

Matemática Discreta

Professor: Diego Nunes Brandão

Aluno: Nícolas Vargas Mery

Lista de Exercícios 001

1)

a) ~~A~~ $A = \{a, b, c\}$

R: $\{\{a, b\}\}; \{\{a, c\}\}; \{\{b, c\}\}; \{\{a\}\}; \{\{b\}\}; \{\{c\}\}; \{\emptyset\}$
 $\{\{a, b, c\}\}$ 8 subconjuntos

b) $B = \{a, \{b, c\}, D\}; D = \{1, 2\}$

R: $\{\{a\}\}; \{\{D\}\}; \{\{b, c\}\}; \{\emptyset\}; \{\{a, \{b, c\}, D\}\};$
 $\{\{a, \{\{b, c\}\}\}; \{\{a, D\}\}; \{\{\{b, c\}, D\}\}; \{\{1, 2\}\}; \{\{1\}\};$
 $\{\{2\}\}; \{\{a, \{\{b\}\}\}; \{\{a, \{\{c\}\}\}; \dots\}$

32 subconjuntos

2) Sim, Porque todo subconjunto contém um subconjunto igual a ele mesmo, conjuntos contendo os arranjos de seus elementos e o subconjunto vazio.

3) $\Sigma = \{a, b, c, d, \dots, z\}$ Dígitos = $\{0, 1, 2, 3, \dots, 9\}$

A) $\Sigma^* = \cancel{\{ \text{Combinacões infinitas de traços} \}}$ B) dígitos* = $\{x | x \in \mathbb{N}\}$

C) Se não for contado os acentos para dizer que português é uma linguagem sobre Σ é um subconjunto de Σ^* .

D) SIM OS UNIÃO NATURAIS SÃO UMA LÍNGUAGEM SOBRE DÍGITOS E SUBCONJUNTO DE DÍGITOS*.

4) Somente quando o alfabeto for vazio.
Porque para um alfabeto nulo, ~~foi só~~ que qualquer palavra formada será um palíndromo com possível comprimento infinito, o mesmo raciocínio vale para alfabetos maiores. Apesar de alfabetos vazios ~~ser possíveis~~ termos uma quantidade finita de palíndromos que seja o próprio vazio.

5)

$$\text{a) } A \times (B \cup C) = (A \times B) \cup (A \times C)$$

Seja $(x, y) \in A \times (B \cup C)$

$\Rightarrow x \in A, y \in B \cup C$

$\Rightarrow x \in A, \{y \in B \vee y \in C\}$

$\Rightarrow \{x \in A, y \in B\} \cup \{x \in A, y \in C\}$

$\Rightarrow (x, y) \in (A \times B) \vee (x, y) \in (A \times C)$

$\Rightarrow (x, y) \in (A \times B) \vee (A \times C) \therefore (x, y) \in A \times (B \cup C)$

$\Rightarrow (x, y) \in (A \times B) \cup (A \times C) \therefore A \times (B \cup C) \subseteq (A \times B) \cup (A \times C)$

Seja $(P, 4) \subseteq (A \times B) \cup (A \times C)$

$$\text{b) } A \times (B \cap C) = (A \times B) \cap (A \times C)$$

Seja $(x, y) \in A \times (B \cap C)$

$\Rightarrow \{x \in A\} \wedge (y \in (B \cap C))$

$\Rightarrow \{x \in A\}, \{y \in B \wedge y \in C\}$

$\Rightarrow \{x \in A, y \in B\} \wedge \{x \in A, y \in C\}$

$\Rightarrow (x, y) \in (A \times B) \wedge (x, y) \in (A \times C)$

$\Rightarrow (x, y) \in (A \times B) \wedge (A \times C) \therefore (x, y) \in A \times (B \cap C)$

$\Rightarrow (x, y) \in (A \times B) \cup (A \times C) \therefore A \times (B \cap C) \subseteq (A \times B) \cup (A \times C)$

7)

Si $A \neq B$ conjuntos

$$\rightarrow A - B = A \Leftrightarrow B - A = B$$

$$\text{Se } A - B = A$$

$$\rightarrow A \cap B = \emptyset$$

$$\rightarrow B \cap A = \emptyset$$

$$\therefore B \cap A = \emptyset$$

$$\therefore B - A = B$$

$$\text{Si } B - A = B$$

$$\rightarrow B \cap A = \emptyset$$

$$\rightarrow A \cap B = \emptyset$$

$$\therefore A \cap B = \emptyset$$

$$\therefore A - B = A$$

$$08) A = \{\emptyset, \{a, b\}, a\}$$

$$P(A) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$$

9) O código pode realizar a diferença de conjuntos finitos pois existem limites de hardware e tempo

```
#include <stdio.h>

void print_conjunto(char *conjunto, int tamanho){
    printf("{");
    for (int i = 0; i < tamanho; i++){
        putchar(conjunto[i]);
        putchar(' ');
    }
    printf("}");
}

void uniao_disjunta(char *conjunto1, char *conjunto2, int tamanho1, int tamanho2){
    char resultado[tamanho1 + tamanho2];
    int tamanhoReal = 0;
    for(int i = 0; i < tamanho1; i++){
        for(int j = 0; j < tamanho2; j++){
            if(conjunto1[i] == conjunto2[j]){
                break;
            }else if(j == (tamanho2-1)){
                resultado[tamanhoReal] = conjunto1[i];
                tamanhoReal++;
            }
        }
    }
    for (int i = 0; i < tamanhoReal; i++){
        for (int j = i + 1; j < tamanhoReal; j++){
            if (resultado[i] == resultado[j]){
                for (int k = j; k < tamanhoReal; k++){
                    resultado[k] = resultado[k + 1];
                }
                tamanhoReal--;
                j--;
            }
        }
    }
    print_conjunto(resultado, tamanhoReal);
}

int main(){
    char conjunto1[12] = {'2','a','5','v','8','f','5','a','3','a','5','a'};
    printf("\nA = ");
    print_conjunto(conjunto1, 12);

    char conjunto2[12] = {'a','5','c','2','g','8','t','r','7','a','4','a'};
    printf("\nB = ");
    print_conjunto(conjunto2, 12);

    printf("\nuniao_disjunta(A,B) = ");
    uniao_disjunta(conjunto1, conjunto2, 12, 12);
    printf("\n");
}
```

10) A justificativa do porque os conjuntos devem ser finitos é a mesma da questão 9

```
#include <stdio.h>

void print_conjunto(char *conjunto, int tamanho){
    printf("{");
    for (int i = 0; i < tamanho; i++){
        putchar(conjunto[i]);
        putchar(' ');
    }
    printf("}");
}

void unir_conjuntos(char *conjunto1, char *conjunto2, int tamanho1, int tamanho2){
    char resultado[tamanho1 + tamanho2];
    int tamanhoReal = 0;

    //juntar conjuntos
    for (int i = 0; i < tamanho1; i++){
        resultado[i] = conjunto1[i];
        tamanhoReal++;
    }

    for (int i = tamanho1; i < tamanho1 + tamanho2; i++){
        resultado[i] = conjunto2[i - tamanho1];
        tamanhoReal++;
    }

    //remover elementos duplicados
    for (int i = 0; i < tamanhoReal; i++){
        for (int j = i + 1; j < tamanhoReal; j++){
            if (resultado[i] == resultado[j]){
                for (int k = j; k < tamanhoReal; k++){
                    resultado[k] = resultado[k + 1];
                }
                tamanhoReal--;
                j--;
            }
        }
    }
    //imprimir array
    print_conjunto(resultado, tamanhoReal);

    printf("\n\n\n");
}

int main(){
    char conjunto1[10] = {'1', 'a', '4', '1', '5', 'e', '8', 'b', '9', 'z'};
    char conjunto2[10] = {'2', 'a', '3', 'c', '6', 'd', '7', 'f', '3', 'b'};
    printf("\n\n\tA = ");
    print_conjunto(conjunto1, 10);
    printf("\n\tB = ");
    print_conjunto(conjunto2, 10);
    printf("\n\tA U B = ");
    unir_conjuntos(conjunto1, conjunto2, 10, 10);
}
```