CT-200 Fundamentos de Linguagens Formais e Automata Aula 01 - Introdução

Primeira aula

(updated just now by *YourName*)

Orientações Gerais: Horários e Avaliação

Horários:
3 tempos semanais (teoria) - terças às 9:00

Avaliação (por bimestre):

- Um exame escrito individual em sala.
- Um projeto realizado em dupla.
- Uma lista de exercício realizada individualmente.

Tópicos

- Teoria de Conjuntos
- Provas de Teoremas
- Conjuntos Infinitos
- Estruturas de Dados e Algoritmos
- Linguagens Regulares
- Autômatos Finitos
- Teorema de Kleene
- Propriedades e Minimização de Autômatos
- Gramáticas
- Hierarquia de Chomsky
- Gramáticas Livres de Contexto
- Normalização de Gramáticas
- Autômato de Pilha
- Análise Sintática
- Máquinas de Turing
- Linguagens Irrestritas e Computabilidade

Bibliografia

- **Introduction to Automata Theory, Languages and Computation** - J.E. Hopcroft e J.D. Ullman. Addison-Wesley.

  > Um clássico na área, razoavelmente completo e didático. Provas sumárias de teoremas. Apresenta soluções comentadas para alguns exercícios.

- **Introdução à Teoria de Autômatos, Linguagens e Computação** – J.E. Hopcroft, J.D. Ullman e R. Motwami. Campus.

  > Tradução da 2a. edição americana de versão estendida do Hopcroft/Ullman. Vários erros de tradução. Bem completo.

- **Elements of the Theory of Computation** - H.R. Lewis e C.H. Papadimitriou. Prentice-Hall.

  > Um pouco mais pesado e menos didático do que o anterior. Conciso e objetivo.

- **Languages and Machines** - T. Sudkamp. Addison-Wesley.

> Mais didático do que os anteriores. Ordenação não muito ortodoxa: primeiro discute linguagens e gramáticas e só então introduz a teoria de autômatos.

- **Introduction to the Theory of Computation** – M. Sipser. PWS.

> Conciso e didático.

- Notas de aula

Motivação

Objetivo do curso:
Apresentar os fundamentos da Teoria da Computação.

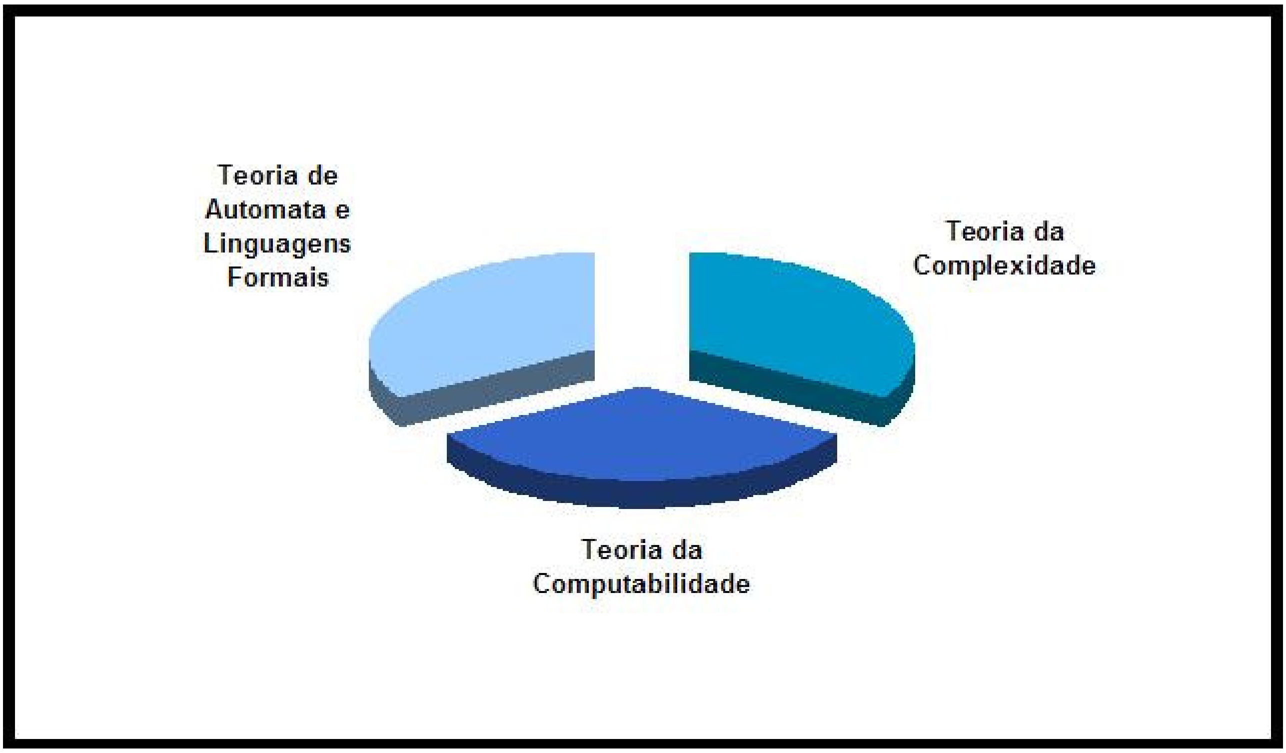
Como definir o processo computacional de maneira formal?
Quais as capacidades / limitações da computação?
Como definir uma máquina abstrata que realiza computações?

Em suma: quais as capacidades e limitações de um computador?

Veremos que existem vários modelos, e que a capacidade computacional de uma máquina é caracterizada pela linguagem formal que ela pode reconhecer.

Quanto mais "poderosa" uma máquina, a linguagem que ela pode reconhecer pode ser mais complexa.

Teoria da Computação - visão geral



Teoria de Autômatos e Linguagens Formais

Teoria da Computabilidade

Teoria da Complexidade

Teoria de Autômatos e Linguagens Formais

O QUE É UM COMPUTADOR?

Existem modelos gerais de computadores que não referenciam nenhuma implementação particular?
Como verificar o que um computador faz?

Exemplos de modelos computacionais:

- Autômato finito: processamento de texto, compiladores e projeto de hardware.
- Gramática livre de contexto: linguagens de programação e Inteligência Artificial.

Noção de Autômato

Autômato:

- uma fita contendo a informação de entrada.
- uma cabeça de leitura que se move pela fita.
- um controle que decide sobre o movimetno da cabeça e produz uma saída.

Modos de operação:

- produzir uma saída de forma autônoma
- entrada traduzida em saída
- entrada aceita ou não

Teoria da Computabilidade

Há problemas que podem ser resolvidos por computadores, e outros que não podem.
Exemplo: Teorema de Gödel

O QUE TORNA ALGUNS PROBLEMAS COMPUTÁVEIS E OUTROS NÃO COMPUTÁVEIS?
QUAIS AS CONSEQUÊNCIAS DISSO PARA O PROJETO DE COMPUTADORES?

Definição de problema como linguagens.
A solução de problemas é equivalente à interpretação de linguagens por autômatos.

Teoria da Complexidade

Há problemas simples e problemas difíceis:

Ordenamento: simples.

Alocação de recursos: complexo.

O QUE TORNA ALGUNS PROBLEMAS FÁCEIS E OUTROS DIFÍCEIS? COMO CLASSIFICAR PROBLEMAS?

Kurt Gödel

- Nascido em 1906 (Brno, Rep. Tcheca), faleceu em 1978 (Princeton, EUA). Doutorado em 1929 (Univ. de Viena).
- Teorema provado em 1931: qualquer sistema matemático baseado em axiomas tem proposições que não podem ser provadas.
- Teorema de Gödel encerrou uma busca de mais de 100 anos por uma base axiomática completa para a Matemática (Hilbert, Russell e outros). É um dos dez mais importantes teoremas provados, considerando-se qualquer área da Ciência.

Mais info

Aplicações diretas

- projeto de hardware (circuitos seqüenciais) e de software (máquinas de estado).
- processamento de texto (corretor ortográfico, reformatação, colorir sintaxe).
- compiladores (análise léxica, sintática, geração de código).
- análise de padrões (combinar modelos matemáticos na forma de linguagens com fenômenos naturais, recursão)

Problemas não computáveis

Demonstrar que nem todo problema pode ser resolvido computacionalmente.

1 - Assumir que para cada função computável há um programa de computador finito.

2 - Assim, o conjunto de todos os programas de computador (possíveis de se escrever) é infinito, mas contável.

3 - Consideramos os problemas que consistem em responder **sim** ou **não** para um determinado número inteiro. (Exemplo: o número é par? o número é primo? o número é perfeito?)

4 - Um problema assim pode ser modelado como uma função f(n) que retorne 0 ou 1.

5 - Suponha que cada problema desse possa ser numerado, assim temos funções f_0, f_1, f_2, ..., f_i, ... (isto é, um conunto

infinito, mas contável de problemas).

6 - A função f(n) definida como

0	se f_n(n)=1
1	caso contrário

não pode corresponder a nenhum inteiro.

Veja que f_j(j)=

0	se f_j(j)=1
1	caso contrário

7 - Logo, a suposição 5 é falsa e o conjunto de problemas é infinito e incontável (maior portanto que o conjunto dos possíveis programas.

Exemplos de problemas não computáveis

1 - O problema da parada:

halting : $Todos.os.programas \to \mathbb{N}$		
halting(p) =	1	se a execução de p vai parar
	0	caso contrário

2 - Verificar se um programa é correto (isto é, se faz sempre o que é previsto para fazer).

Revisão de Teoria dos Conjuntos

Notação de conjunto

A={1,3,7,10}

> A ordem não importa: {2,3}={3,2}

B={ x | x é par}

> Notação Zermelo-Fraenkel, onde o membro após a barra vertical (lida como "tal que") é uma expressão lógica.
> P={x | x é o nome de um rio e x está no Brasil}

Pertinência

$x \in A$

$x \notin A$

No caso anterior,
$1 \in A$
$2 \notin A$

Conjunto infinito:
$C = \{2, 4, 6, 8, \ ...\}$

Conjunto vazio ou nulo

$\emptyset = \{\}$

$B = \{x | x^2 = 4 \text{ e x é impar}\} = \emptyset$

$x \notin \emptyset, \forall x$

> Qualquer que seja x, x não pertence ao conjunto vazio

Universo de discurso

U é o conjunto de todos os elementos do discurso.
$x \in U, \forall x$

Subconjuntos

A={1,3,7}

São subconjuntos de A: $\emptyset$, {1}, {3}, {7}, {1,3}, {1,7}, {3,7}, {1,3,7} (que é o próprio A)

$\{1, 3\} \subset A$

> está contido em A (é subconjunto de A)

$\{1, 7\} \subsetneq A$

> é subconjunto próprio de A, isto é, é subconjunto de A, mas é diferente de A

$A \subsetneq A$ é falso.

Inclusão e Comparabilidade

$\emptyset \subset A, \forall \text{ conjunto } A$

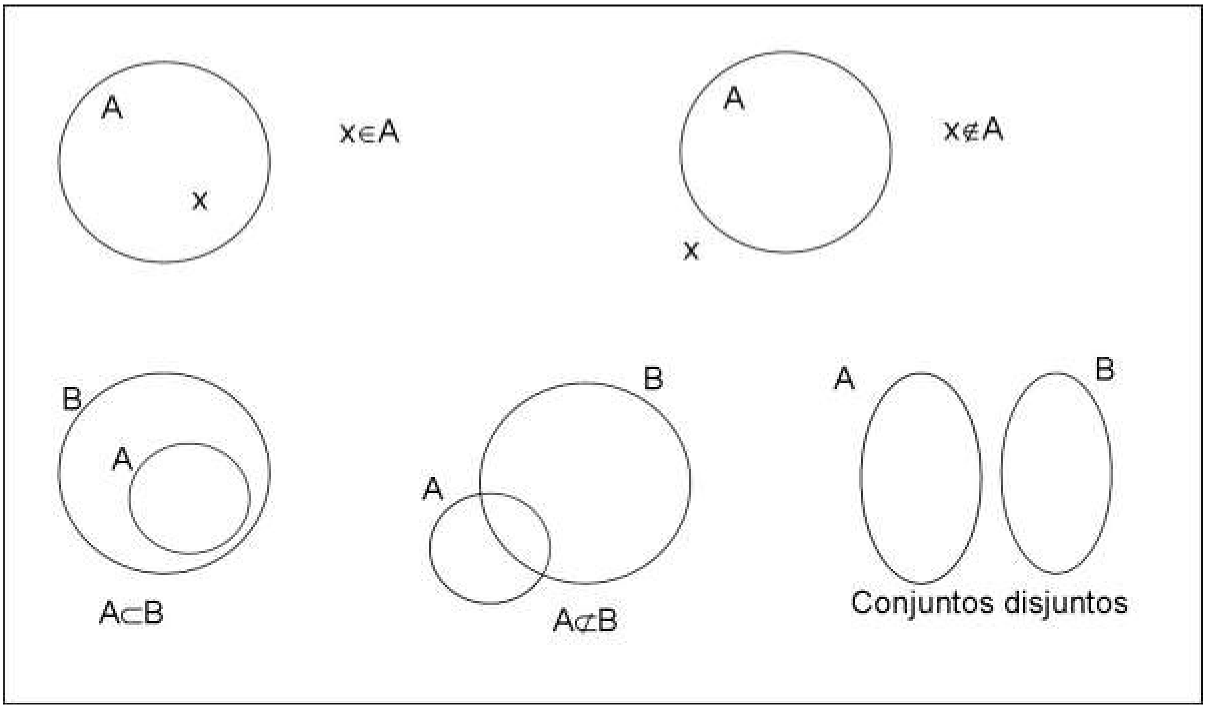
$A \subset B \Rightarrow B \supset A$

> A está contido em B, então B contém A

Se $A \subset B$ ou $B \subset A$, então A e B são comparáveis.

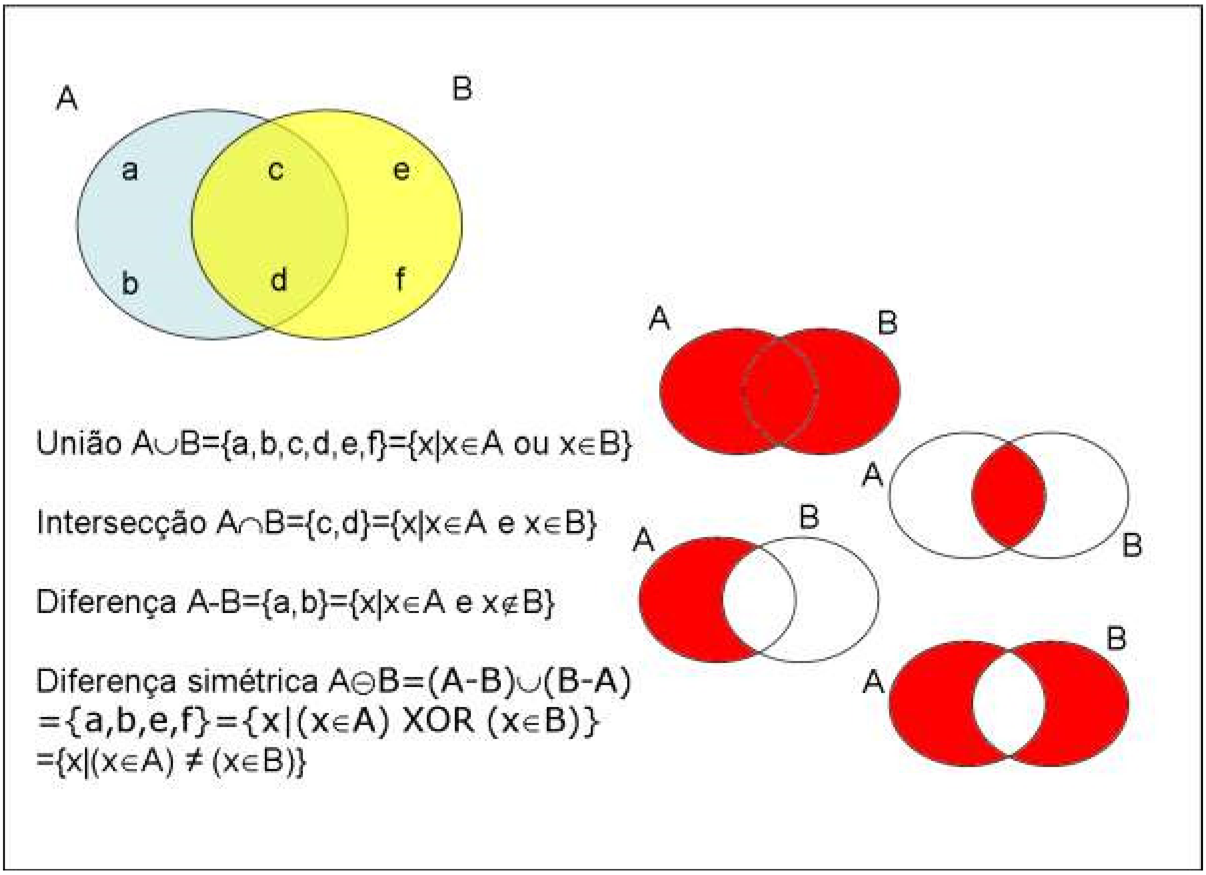
Se $A \not\subset B$ e $B \not\subset A$, então A e B não são comparáveis.

Diagramas de *Venn-Euler*



Operações sobre conjuntos

Considerar os conjuntos e elementos abaixo:
A={a,b,c,d} B={c,d,e,f}



Complemento

Complemento de A é $\overline{A} = \{x | x \notin A\}$

Complemento relativo de A com relação a B
$\overline{A}^B = B - A$

Álgebra de Boole

Valores: **verdadeiro** e **falso**

Proposição: uma afirmação que pode assumir um desses valores.

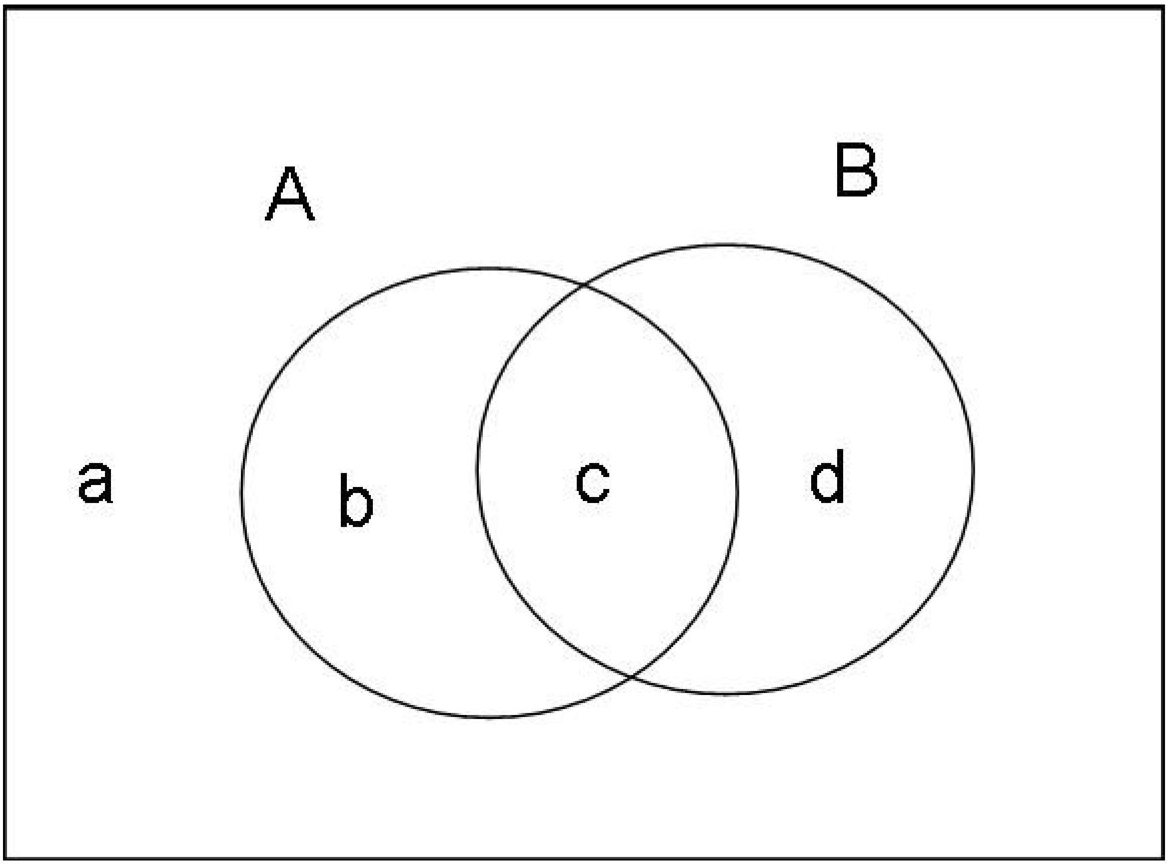
Conjunção

$p \wedge q$ é uma conjunção (operação E)

Tabela verdade:

p	q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Exemplo



$p = a \in A$ falso

$q = a \in B$ falso

$p \wedge q$ falso

Disjunção

$p \vee q$ é uma disjunção (operação OU)

Tabela verdade:

p	q	$p \vee q$
V	V	V
V	F	V
F	V	V
F	F	F

Negação

$\neg p$ é uma negação, uma afirmação de que p é falsa.

Tabela Verdade:

p	$\neg p$
V	F
F	V

Implicação ou condicional

$p \rightarrow q$, p implica q, é o mesmo que $q \vee \neg p$

Se p for verdadeira, q necessariamente deve ser verdadeira.
Se p for falsa, não importa o valor de q.

> Exercício:
>
> Fazer a tabela verdade da implicação.
>
> Fazer o diagrama de Venn para $a \in A \rightarrow a \in B$ e explicar a relação entre os conjuntos A e B.

Bicondicional

$p \leftrightarrow q$, indica que a implicação é válida em ambos sentidos.

Se p for verdadeira, então q é verdadeira.
Se q for verdadeira, então p é verdadeira.

> Exercício:
>
> Fazer a tabela verdade da implicação.
>
> Fazer o diagrama de Venn para $a \in A \leftrightarrow a \in B$ e explicar a relação entre os conjuntos A e B.

Leis de DeMorgan

$\neg(p \vee q) = \neg p \wedge \neg q$

$\neg(p \wedge q) = \neg p \vee \neg q$

$\overline{A \cup B} = \overline{A} \cap \overline{B}$

$\overline{A \cap B} = \overline{A} \cup \overline{B}$

> Exercício:
>
> Verificar a validade das leis de De Morgan através de tabela-verdade e de diagrama de Venn

Quantificadores

p(x) é uma sentença aberta dependente de x.

Exemplo: $p(x) = x \in A$

Se $A = \{1, 2, 3\}$, p(1) é verdadeira e p(4) é falsa.

Quantificador universal $\forall$ (qualquer)

$\forall\ x \in B,\ p(x)$
é verdadeira se $B \subset A$

Quantificar existencial $\exists$ (existe)

$\exists\ x \in B,\ p(x)$
é verdadeira se A e B não são disjuntos.

Construção de conjuntos com quantificadores

Exemplo
$B = \{3, 5, 7\}$

$A = \{x | \exists\ y \in B,\ y < x \text{ e } x < 10\} = \{4, 5, 6, 7, 8, 9\}$
$C = \{x | \forall\ y \in B,\ y < x \text{ e } x < 10\} = \{8, 9\}$

Cardinalidade de um conjunto finito

é o número de elementos de um conjunto
$A = \{2, 3, 7, 8\}$
$|A| = 4$

Conjunto das partes

2^A é o conjunto de todos os subconjuntos de A.

$A = \{1, 2\}$
$2^A = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$

$|2^A| = 2^{|A|}$

> Demonstração:
>
> Associe a cada elemento a_i de A uma proposição $p_i(x)$:
> $p_i(x) = a_i \in x$ que pode ser verdadeira ou falsa.
>
> Para A={1,2,3}
> | $1 \in X$ | $2 \in X$ | $3 \in X$ | conjunto |
> | V | V | V | {1,2,3} |
> | V | V | F | {1,2} |
> | V | F | V | {1,3} |
> | V | F | F | {1} |
> | F | V | V | {2,3} |
> | F | V | F | {2} |
> | F | F | V | {3} |
> | F | F | F | emptyset |

Pares ordenados

$(a, b) = \{\{a\}, \{a, b\}\}$

A ordem do par importa.
$(a, b) = (c, d) \text{ sse } a = c \text{ e } b = d$

Produto cartesiano

$A \times B = \{(a, b) | a \in A \wedge b \in B\}$

Exemplo:
A={1,2,3}
B={x,y}

$A \times B = \{(1, x), (1, y), (2, x), (2, y), (3, x), (3, y)\}$

> $|A \times B| = ?$

$A \times B$ não é necessariamente igual a $B \times A$

> Quando são iguais?

Tuplas

(a,b,c)=((a,b),c)

Assim $(a, b, c) = (d, e, f) \Leftrightarrow a = d,\ b = e \text{ e } c = f$

Produto cartesiano simultâneo de vários conjuntos $A \times B \times C$

Uma relação nos conjuntos $A_1, A_2, ..., A_n$ é um conjunto qualquer de tuplas de elementos de $A_1, A_2, ..., A_n$.

Assim,

$R \subset A_1 \times A_2 \times ... \times A_n = \prod_{i=1}^{n} A_i$

Quando n=2, a relação é binária.

É comum haver relações binárias sobre o mesmo conjunto A.

$R \subset A \times A$

Exemplo

A relação < no conjunto {1,2,3}

$R_1 = \{(1, 2), (1, 3), (2, 3)\}$

Dizemos que $1R_1 2$ é verdadeira e $2R_1 1$ é falsa.

A relação $\leq$ no mesmo conjunto
$R_2 = \{(1, 2), (1, 3), (2, 3), (1, 1), (2, 2), (3, 3)\}$

A relação "é o dobro de" no mesmo conjunto

$R_3 = \{(2, 1)\}$

Relação vazia
$R_4 = \emptyset$

Relação conversa

R^C é o converso de uma relação R.

$R^C = \{(x, y) | (y, x) \in R\}$

> Qual o converso de "<"?

Funções

Uma relação f em $A \times B$ é uma função com domínio A e codomínio B (ou contra-domínio) se para cada $x \in A$ existe um único y em B tl que $(x,\ y) \in f$. Em outros termos,

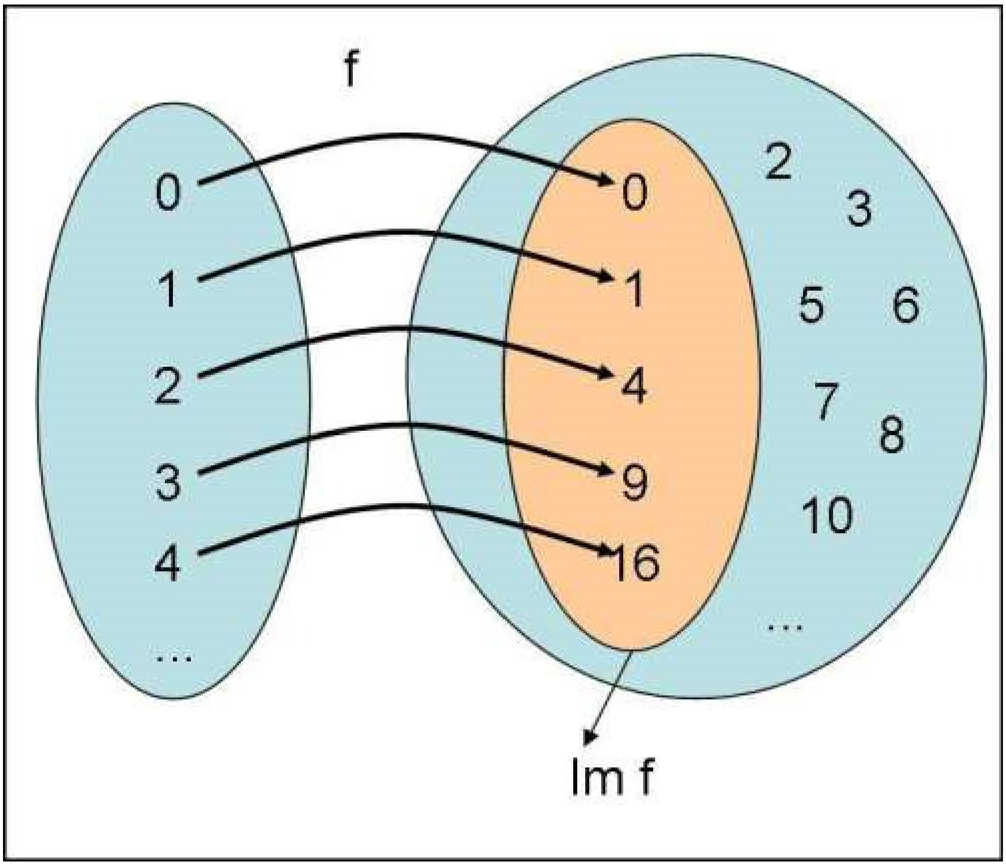
$(x,\ y) \in f$ e $(x,\ z) \in f \rightarrow y = z$

Imagem ou amplitude de f:
$Im f = \{y \in B | \exists\ x \in A,\ f(x) = y\}$

Exemplo: função quadrado

$f : \mathbb{N} \rightarrow \mathbb{N}$
$x \mapsto x^2$

ou $f(x) = x^2$



> qual a cardinalidade de uma função?
>
> > = a cardinalidade do domínio

Composição de relações

$R_1 \subset X \times Z,\ R_2 \subset Z \times Y$

$x(R_1 R_2)y \Leftrightarrow \exists\ z \in Z,\ xR_1 z \wedge zR_2 y$

Correspondências

Dados dois conjuntos A e B e uma função $f : A \rightarrow B$

- f é injetora, biunívoca ou "1 para 1" se nunca mapeia elementos diferentes para o mesmo lugar, ou seja:

$a \neq qb \rightarrow f(a) \neq qf(b)$

- f é sobrejetora, "sobre" ou "onto" se atinge todos os elementos de B, ou seja:

$\forall\ b \in B,\ \exists\ a \in A | f(a) = b$
em outras palavras, $Im f = B$

- f é bijetora ou uma correspondência se for injetora e sobrejetora

Uma bijeção num conjunto finito pode ser asociada a uma permutação ou a uma rerrotulação.

> Exercício:
>
> Proponha contra-exemplos de funções injetoras e sobrejetoras.

Função Inversa

Seja $f : a \rightarrow B$

Se f é injetora, contém uma inversa f^{-1}.

Se f for também sobrejetora, o domínio de f^{-1} é B.

Se f^C for uma função, existe a inversa f^{-1} e $f^{-1} = f^C$