Aula 00 – Introdução

Prof. Eduardo Zambon

Departamento de Informática (DI)
Centro Tecnológico (CT)
Universidade Federal do Espírito Santo (Ufes)

Algoritmos e Fundamentos da Teoria de Computação (ToCE)
Engenharia de Computação

Introdução

- Teoria de conjuntos e matemática discreta são a fundação matemática para:
  - Teoria de linguagens formais;
  - Teoria da computação; e
  - Análise de complexidade computacional.
- **Estes *slides***: revisão/introdução de conceitos básicos de teoria de conjuntos e matemática discreta.
- **Objetivos**: fixar termos, definições e notações para o restante do curso.

Referências

Chapter 1 – Mathematical Preliminaries
T. Sudkamp
Chapter 0 – Introduction
M. Sipser
Chapter 1 – Introduction
A. Maheshwari

- Este curso foca em três áreas centrais da Teoria da Computação:
  - Máquinas abstratas.
  - Computabilidade.
  - Complexidade.
- Essas três áreas estão ligadas pela questão essencial:

Quais são as capacidades e limitações fundamentais dos computadores?

- Essa questão surgiu na década de 1930, quando os logicistas começaram a explorar o significado da computação.
- Com o surgimento de máquinas funcionais a partir da década de 1940, essa pergunta deixou o campo da teoria pura e passou a ter implicações práticas.

Teoria de Complexidade

- A Teoria de Complexidade busca investigar a dificuldade inerente de se resolver diferentes problemas.
- Nessa área, a questão central anterior fica:

O quê torna alguns problemas computacionalmente difíceis e outros fáceis?

- O mais interessante é que ainda não sabemos a resposta, apesar de intensas pesquisas nos últimos 50 anos!
- Felizmente, temos pelo menos um sistema de classificação da dificuldade dos problemas.
- O objetivo fundamental da Teoria de Complexidade é separar os problemas em tratáveis e intratáveis.
- Tratabilidade está associada à dificuldade de se resolver um problema computacionalmente.

Teoria de Computabilidade

- A Teoria de Computabilidade busca investigar os limites das máquinas abstratas de computação.
- Na primeira metade do século XX, matemáticos como Kurt Gödel, Alan Turing e Alonzo Church descobriram que certos problemas básicos não podem ser resolvidos por computadores.
- Pode parecer incrível, mas algo quase banal como determinar se uma declaração matemática é falsa ou não, é um problema indecidível para computadores.
- O objetivo fundamental da Teoria de Computabilidade é separar os problemas em decidíveis e indecidíveis.
- O estudo de Computabilidade é algo fascinante para os que possuem uma inclinação matemática...

Teoria de Máquinas e Linguagens Abstratas

- A Teoria de Máquinas Abstratas lida com as definições e propriedades de modelos matemáticos de computação.
- Esses modelos possuem inúmeras aplicações em diferentes áreas da Ciência da Computação.
- Por exemplo, os Autômatos Finitos são utilizados em processamento de textos, compiladores e projetos de *hardware*.
- Já os ditos Autômatos de Pilha são empregados em análise sintática de linguagens de programação e inteligência artificial.
- As relações entre os diferentes tipos de máquinas e linguagens abstratas estão condensadas na Hierarquia de Chomsky (que será vista no Módulo 03).

- Vamos agora começar pelo estudo de conceitos básicos matemáticos que são necessários ao longo do curso.

Suponha a existência de conjuntos.

- Georg Cantor (1874): *"On a Property of the Collection of All Real Algebraic Numbers"*.
- Gottlob Frege (1879): *"Begriffsschrift: a formula language, modeled on that of arithmetic, of pure thought"*.

Notação:

- Conjuntos são representados por letras maiúsculas:

$$\mathrm{X}, \mathrm{Y} \text{ e } \mathrm{Z}.$$

- Elementos de conjuntos são escritos em *itálico*:

$$a, b, A, B, aaaa \text{ e } abc.$$

Pertinência:

- $x \in \mathrm{X}$: elemento x **pertence** ao conjunto X.
- $x \notin \mathrm{X}$: elemento x **não pertence** ao conjunto X.

Teoria de Conjuntos

- Conjuntos com um número pequeno de elementos podem ser definidos explicitamente:

$$\begin{aligned} X &= \{1, 2, 3\} \\ Y &= \{a, b, c, d, e\}. \end{aligned}$$

- Conjuntos infinitos ou finitos mas com um número grande de elementos devem ser definidos implicitamente (*set comprehension*):

$$\{n \mid n > 9\}.$$

  Acima: conjunto dos números maiores que 9.
- Leia a barra vertical | como "tal que".

Teoria de Conjuntos

- $\mathbf{N} = \{0, 1, 2, \ldots\}$: conjunto dos **números naturais**.
- $\emptyset = \{\}$: conjunto **vazio**.
- Um conjunto é completamente determinado pelos seus elementos.
- A ordem e a quantidade dos elementos é irrelevante.
- *Exemplo* – as três definições abaixo descrevem o mesmo conjunto:

$$X = \{1, 2, 3\}, \quad Y = \{2, 1, 3\}, \quad Z = \{1, 3, 2, 2, 2\}.$$

- Igualdade de conjuntos requer que os conjuntos tenham exatamente os mesmo elementos, como acima.

Teoria de Conjuntos – Subconjuntos

- $Y \subseteq X$: Y é um **subconjunto** de X.
  - Requer que todo elemento de Y também pertença a X.
  - Inverso não necessariamente é verdadeiro! (Relação não-simétrica.)
- $\emptyset$ é subconjunto de qualquer outro.
- Qualquer conjunto é subconjunto dele mesmo.
- Se $Y \subseteq X$ e $Y \neq X$, Y é um **subconjunto próprio** de X. *Notação*: $Y \subset X$.
- Conjuntos X e Y são iguais se $X \subseteq Y$ e $Y \subseteq X$. Muitas vezes é necessário provar as inclusões separadamente.

Exemplo – Provar que os conjuntos abaixo são iguais:

$$\begin{aligned} \mathrm{X} &= \{n \mid n = m^2 \text{ para algum número natural } m > 0\} \\ \mathrm{Y} &= \{n^2 + 2n + 1 \mid n \geq 0\}. \end{aligned}$$

Prova de $X \subseteq Y$:

- Seja $x \in X$, então $x = m^2$ para algum número natural $m > 0$ arbitrário. Tome m_0 como esse número. Então podemos reescrever

$$
\begin{aligned}
x &= (m_0)^2 \\
&= (m_0 - 1 + 1)^2 \\
&= (m_0 - 1)^2 + 2(m_0 - 1) + 1.
\end{aligned}
$$

- Fazendo $n = m_0 - 1$, vemos que $x = n^2 + 2n + 1$ com $n \geq 0$.
- Portanto, $x \in Y$.

Prova de Y $\subseteq$ X:

- Seja $y \in$ Y, então $y = n^2 + 2n + 1$ para algum número natural $n \geq 0$ arbitrário. Tome n_0 como esse número.
- Fatorando a expressão temos

$$y = (n_0)^2 + 2n_0 + 1 = (n_0 + 1)^2.$$

- Logo, y é o quadrado de um número natural positivo e portanto $y \in$ X.

$\Rightarrow$ Como X $\subseteq$ Y e Y $\subseteq$ X, concluímos que X $=$ Y.

Teoria de Conjuntos – Conjunto Potência

- $\mathcal{P}(X)$: **conjunto potência** (*power set*) de X.
- $\mathcal{P}(X)$ é um conjunto cujos elementos são todos os subconjuntos de X.
- *Exemplo* – seja $X = \{1, 2, 3\}$. O conjunto $\mathcal{P}(X)$ é formado pelos elementos abaixo:

$$\emptyset \quad \{1\} \quad \{2\} \quad \{3\} \quad \{1,2\} \quad \{1,3\} \quad \{2,3\} \quad \{1,2,3\}.$$

- União: $X \cup Y = \{z \mid z \in X \text{ ou } z \in Y\}$.
- Interseção: $X \cap Y = \{z \mid z \in X \text{ e } z \in Y\}$.
- Conjuntos com interseção vazia são ditos disjuntos.
- Operações para n conjuntos:

$$\bigcup_{i=1}^{n} X_i = X_1 \cup X_2 \cup \cdots \cup X_n$$

$$\bigcap_{i=1}^{n} X_i = X_1 \cap X_2 \cap \cdots \cap X_n$$

- Diferença: $X - Y = \{z \mid z \in X \text{ e } z \notin Y\}$.

Teoria de Conjuntos – Partição e Complemento

- Partição de um conjunto X: subconjuntos $X_1, X_2, \ldots, X_n$ satisfazendo as condições
  1. $X = \bigcup_{i=1}^{n} X_i$
  2. $X_i \cap X_j = \emptyset$, para $1 \leq i, j \leq n$, e $i \neq j$.
- *Exemplo* – indique uma partição do conjunto **N**?
- $\Rightarrow$ Divisão dos números naturais entre pares e ímpares (subconjuntos disjuntos).

- Complemento de X em relação um conjunto universal U:

$$\overline{X} = U - X.$$

- *DeMorgan*:

$$\overline{(X \cup Y)} = \overline{X} \cap \overline{Y} \qquad \overline{(X \cap Y)} = \overline{X} \cup \overline{Y}.$$

Teoria de Conjuntos – Exemplo

Sejam dois conjuntos $X = \{0, 1, 2, 3\}$ e $Y = \{2, 3, 4, 5\}$, e tome $\overline{X}$ e $\overline{Y}$ como complementos com relação ao conjunto **N**. Faça:

$X \cup Y =$

$X \cap Y =$

$X - Y =$

$Y - X =$

$\overline{X} =$

$\overline{Y} =$

$\overline{X} \cap \overline{Y} =$

$\overline{(X \cup Y)} =$

Sejam dois conjuntos $X = \{0, 1, 2, 3\}$ e $Y = \{2, 3, 4, 5\}$, e tome $\overline{X}$ e $\overline{Y}$ como complementos com relação ao conjunto **N**. Faça:

$$X \cup Y = \{0, 1, 2, 3, 4, 5\} \qquad \overline{X} = \{n \mid n > 3\}$$

$$X \cap Y = \{2, 3\} \qquad \overline{Y} = \{0, 1\} \cup \{n \mid n > 5\}$$

$$X - Y = \{0, 1\} \qquad \overline{X} \cap \overline{Y} = \{n \mid n > 5\}$$

$$Y - X = \{4, 5\} \qquad \overline{(X \cup Y)} = \{n \mid n > 5\}$$

Produto Cartesiano

- Produto Cartesiano – operação que constrói um conjunto de pares ordenados:

$$\mathrm{X} \times \mathrm{Y} = \{[x, y] \mid x \in \mathrm{X} \text{ e } y \in \mathrm{Y}\}.$$

- *Obs.:* Vamos usar a notação do livro do Sudkamp para tuplas: $[x, y]$. Notações alternativas incluem (x, y) e $\langle x, y \rangle$.

- Relação binária entre X e Y: um subconjunto de X $\times$ Y.
- *Exemplo*:

$$\mathrm{LT} = \{[i, j] \mid i < j \text{ e } i, j \in \mathbf{N}\}.$$

- A notação $[i, j] \in \mathrm{LT}$ indica que i é menor que j.
- O produto Cartesiano pode ser generalizado para n conjuntos.

Produto Cartesiano – Exemplo

Sejam $X = \{1, 2, 3\}$ e $Y = \{a, b\}$. Determine:

a. $X \times Y =$

b. $Y \times X =$

c. $Y \times Y =$

d. $X \times Y \times Y =$

Produto Cartesiano – Exemplo

Sejam $X = \{1, 2, 3\}$ e $Y = \{a, b\}$. Determine:

a. $X \times Y = \{[1, a], [1, b], [2, a], [2, b], [3, a], [3, b]\}$

b. $Y \times X = \{[a, 1], [a, 2], [a, 3], [b, 1], [b, 2], [b, 3]\}$

c. $Y \times Y = \{[a, a], [a, b], [b, a], [b, b]\}$

d. $$X \times Y \times Y = \{[1, a, a], [1, a, b], [1, b, a], [1, b, b], [2, a, a], [2, a, b], [2, b, a], [2, b, b], [3, a, a], [3, a, b], [3, b, a], [3, b, b]\}$$

Funções

- Uma função f de um conjunto X para um conjunto Y mapeia cada um dos elementos de X em no máximo um elemento de Y.
- *Notação*: $f : X \to Y$.
- Para $x \in X$, $f(x) \in Y$ é o elemento mapeado por f.
- Domínio (*domain*): é o conjunto X.
- Co-domínio (*codomain*, contra-domínio): é o conjunto Y.
- Imagem (*range*): é um subconjunto de Y tal que

$$\{y \in Y \mid y = f(x) \text{ para algum } x \in X\}.$$

- *Exemplo* – função idade: mapeia um conjunto de pessoas em um número natural.

Funções

- O domínio de uma função de n variáveis é um produto Cartesiano:

$$f : \mathrm{X}_1 \times \mathrm{X}_2 \times \cdots \times \mathrm{X}_n \to \mathrm{Y}.$$

- Função total $f : \mathrm{X} \to \mathrm{Y}$ é uma relação binária sobre $\mathrm{X} \times \mathrm{Y}$ que satisfaz
  1. Para cada $x \in \mathrm{X}$, existe um $y \in \mathrm{Y}$ tal que $[x, y] \in f$.
  2. Se $[x, y_1] \in f$ e $[x, y_2] \in f$, então $y_1 = y_2$.
- Função parcial: satisfaz somente a condição 2 acima.
- $\Rightarrow$ Não está definida para todos os elementos do domínio. *Exemplo*: função de divisão.
- *Notação*: $f : \mathrm{X} \rightharpoonup \mathrm{Y}$.
- $f(x)\uparrow$ indica que f não é definida para x.
- $f(x)\downarrow$ indica que f é definida mas o valor não importa.

Funções – Propriedades

Seja $f : X \to Y$ uma função total.

- Função injetiva (*injective*, injetora):

$$x_1 \neq x_2 \Rightarrow f(x_1) \neq f(x_2)$$

  isto é, f mapeia cada elemento de X em um elemento distinto da imagem.

- Função surjetiva (*surjective, onto*, sobrejetora):

  imagem de f é o conjunto Y.

- Função bijetiva (*bijective*, bijetora): injetiva e surjetiva ao mesmo tempo.
- Estabelece uma correspondência entre elementos do domínio e co-domínio: relação de um-para-um.

Funções – Exemplos

As funções *f*, *g* e *s* são definidas de **N** para **N** $- \{0\}$, o conjunto dos naturais positivos.

a. $f(n) = 2n + 1$
b. $g(n) = \begin{cases} 1 & \text{se } n = 0 \\ n & \text{caso contrário} \end{cases}$
c. $s(n) = n + 1$

- Função f é injetiva mas não é surjetiva; imagem de f é só os números ímpares.
- Função g é surjetiva mas não é injetiva; $g(0) = g(1) = 1$.
- Função s é bijetiva; calcula o sucessor de um número.

Conjuntos Contáveis e Incontáveis

- Cardinalidade é o número de elementos de um conjunto.
- OK para conjuntos finitos, mas e para os infinitos?
- Contagem indireta através de uma relação de um-para-um com outro conjunto.

Definição 1.4.1 (Sudkamp)

a. Dois conjuntos X e Y têm a mesma cardinalidade se existe uma função bijetiva $f : X \to Y$.
b. A cardinalidade do conjunto X é menor ou igual que a de Y se existe uma função injetiva $f : X \to Y$.

- Importante notar a diferença entre itens a) e b).
- *Notação*: *card*(X).

Conjuntos Contáveis e Incontáveis

- Conjunto infinito contável ou enumerável: mesma cardinalidade de $\mathbf{N}$.
- Intuição: conjunto é enumerável se os elementos podem ser ordenados e contados.
- Formalmente: definir bijeção f que estabelece correspondência com os números naturais.
- Conjunto contável: conjunto **finito** ou **enumerável**.
- Conjunto incontável: conjunto que não é *contável*.

Conjuntos Contáveis e Incontáveis

Exemplo

- Conjunto $\mathbf{N} - \{0\}$ é enumerável.
- Função sucessor $s(n) = n + 1$ define uma bijeção.
- Contra-intuitivo: tirar um elemento de **N** mantém o mesmo número de elementos!
- $card(\mathbf{N}) = card(\mathbf{N} - \{0\}) = \aleph_0$ (aleph zero – Cantor).

O exemplo acima na verdade leva à definição formal de um conjunto infinito.

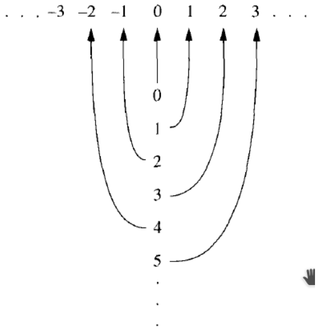
Definição 1.4.3 (Sudkamp)

Um conjunto é infinito se ele possui um subconjunto próprio com a mesma cardinalidade.

Conjuntos Contáveis e Incontáveis

Exemplo 1.4.1 (Sudkamp)

- Conjunto dos números naturais ímpares é enumerável.
- Função $f(n) = 2n + 1$ define uma bijeção entre $\mathbf{N}$ e os ímpares.



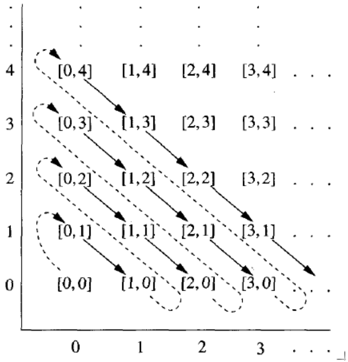
Exemplo

Há uma bijeção entre $\mathbf{N}$ e o conjunto dos números inteiros $\mathbf{Z}$.

$$f(n) = \begin{cases} div(n,2) + 1 & n \text{ ímpar} \\ -div(n,2) & n \text{ par.} \end{cases}$$

Exemplo 1.4.2 (Sudkamp)

- Conjunto dos pares ordenados de números naturais $\mathbf{N} \times \mathbf{N}$ é enumerável.
- Mapear $[i, j]$ no número natural $((i + j) \cdot (i + j + 1)/2) + i$.



Conjuntos Contáveis e Incontáveis

- Praticamente todos os conjuntos de interesse em computabilidade são contáveis.
- Algumas propriedades interessantes:

Teorema 1.4.4 (Sudkamp)

a. A união de dois conjuntos contáveis é contável.
b. O produto Cartesiano de dois conjuntos contáveis é contável.
c. O conjunto de subconjuntos finitos de um conjunto contável é contável.

Conjuntos Contáveis e Incontáveis

- Um conjunto é incontável se é impossível listar os seus elementos de forma sequencial.
- *Exemplo* – conjunto dos números reais **R**.
- Provar que um conjunto é incontável normalmente requer uma técnica especial de prova por contradição: argumento de diagonalização de Cantor.
- *Exemplo* – como mostrar que há um número incontável de funções totais de **N** para **N**?
- Dadas duas funções f e g, temos que $f = g$, se $f(n) = g(n)$ para todo $n \in \mathbf{N}$.
- Para mostrar que f e g são distintas basta achar um n aonde $f(n) \neq g(n)$.

Prova por diagonalização:

- Assuma que o conjunto de funções totais de **N** para **N** é enumerável.
- Então existe uma sequência $f_0, f_1, f_2, \ldots$ que contém todas as funções.
- Os valores das funções formam um grid bidimensional.

	0	1	2	3	4	$\ldots$
f_0	$f_0(0)$	$f_0(1)$	$f_0(2)$	$f_0(3)$	$f_0(4)$	$\ldots$
f_1	$f_1(0)$	$f_1(1)$	$f_1(2)$	$f_1(3)$	$f_1(4)$	$\ldots$
f_2	$f_2(0)$	$f_2(1)$	$f_2(2)$	$f_2(3)$	$f_2(4)$	$\ldots$
f_3	$f_3(0)$	$f_3(1)$	$f_3(2)$	$f_3(3)$	$f_3(4)$	$\ldots$
f_4	$f_4(0)$	$f_4(1)$	$f_4(2)$	$f_4(3)$	$f_4(4)$	$\ldots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Conjuntos Contáveis e Incontáveis

Prova por diagonalização (cont.):

- Defina a função $f : \mathbf{N} \to \mathbf{N}$ como $f(n) = f_n(n) + 1$.
- Valores de f vêm da diagonal do grid.
- Pela definição de f, temos que $f(i) \neq f_i(i)$ para todo i.
- Portanto, f não está na sequência $f_0, f_1, f_2, \ldots$.
- Contradição! Assumiu-se que a sequência tinha todas as funções totais.
- Assumir que o conjunto de funções totais de **N** para **N** é enumerável leva a uma contradição.
- $\Rightarrow$ O conjunto é incontável.

- A prova de que o conjunto dos números reais **R** é incontável é similar.

Diagonalização e Auto-Referência

- Diagonalização é usada em:
  - Provas de cardinalidade (slides anteriores).
  - Demonstrações que certas propriedades e/ou relações são contraditórias.
- Prova por contradição que surge por auto-referência: um objeto analisando suas próprias ações, propriedades ou características.
- *Exemplos*:
  - Paradoxo de Russell.
  - Indecidibilidade do Problema da Parada para Máquinas de Turing.
  - Indecidibilidade da Teoria dos Números (Gödel).

Paradoxo de Russell:

- Investiga uma relação de pertinência entre dois conjuntos.
- Para cada conjunto X, deseja-se saber:
  Um outro conjunto Y é elemento de X?
- Esta é uma questão totalmente razoável, já que um conjunto pode ser elemento de outro.
- Note que a pergunta sendo feita é $Y \in X$ e não $Y \subseteq X$.
- Alguns exemplos:

X	Y	$Y \in X$
$\{a\}$	$\{a\}$	não
$\{\{a\}, b\}$	$\{a\}$	sim
$\{\{a\}, a, \emptyset\}$	$\emptyset$	sim
$\{\{a, b\}, \{a\}\}$	$\{\{a\}\}$	não

Paradoxo de Russell – Formalização:

- Seja a propriedade P: "um conjunto não ser elemento dele mesmo".
- Essa propriedade define um conjunto? Assuma que sim.
- Assim, temos que $S = \{X \mid X \notin X\}$ é um conjunto.
- $S \in S$?
- Se sim, então $S \notin S$ pela definição.
- Se não, então $S \in S$ pela definição.
- Temos uma contradição clara. Onde está o problema?
- Na suposição de que a coleção dos conjuntos que satisfazem P também é um conjunto.

Diagonalização e Auto-Referência

Paradoxo de Russell – Consequências:

- Até o final do século XIX a fundação da matemática era a teoria de conjuntos de Cantor.
- Princípio central dessa fundação: toda propriedade ou condição que pode ser descrita forma um conjunto – o conjunto de objetos que satisfazem a condição.
- O Paradoxo de Russell mostra que esse princípio não vale.
- Descoberta levou ao início da filosofia formalista da matemática.
- Desenvolvimento de sistemas formais de dedução com axiomas e regras de inferência.
- *Exemplo* – Lógica de Primeira Ordem (FOL), Whitehead and Russell, *Principia Mathematica*, 1910.

Definições Recursivas

- A maioria dos conjuntos de interesse em computabilidade são infinitos.
- Precisamos de técnicas para descrever, gerar e reconhecer elementos de um conjunto infinito.
- Anteriormente o conjunto dos naturais foi descrito como $\mathbf{N} = \{0, 1, 2, \ldots\}$.
- Razoável para humanos, mas e para um *alien* que não conhece o nosso sistema decimal?
- Computador = *alien*.
- $\Rightarrow$ Teoremas e definições precisam ser formalizados (axiomatizados); nada pode ficar implícito ou deixado para a intuição.

Definições Recursivas

- Uma definição recursiva de um conjunto X especifica um método para construir (gerar) os elementos de X.
- Base: conjunto finito de elementos que fazem parte de X.
- Operações: usadas para construir novos elementos de X a partir dos elementos anteriormente gerados.
- Conjunto X definido recursivamente: formado por todos os elementos gerados a partir da base por um número *finito* de operações.

Definições Recursivas

Definição 1.6.1 (Sudkamp) – Peano (1889)

Uma definição recursiva de **N**, o conjunto dos números naturais, é construída usando a função sucessor s.

1. **Base:** $0 \in \mathbf{N}$.
2. **Passo recursivo:** se $n \in \mathbf{N}$, então $s(n) \in \mathbf{N}$.
3. **Fecho:** $n \in \mathbf{N}$ somente se n pode ser obtido de 0 por um número finito de aplicações do passo 2.

- Definição acima gera a sequência infinita $0, s(0), s(s(0)), s(s(s(0))), \ldots$.
- Normalmente abreviada por $0, 1, 2, 3, \ldots$.
- Tudo que se faz com numerais Arábicos se faz também com números de Peano.

Definições Recursivas

- Como definir as operações aritméticas sobre os números de Peano?
- Aprendizado usual (força bruta – memorização) não serve para o computador.
- Até agora a função sucessor é a única função sobre **N** que foi introduzida.
- $\Rightarrow$ Uma operação de soma só pode usar 0 e s.

Definições Recursivas

Definição 1.6.2 (Sudkamp) – Soma de Peano

A soma de m e n é definida recursivamente sobre n.

1. **Base:** Se $n = 0$, então $m + n = m$.
2. **Passo recursivo:** $m + s(n) = s(m + n)$.
3. **Fecho:** $m + n = k$ somente se a igualdade pode ser obtida de $m + 0 = m$ por um número finito de aplicações do passo 2.

Exemplo 1.6.1 (Sudkamp)

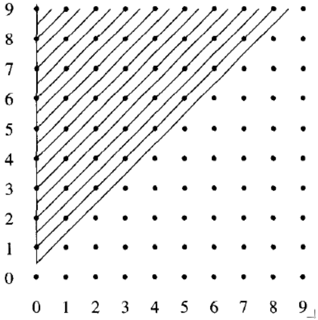
Soma de 3 e 2 (representados por $s(s(s(0)))$ e $s(s(0))$):

$$
\begin{aligned}
& s(s(s(0))) + s(s(0)) = s(s(s(s(0))) + s(0)) \\
& = s(s(s(s(s(0))) + 0)) = s(s(s(s(s(0))))) = 5
\end{aligned}
$$

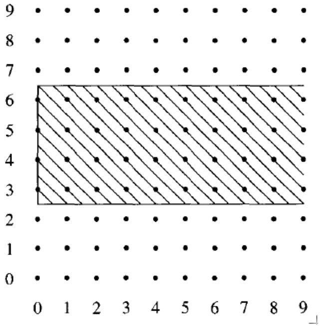
Exemplo 1.6.2 (Sudkamp)

Relação LT (*less than*):

1. **Base:** $[0, 1] \in$ LT.
2. **PR:** Se $[m, n] \in$ LT então $[m, s(n)] \in$ LT e $[s(m), s(n)] \in$ LT.
3. **Fecho.**



Definições Recursivas



Exemplo 1.6.3 (Sudkamp)

Definição recursiva do conjunto X de valores ao lado.

1. **Base:** $[0,3], [0,4], [0,5], [0,6] \in X$.
2. **PR:** Se $[m,n] \in X$ então $[s(m),n] \in X$.
3. **Fecho.**

Indução Matemática

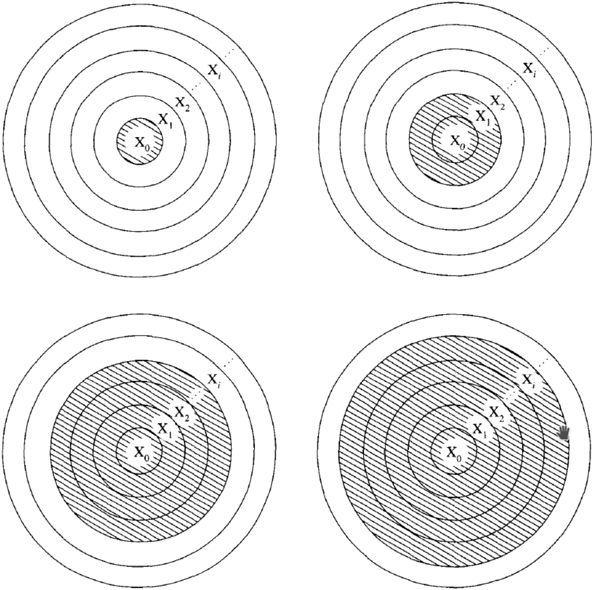
- Como provar que uma propriedade **P** vale para todos os elementos de um conjunto X?
- Se X é finito, OK. Testar **P** para todos os elementos de X.
- Se X é definido recursivamente $\Rightarrow$ indução matemática.

Indução Matemática (Forte)

- Seja X um conjunto definido recursivamente a partir da base X_0 e seja $X_0, X_1, X_2, \ldots$ a sequência de conjuntos gerados pela recursão.
- Seja **P** uma propriedade definida sobre os elementos de X.
- Se é possível provar que:
  1. **P** vale para todo elemento de X_0, e
  2. se **P** vale para todos os elementos de $X_0, X_1, \ldots, X_i$, então **P** também vale para todos os elementos de X_{i+1},

  então **P** vale para todos os elementos de X.

Indução Matemática – Princípio



Indução Matemática – Exemplo

- Provar que $n! > 2^n$, para $n \geq 4$.
- **Caso Base** ($n = 4$): $4! = 24 > 16 = 2^4$.
- **Hipótese Indutiva (HI)**: Assuma que $k! > 2^k$ para $k = 4, 5, \ldots, n$.
- **Passo Indutivo**: Devemos provar que $(n+1)! > 2^{n+1}$.

$$\begin{array}{rcll} (n+1)! & = & n!(n+1) & \\ & > & 2^n(n+1) & \text{(HI)} \\ & > & 2^n 2 & \text{(dado que } n+1 > 2\text{)} \\ & = & 2^{n+1} & \end{array}$$

Aula 00 – Introdução

Prof. Eduardo Zambon

Departamento de Informática (DI)
Centro Tecnológico (CT)
Universidade Federal do Espírito Santo (Ufes)

Algoritmos e Fundamentos da Teoria de Computação (ToCE)
Engenharia de Computação