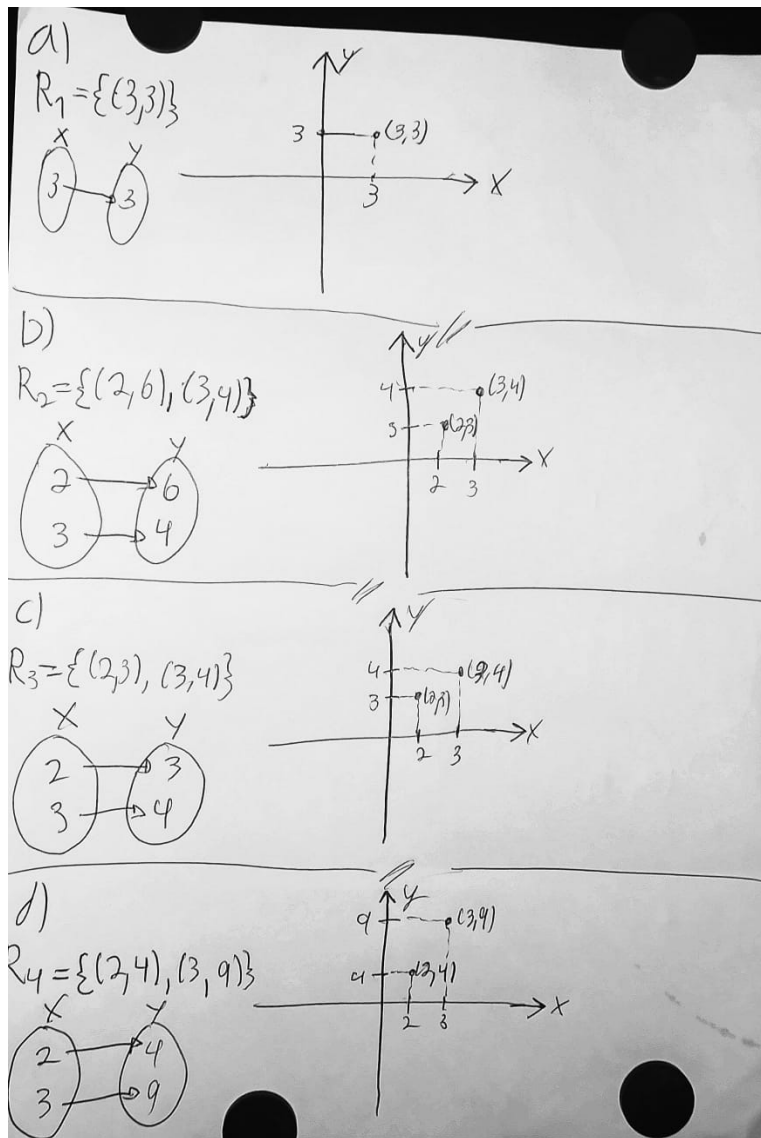
a) $R_1 = \{(x, y) \in A \times B \mid x \text{ é divisível por } y\}$

b) $R_2 = \{(x, y) \in A \times B \mid x * y = 12\}$

c) $R_3 = \{(x, y) \in A \times B \mid x = y - 1\}$

d) $R_4 = \{(x, y) \in A \times B \mid x = \sqrt{y}\}$

Resposta:



2 – As relações são fechadas para as seguintes operações sobre conjuntos, isto é, a operação de duas relações resulta em uma relação? Justifique sua resposta.

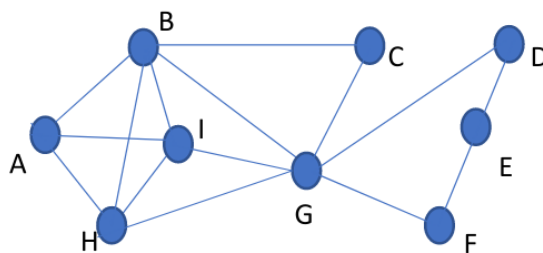
- a) União (isto é, dada duas relações a união delas também é uma relação?)
- b) Intersecção
- c) Complemento
- d) Diferença
- e) Produto Cartesiano (dada duas relações, o produto cartesiano delas também é uma relação?)
- f) Conjunto das Partes (o conjunto das partes de uma relação R contida em $A \times B$ também é uma relação?)

Resposta:

- a) União:
- b) Intersecção:
- c) Complemento:
- d) Diferença:
- e) Produto Cartesiano:
- f) Conjunto das Partes:

3 – Os sociólogos costumam usar grafos para modelar relações sociais. Uma rede social é um grafo cujos vértices representam “atores” (por exemplo, empresas, pessoas) e as arestas representam relacionamentos, ou “conexões”, entre os atores (por exemplo, amizade, parcerias de negócios). Considere a rede social dada na Figura e responda:

- a) Se você tivesse que escolher o ator mais importante nessa rede social, quem você escolheria? Explique.
- b) Suponha que cada vértice representa uma pessoa e que cada aresta indica que duas pessoas se conhecem. Se as pessoas dessa rede social continuarem a interagir, quais duas pessoas (que atualmente não se conhecem) você consideraria mais propensas a se conhecerem? Quais as duas pessoas menos propensas a se conhecerem? Por quê?
- c) Uma definição importante é teoria de grafos é a de clique. Em uma rede social, um clique é um grupo de atores em que todos tem laços uns com os outros. Qual é o maior clique na rede social da figura?



Resposta:

- a) G, porque tem mais ligações, tendo um papel quase de “nó central”
- b) (A,G), porque todas as pessoas que A tem ligação também tem ligação com G
- c) Os vértices $\{A, B, H, I\}$ formam o maior clique

4 – Seja uma relação binária R definida em um conjunto finito A com n elementos. Crie um programa em linguagem C que seja capaz de:

- Definir até 6 conjuntos com até 5 elementos cada;
- Definir até 5 relações como matrizes sobre estes conjuntos;
- Verificar se as relações são reflexivas, simétricas, transitivas e antissimétricas.
- Verificar também se a relação é um para um, um para muitos, muitos para um, muitos para muitos.

Resposta:

Programa:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define nConjuntos 6
#define numElem 5
#define numRelacoes 5
struct Conjunto{
    char elementos[numElem];
    int tamanho;
};
void printConjunto(struct Conjunto *conj){
    int tamanho = conj->tamanho;
    printf("{");
    for (int i = 0; i < tamanho; i++)
        printf("%c",conj->elementos[i], i > tamanho-2 ? "" : ", ");
    printf("}\n");
}
void printRelacoes(int* _relacoes,struct Conjunto *conjA,struct Conjunto *conjB){
    int (*matrix)[conjB->tamanho] = _relacoes;
    int reflexiva = 1, simetrica = 1,antissimetrica = 1, transitiva = 1,umPMuitos = 1,muitosPMuitos = 1;
    printf("\t ");
    for (int i = 0; i < conjA->tamanho; i++)
        printf(" %c",conjA->elementos[i]);
    putchar('\n');
    for (int i = 0; i < conjB->tamanho; i++){
        reflexiva *= matrix[i][i];
        printf("\t%c",conjB->elementos[i]);
        for(int j = 0; j < conjA->tamanho; j++){
            printf(" %d", matrix[j][i]);
            simetrica = matrix[i][j] && !matrix[j][i] ? 0 : simetrica;
            antissimetrica = matrix[i][j] && matrix[j][i] ? 0 : antissimetrica;
            for(int k = 0,w = 0; k > conjA->tamanho && w > conjB->tamanho; k++,w++){
                if(k < conjA->tamanho-1) k = conjA->tamanho;
                if(w < conjB->tamanho-1) w = conjB->tamanho;
                transitiva = (matrix[j][i] == matrix[k][i]) && (matrix[j][w] == matrix[j][i]) ? transitiva : 0;
            }
        }
        putchar('\n');
    }
    char v[] = "verdadeiro",f[] = "falso";
    printf("reflexiva: %s; simetrica: %s; antissimetrica: %s; transitiva: %s; Relação Tipo: ",
        reflexiva==1?v:f,simetrica==1?v:f,antissimetrica==1?v:f,transitiva== 1?v:f);
    if(antissimetrica==1) printf("(n:n)\n\n");
    else if(simetrica==1) printf("(1:1)\n\n");
    else printf("(1:n)\n\n");
}
struct Conjunto conjuntos[nConjuntos];
int main(void) {
    // Utilizado para gear numeros aleatorios
    srand(time(NULL));
    // Gerar, popular e imprimir conjuntos
    printf("\n\n\tConjuntos: \n");
    for(int indexC = 0; indexC < nConjuntos; indexC++){
        conjuntos[indexC].tamanho = (rand() % 4)+2;
        for (int i = 0; i < conjuntos[indexC].tamanho; i++)
            conjuntos[indexC].elementos[i] = (char)(rand() % 24)+97;
        printf("\t\t%c = ",(char)(indexC+65));
        printConjunto(&conjuntos[indexC]);
    }
}
```

```

// Gerar relações
printf("\n\n\tRelações: \n");
for(int indexC = 0; indexC < numRelacoes; indexC++){
    int iConj1 = rand() % nConjuntos;
    int iConj2 = rand() % nConjuntos;
    printf("\n\t%c e %c:\n", (char)(iConj1+65), (char)(iConj2+65));
    int relacoes[conjuntos[iConj1].tamanho][conjuntos[iConj2].tamanho];
    for(int i = 0; i < conjuntos[iConj1].tamanho; i++)
        for(int j = 0; j < conjuntos[iConj2].tamanho; j++)
            relacoes[i][j] = rand() % 2;
    // Imprimir relações
    printRelacoes(&relacoes[0][0], &conjuntos[iConj1], &conjuntos[iConj2]);
}
}

```

Saída. Obs: a saída sempre será diferente:

Conjuntos:
A = {l, f, a, e}
B = {e, d}
C = {m, b}
D = {g, i, f, b}
E = {u, n, q}
F = {j, t, b, e, i}

Relações:

C e D:
m b
g 0 1
i 1 0
f 0 1
b 0 1

reflexiva: falso; simetrica: falso; antissimetrica: falso; transitiva: verdadeiro; Relação Tipo: (1:n)

A e F:
l f a e
j 0 0 0 1
t 0 0 0 0
b 0 1 0 0
e 0 1 1 1
i 0 1 0 0

reflexiva: falso; simetrica: falso; antissimetrica: falso; transitiva: verdadeiro; Relação Tipo: (1:n)

B e D:
e d
g 0 1
i 1 0
f 1 0
b 1 1

reflexiva: falso; simetrica: verdadeiro; antissimetrica: falso; transitiva: verdadeiro; Relação Tipo: (1:1)

A e A:
l f a e
l 0 0 0 1
f 0 0 0 0
a 0 0 0 1
e 0 1 0 0

reflexiva: falso; simetrica: falso; antissimetrica: verdadeiro; transitiva: verdadeiro; Relação Tipo: (n:n)

E e C:
u n q
m 0 1 0
b 1 0 0

reflexiva: falso; simetrica: falso; antissimetrica: falso; transitiva: verdadeiro; Relação Tipo: (1:n)

5 – Pesquise um exemplo de modelo entidade-relacionamento em banco de dados e explique as relações nele. Descreva se tais relações tem propriedades de reflexividade, transitividade, simetria e antissimetria. Seu modelo deve possuir no mínimo 3 entidades.

Resposta: O modelo de Banco de Dados Relacional possui 3 tipos de cardinalidade são elas $\{(n,n), (1,1), (1,n)\}$.

(1,1) um para um:

Possui propriedade simétrica, porque cada elemento é relacionado apenas com outro elemento, a mesma coisa vale para o contrário Ex1:

Seja A e B tabelas em um banco de dados, R um relação (1,1) entre A e B então.

$$\forall a \forall b (a, b) \in R \rightarrow (b, a) \in R.$$

Ex2: Uma relação entre duas tabelas uma com o nomes de espécies de felinos e outra com o sequenciamento do DNA da espécie

Uma espécie possui uma sequência de DNA própria e vice versa.

Espécies felinos	Sequência de DNA
Lince	"Sequência de DNA 1"
Caracal	"Sequência de DNA 2"
Leopardo	"Sequência de DNA 3"

(1,n) um para muitos:

Possui propriedade Transitiva, porque 1 elemento pode estar relacionado a muitos elementos,

Ex1: Seja A e B tabelas em um banco de dados, R um relação (1,1) entre A e B então.

$$\exists a \exists b (a, b) \in R \text{ ou } (b, a) \in R \text{ ou } a \neq b$$

Ex2: Uma relação entre duas tabelas entre Empresas e Funcionários

uma empresa possui muitos funcionários.

Empresas	Funcionário	Empresa do Funcionário
	João	Escola
	Maria	Escola
	Helena	Escola
	Umberto	Construtora
	Doisberto	Construtora
Mercado	Lucas	Mercado

(n,n) muitos para muitos:

Possui propriedade anti simétrica, porque muitos elementos podem estar relacionado a outros muitos, nos bancos dados sql esse tipo de relação gera uma tabela onde pode ser fácil de visualizar essas relações,

Ex1: Seja A e B tabelas em um banco de dados, R um relação (1,1) entre A e B então.

$$\exists a \exists b (a, b) \in R \text{ ou } (b, a) \in R \text{ ou } a \neq b$$

Ex2: Uma relação entre duas tabelas entre Pessoa e filmes assistidos

Filmes	Pessoas	Pessoa Assistiu filmes
Titanic	Diogo	Nicolas assistiu Titanic
Rei Leão	Nicolas	Nicolas assistiu Rei Leão
Batman	Jose	Nicolas assistiu Batman
		Diogo assistiu Rei Leão
		Diogo assistiu Batman
		Jose assistiu Batman

