

# SBVLIFA: Linguagens Formais e Autômatos

Aula 01: Apresentação da Disciplina e Introdução à Teoria da Computação

Bacharelado em Ciência da Computação  
Prof. Dr. David Buzatto

# Apresentação da Disciplina

- ▶ Linguagens Formais e Autômatos;
- ▶ 4 aulas semanais, durante 19 semanas, totalizando 76 aulas semestrais.

# Logística

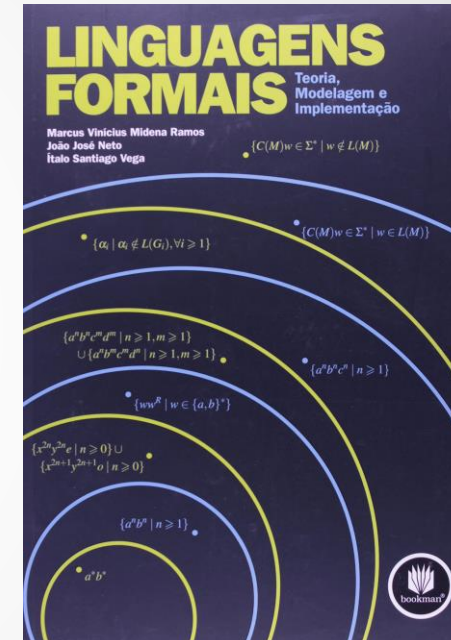
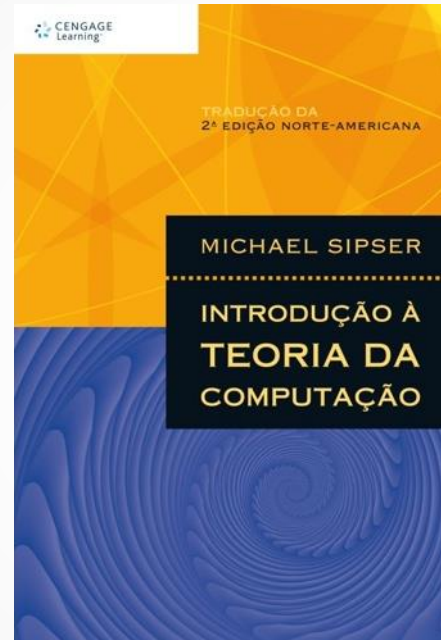
## Critérios de Avaliação

$$M = \begin{cases} \frac{\sum_{i=1}^q \left( \frac{A_i}{2} + \frac{A_i}{2} * \frac{\sum_{j_i=1}^{e_i} E_{j_i}}{10} \right)}{q} + D, & \text{se } p = 0 \\ \frac{\sum_{i=1}^q \left( \frac{A_i}{2} + \frac{A_i}{2} * \frac{\sum_{j_i=1}^{e_i} E_{j_i}}{10} \right)}{q} + \frac{\sum_{i=1}^p P_i}{p} + D, & \text{se } p > 0 \end{cases}$$

Onde:

- $M$ : média final;
- $q$ : quantidade de agrupamentos temáticos;
  - $A_i$ : nota da avaliação diagnóstica de um agrupamento temático  $i$ , sendo que  $A_i = \{x \mid 0 \leq x \leq 10 \wedge x \in \mathbb{Q}\}$ ;
  - $e_i$ : quantidade de listas de exercícios de um agrupamento temático  $i$ ;
    - $E_{j_i}$ : nota da lista de exercícios  $j$  de um agrupamento temático  $i$ , sendo que  $E_{j_i} = \{x \mid 0 \leq x \leq 10 \wedge x \in \mathbb{Q}\}$ ;
- $p$ : quantidade de projetos;
  - $P_i$ : nota do projeto  $i$ , sendo que  $P_i = \{x \mid 0 \leq x \leq 10 \wedge x \in \mathbb{Q}\}$ ;
- $D$ : desafio opcional, onde somente o primeiro a entregar e a acertar ganha meio ponto. Se não acertar, o segundo a entregar é avaliado e assim por diante. Um aluno só pode ganhar uma vez por semestre.

$$, q \in \mathbb{N}^* \wedge p \in \mathbb{N}$$

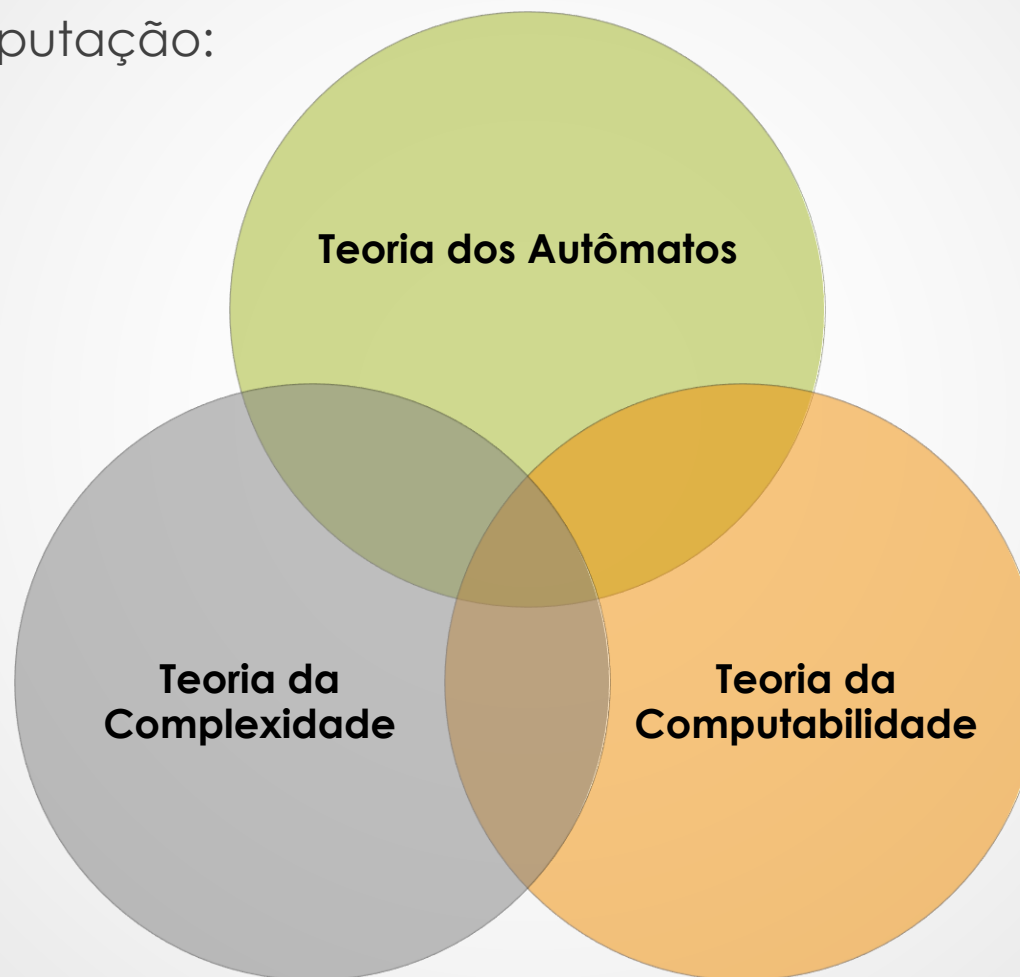


# Apresentação da Disciplina

- ▶ Também conhecida como...:
  - ▶ Teoria de Linguagens;
  - ▶ Aspectos Formais da Computação;
  - ▶ Teoria da Computação.

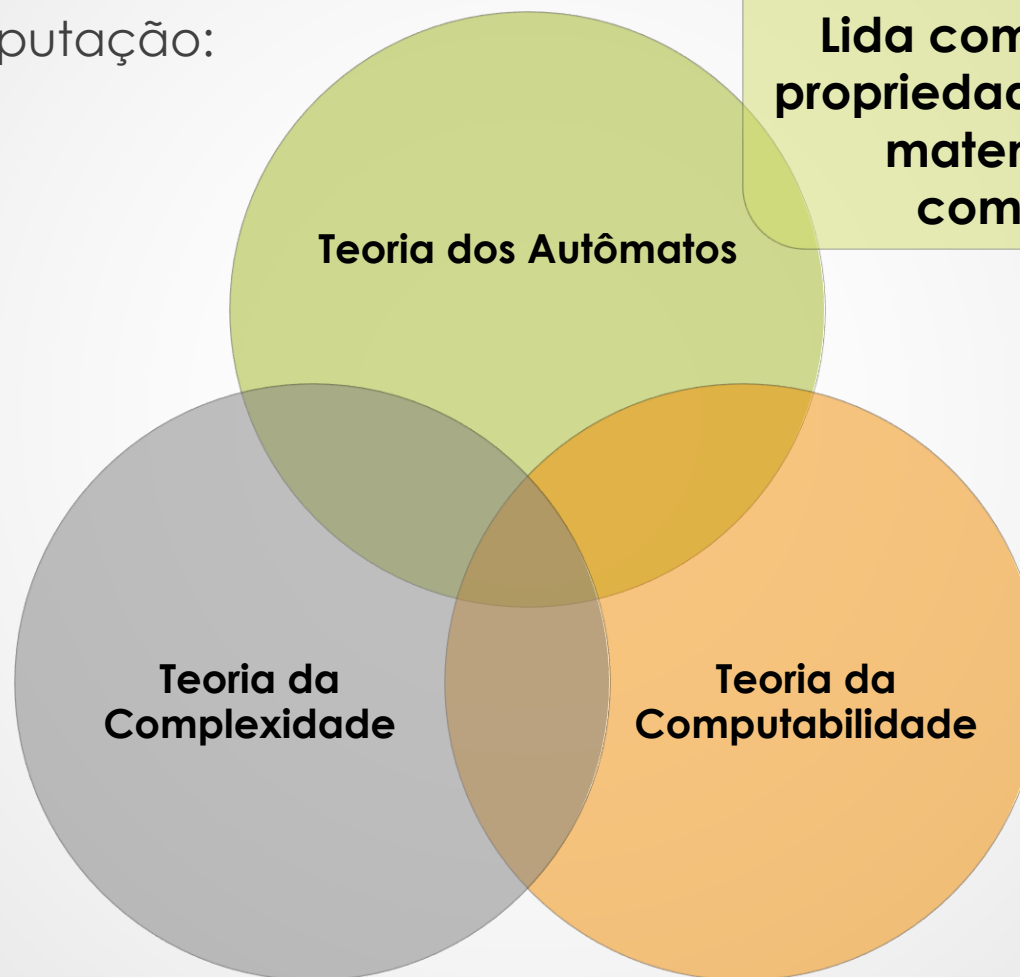
# Apresentação da Disciplina

► Teoria da Computação:



# Apresentação da Disciplina

► Teoria da Computação:

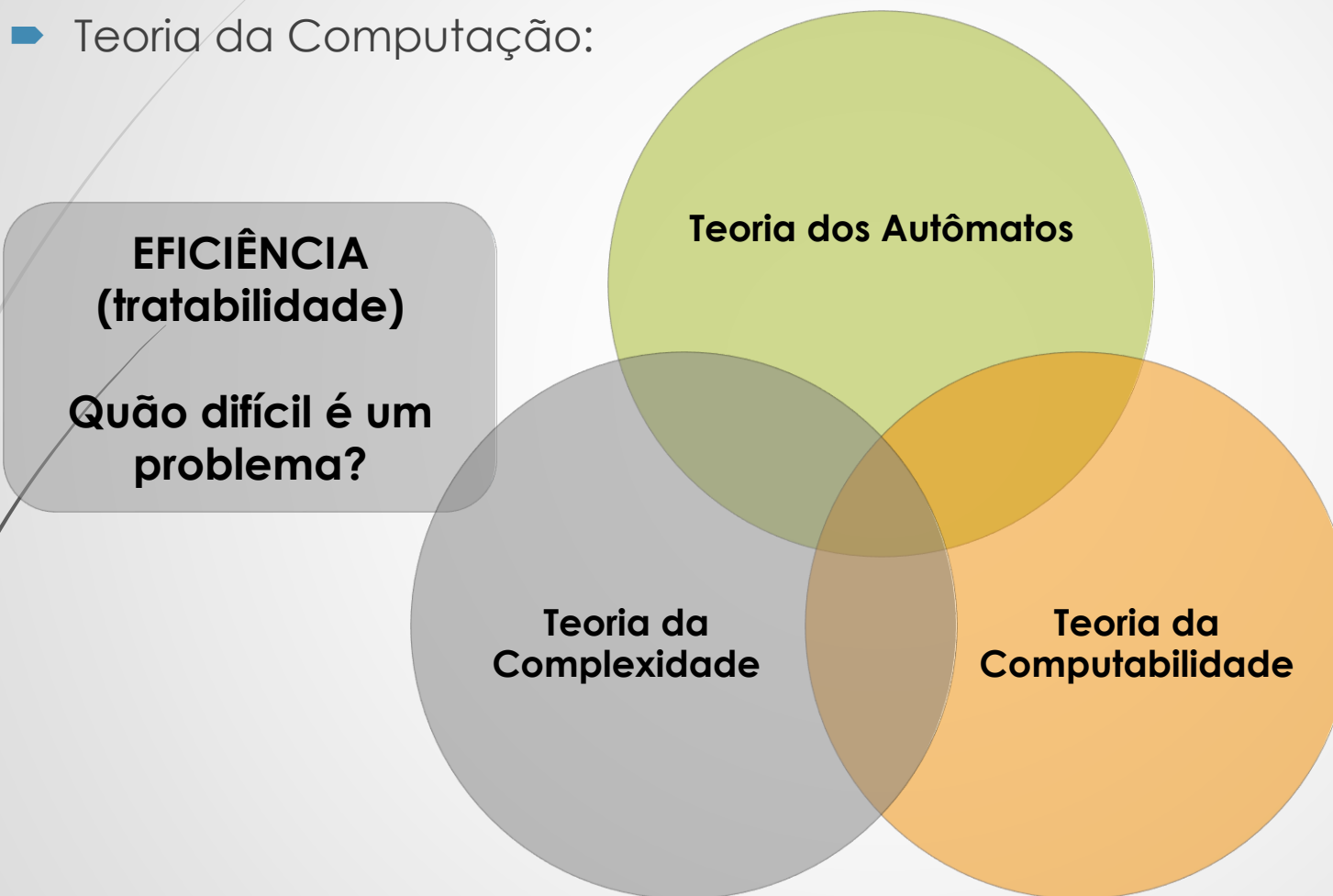


## MODELOS

**Lida com definições e propriedades de modelos matemáticos de computação.**

# Apresentação da Disciplina

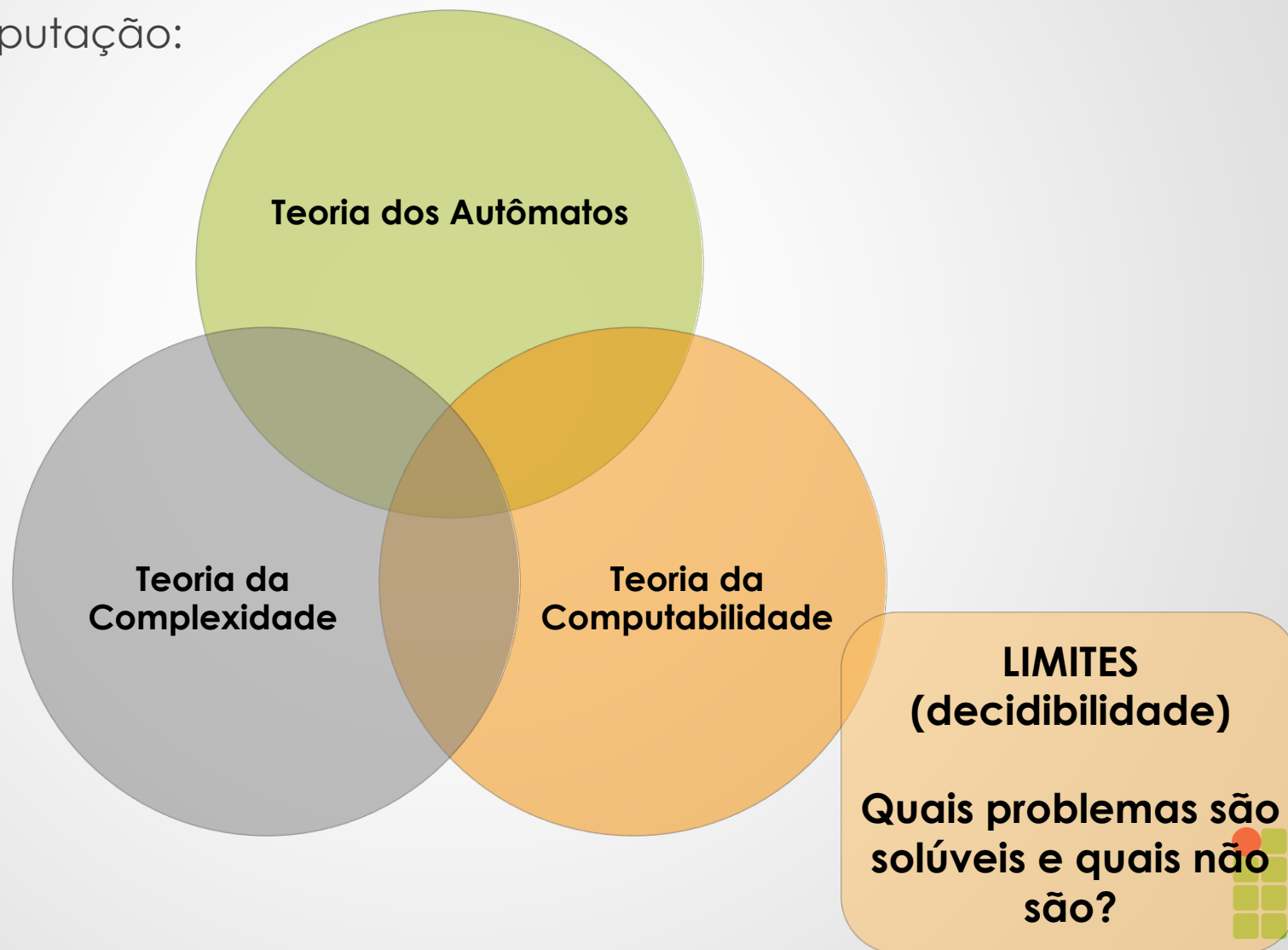
► Teoria da Computação:





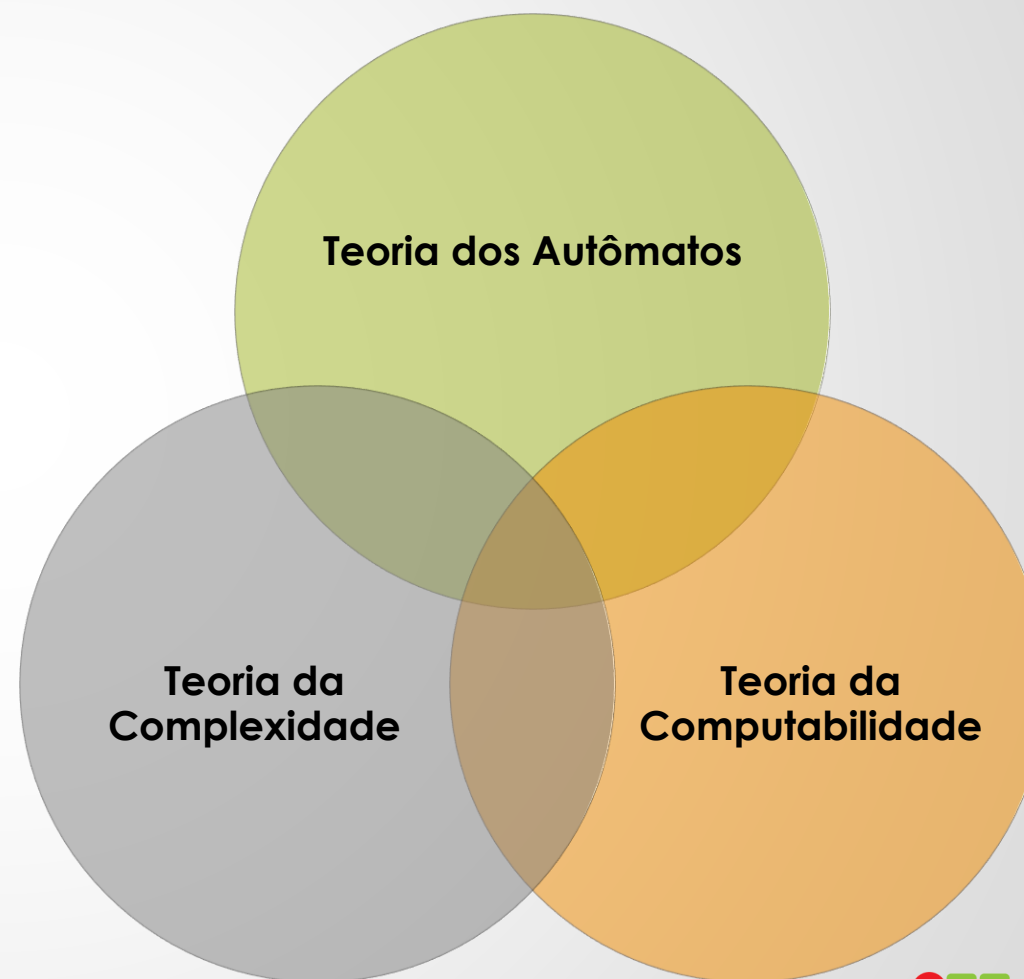
# Apresentação da Disciplina

► Teoria da Computação:



# Apresentação da Disciplina

- ▶ Teoria da Computação:
  - ▶ Ordem que trabalharemos:
    1. Teoria dos Autômatos:
      - ▶ Nosso principal foco;
    2. Teoria da Complexidade:
      - ▶ Já trabalhado em Projeto e Análise de Algoritmos;
    3. Teoria da Computabilidade:
      - ▶ Máquinas de Turing.



# Apresentação da Disciplina

- Conteúdo programático:
  - Conceitos centrais da teoria de autômatos:
    - Alfabetos, strings, linguagens e problemas;
  - Linguagens Regulares:
    - Autômatos finitos, expressões regulares e propriedades;
  - Linguagens Livres de Contexto:
    - Gramáticas livres de contexto, autômatos de pilha e propriedades;
  - Linguagens Sensíveis ao Contexto:
    - Máquinas de Turing com fita limitada;
  - Linguagens Recursivas:
    - Máquinas de Turing que sempre param e extensões;
  - Linguagens Recursivamente Enumeráveis:
    - Decidibilidade, redutibilidade, Máquinas de Turing Universais.

# Antes de Começar...

- O que vamos precisar?
  - Teoria dos conjuntos;
  - Funções e relações;
  - Grafos;
  - Lógica Booleana;
  - Definições:
    - Descrição de objetos e noções;
  - Teoremas:
    - Enunciados matemáticos demonstrados como verdadeiros, geralmente de interesse especial;
  - Lemas:
    - Enunciados que ajudam na prova de um enunciado mais significativos;
  - Corolários:
    - Enunciados tidos como verdadeiros por consequência de um teorema ou sua prova;
- Provas formais.

# Autômatos: os métodos e a loucura

# Conceitos Centrais da Teoria dos Autômatos

# Conceitos Centrais da Teoria dos Autômatos

## ➤ Alfabetos:

- Um conjunto de símbolos finito não-vazio, designados normalmente usando as letras gregas maiúsculas  $\Sigma$  (sigma) e  $\Gamma$  (gama).
  - $\Sigma = \{0, 1\}$ , o alfabeto binário;
  - $\Sigma = \{a, b, c, \dots, z\}$ , o conjunto de todas as letras minúsculas;
  - O conjunto de todos os caracteres UNICODE imprimíveis;

## ➤ Strings/Cadeias:

- Sequência finita de símbolos escolhidos de algum alfabeto.
  - 01011, 000 e 11 são exemplos de strings do alfabeto binário  $\Sigma = \{0, 1\}$
- **String vazia:**
  - É uma string com zero ocorrências de símbolos, denotada pela letra  $\varepsilon$  (épsilon). **Toda linguagem possui essa string!**
- **Comprimento:**
  - Número de símbolos de uma cadeia. Por exemplo, se  $w = 0111$ ,  $|w| = 4$
  - Outros exemplos:  $|cadeia| = 6$ ,  $|\varepsilon| = 0$

# Conceitos Centrais da Teoria dos Autômatos

## ► Strings/Cadeias (continuação):

### ► Potências de um Alfabeto:

- $\Sigma^k$ : conjunto de strings de comprimento  $k$  sobre  $\Sigma$ ;
- Se  $\Sigma = \{0, 1\}$ , então:
  - $\Sigma^0 = \{\varepsilon\}$
  - $\Sigma^1 = \{0, 1\}$  (cuidado,  $\Sigma$  é o alfabeto,  $\Sigma^1$  é o conjunto de strings de comprimento 1 sobre  $\Sigma$ )
  - $\Sigma^2 = \{00, 01, 10, 11\}$
  - $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$
  - $\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \dots\}$
  - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$
  - $\Sigma^+ = \{0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \dots\}$
  - $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \dots$
  - $\Sigma^+ = \Sigma^* - \Sigma^0$

$\Sigma^*$ : sigma estrela ou fecho de sigma.  
**Operador unário  $*$** : fecho de Kleene ou estrela de Kleene ou operador de Kleene.

$\Sigma^+$ : sigma mais ou fecho positivo de sigma.  
**Operador unário  $+$** : soma de Kleene.



# Conceitos Centrais da Teoria dos Autômatos

## ► Strings/Cadeias (continuação):

### ► Concatenação:

- Se  $x$  e  $y$  são strings,  $xy$  é a concatenação de  $x$  e  $y$ :
- Se  $x = a_1a_2 \dots a_i$  e  $y = b_1b_2 \dots b_j$ ,  $xy = a_1a_2 \dots a_ib_1b_2 \dots b_j$
- Se  $|x| = i$  e  $|y| = j$ ,  $|xy| = i + j$
- Para qualquer string  $w$ , as equações  $\varepsilon w = w\varepsilon = w$  são válidas, visto que  $\varepsilon$  é a *identidade para a concatenação*, assim como o 0 é a identidade para a soma e o 1 é a identidade para a multiplicação.

### ► Reverso:

- Uma string  $x$  é dita o reverso de uma string  $y$ , ou seja,  $x = y^R$ , se  $x$  contém os mesmos símbolos que  $y$ , porém no sentido inverso, ou seja:
  - Se  $x = a_1a_2 \dots a_{i-1}a_i$ , então  $y = a_ia_{i-1} \dots a_2a_1$

# Conceitos Centrais da Teoria dos Autômatos

## ► Strings/Cadeias (continuação):

### ► Prefixos e Sufixos:

- Uma string  $x$  é um prefixo de uma string  $y$  se for possível escrever  $y$  como  $xw$
- Uma string  $x$  é um sufixo de uma string  $y$  se for possível escrever  $y$  como  $wx$
- Nos dois casos anteriores, pode-se admitir que  $w = \varepsilon$
- Quando  $w \neq \varepsilon$ , diz-se que  $x$  é prefixo próprio ou sufixo próprio da string  $y$
- A string vazia,  $\varepsilon$ , pode ser considerada simultaneamente tanto prefixo quanto sufixo de qualquer string;

### ► Substring/Subcadeia:

- Dadas quatro strings  $x$ ,  $y$ ,  $w$  e  $z$ , a string  $x$  é chamada substring de  $y$  sempre que  $y = wxz$
- Tanto  $w$  quanto  $z$  podem ser vazios;
- Prefixos e sufixos são casos particulares de substrings.

# Conceitos Centrais da Teoria dos Autômatos

## ► Linguagens:

- Um conjunto de strings em que todos os elementos são escolhidos a partir de algum  $\Sigma^*$ , sendo  $\Sigma$  um alfabeto específico, é chamado de linguagem;
- Se  $\Sigma$  é um alfabeto e  $L \subseteq \Sigma^*$ , então  $L$  é uma linguagem sobre  $\Sigma$ ;

## ► Exemplos:

1. A linguagem de todas as strings que consistem em  $n$  0's seguidos por  $n$  valores 1, para algum  $n \geq 0$ :  $\{\epsilon, 01, 0011, 000111, \dots\}$ ;
2. O conjunto de strings de 0's e 1's com um número igual de cada um deles:  $\{\epsilon, 01, 10, 0011, 0101, 1100, 1001, \dots\}$ ;
3. O conjunto de números binários cujo valor é primo:  $\{10, 11, 101, 111, 1011, \dots\}$ ;
4.  $\Sigma^*$  é uma linguagem para qualquer alfabeto  $\Sigma$ ;
5.  $\emptyset$ , a linguagem vazia (também denotado por  $\{\}$ ), é uma linguagem sobre qualquer alfabeto;
6.  $\{\epsilon\}$ , a linguagem que consiste em apenas na string vazia, também é uma linguagem sobre qualquer alfabeto. **Importante:**  $\emptyset \neq \{\epsilon\}$ , pois o primeiro é uma linguagem que não possui nenhuma string, enquanto o segundo é uma linguagem que possui apenas uma string, a string vazia.

# Conceitos Centrais da Teoria dos Autômatos

## ► Problemas:

- É a questão de decidir se uma dada string é elemento de alguma linguagem específica, ou seja, um problema pode ser expresso como pertinência a uma linguagem;
- Se  $\Sigma$  é um alfabeto e  $L$  é uma linguagem sobre  $\Sigma$ , então o problema  $L$  é:
  - Dado uma string  $w$  em  $\Sigma^*$ , definir se  $w$  está ou não em  $L$ ;

## ► Definição de linguagens usando formadores de conjuntos:

- $\{w \mid \text{algo sobre } w\}$ : essa expressão é lida como “o conjunto de palavras  $w$  tais que (seja o que for dito sobre  $w$  à direita da barra vertical)”.

### ► Exemplos:

1.  $\{w \mid w \text{ consiste em um número igual de 0's e 1's}\}$ ;
  2.  $\{w \mid w \text{ é um número inteiro binário primo}\}$ ;
  3.  $\{w \mid w \text{ é um programa em C sintaticamente correto}\}$
- Também é comum substituir  $w$  por alguma expressão com parâmetros e descrever as strings da linguagem declarando condições sobre os parâmetros:
    1.  $\{0^n 1^n \mid n \geq 1\}$ :  $\{01, 0011, 000111, 00001111, \dots\}$
    2.  $\{0^i 1^j \mid 0 \leq i \leq j\}$ :  $\{\epsilon, 1, 11, 111, \dots, 01, 011, 0111, \dots, 0011, 00111, 001111, \dots, 000111, 0001111, 00011111, \dots\}$

# Conceitos Centrais da Teoria dos Autômatos

## ➤ Operações Regulares:

➤ Para  $L_1 = \{x_1, x_2, \dots, x_i\}$  e  $L_2 = \{y_1, y_2, \dots, y_j\}$

### ➤ União:

➤  $L_1 \cup L_2$  OU  $L_1 + L_2 = \{x_1, x_2, \dots, x_i, y_1, y_2, \dots, y_j\}$

### ➤ Concatenação:

➤  $L_1 \cdot L_2$  OU  $L_1 L_2 = \{x_1 y_1, x_1 y_2, \dots, x_1 y_j, x_2 y_1, x_2 y_2, \dots, x_2 y_j, \dots, x_i y_1, x_i y_2, \dots, x_i y_j\}$

### ➤ Fecho:

➤  $L_1^* = \{\epsilon, x_1, x_2, \dots, x_i, x_1 x_1, x_1 x_2, \dots, x_1 x_i, x_2 x_1, x_2 x_2, \dots, x_2 x_i, x_i x_1, x_i x_2, \dots, x_i x_i, x_1 x_1 x_1, x_1 x_1 x_2, x_1 x_2 x_1, x_1 x_2 x_2, \dots\}$

## ➤ Exemplos:

➤ Dado  $L_1 = \{a, ab, abc\}$  e  $L_2 = \{w, ww, www\}$

➤  $L_1 \cup L_2 = \{a, ab, abc, w, ww, www\}$

➤  $L_1 \cdot L_2 = \{aw, aww, awww, abw, abww, abwww, abcw, abcww, abcwww\}$

➤  $L_1^* = \{\epsilon, a, ab, abc, aa, aab, aabc, aba, abab, ababc, abca, abcab, abcabc, \dots\}$

# Conceitos Centrais da Teoria dos Autômatos

## ➤ Classificação de Linguagens segundo a Hierarquia de Chomsky:

- Tipo 0: linguagens recursivamente enumeráveis;
- Tipo 1: linguagens sensíveis ao contexto;
- Tipo 2: linguagens livres de contexto;
- Tipo 3: linguagens regulares.



# Conceitos Centrais da Teoria dos Autômatos

## ► Linguagens, gramáticas e reconhecedores:

Tipo	Classe de Linguagens	Modelo de Gramática	Modelo de Reconhecedor
0	Recursivamente enumeráveis	Irrestrita	Máquina de Turing
1	Sensíveis ao contexto	Sensível ao contexto	Máquina de Turing com fita limitada
2	Livres de contexto	Livre de contexto	Autômato de pilha
3	Regulares	Linear (direita ou esquerda)	Autômato finito





# Conceitos Centrais da Teoria dos Autômatos

## ➤ Relação entre gramáticas dos tipos 0, 1, 2 e 3:

- Toda gramática do Tipo 3 é também do Tipo 2;
- Nem toda gramática do Tipo 2 é também do Tipo 1. São do Tipo 1 apenas aquelas que não possuem produções  $\alpha \rightarrow \beta$  em que  $\beta = \varepsilon$ ;
- Toda gramática do Tipo 1 é também do Tipo 0;



# Conceitos Centrais da Teoria dos Autômatos

## Exercícios Escritos

**Exercício e1.1:** Para  $\Sigma = \{a, b, c\}$ , apresente  $\Sigma^0$ ,  $\Sigma^1$ ,  $\Sigma^2$  e  $\Sigma^3$ .

**Exercício e1.2:** Para  $\Sigma = \{\heartsuit, \star\}$ , apresente as vinte primeiras strings de  $\Sigma^*$  e  $\Sigma^+$ .

**Exercício e1.3:** Seja  $L = \{w \mid w \text{ consiste em um número par de } 0\text{'s ou um número ímpar de } 1\text{'s}\}$ , assinale as strings que pertencem à essa linguagem:

- a) 000
- b)  $\varepsilon$
- c) 0000
- d) 11111
- e) 00111

**Exercício e1.4:** Seja  $L = \{a^n b^m c^n \mid n \geq 0 \text{ e } m \text{ é ímpar}\}$ , assinale as strings que pertencem à essa linguagem:

- a) bbbbbb
- b) bccc
- c) aabbbcc
- d)  $\varepsilon$
- e) aaaabbbbccccc

**Exercício e1.5:** Considerando a linguagem  $L = \Sigma^*$  formada pelos símbolos 😊, 😐 e 😞:

- a) Apresente o alfabeto dessa linguagem;
- b) Dê um exemplo de uma string de comprimento 5.

# Conceitos Centrais da Teoria dos Autômatos

## Exercícios Escritos

**Exercício e1.6:** Os termos  $\varepsilon$ ,  $\emptyset$ ,  $\{\varepsilon\}$  e  $\{\emptyset\}$  denotam, respectivamente:

- a) O conjunto vazio, a string vazia, o conjunto unitário formado pelo conjunto vazio e o conjunto unitário formado pela string vazia;
- b) A string vazia, o conjunto formado pela string vazia, o conjunto vazio e o conjunto unitário formado pelo conjunto vazio;
- c) O conjunto unitário formado pela string vazia, o conjunto unitário formado pelo conjunto vazio, a string vazia e o conjunto vazio;
- d) A string vazia, o conjunto vazio, o conjunto unitário formado pela string vazia e o conjunto unitário formado pelo conjunto vazio.

**Exercício e1.7:** Suponha que  $\Sigma = \{a\}$ , assinale a alternativa correta:

- a)  $a$  é um símbolo, mas não é string;
- b)  $a$  é uma string, mas não é um símbolo;
- c)  $a$  é um símbolo e também uma string;
- d)  $a$  não é símbolo nem string.

# Conceitos Centrais da Teoria dos Autômatos

## Exercícios Escritos

**Exercício e1.8:** Se a string  $x$  é um prefixo próprio de  $y$ , então:

- a)  $y = xw$ , com  $w \neq \varepsilon$
- b)  $y = xw$ , com  $w = \varepsilon$
- c)  $y = wx$ , com  $w \neq \varepsilon$
- d)  $y = wx$ , com  $w = \varepsilon$

**Exercício e1.9:** Se  $x \in L_1$  e  $y \in L_2$ , então:

- a)  $xy \in L_2L_1$
- b)  $yx \in L_1L_2$
- c)  $xy \in L_1L_2$
- d)  $yx \notin L_2L_1$

**Exercício e1.10:** Se  $x \in L$ , então é falso afirmar que:

- a)  $x^R \in L^*$
- b)  $x \in L^*$
- c)  $xx \in L^*$
- d)  $\varepsilon \in L^*$

# Conceitos Centrais da Teoria dos Autômatos

## Exercícios Escritos

**Exercício e1.11:** Os reversos das cadeias  $\varepsilon$ ,  $a$ ,  $aaa$ ,  $aabb$  são, respectivamente:

- a)  $a, \varepsilon, aaa \in bbaa$
- b)  $\varepsilon, a, aaa \in bbaa$
- c)  $a, \varepsilon, \varepsilon \in bbaa$
- d)  $\varepsilon, a, aaa \in aabb$

**Exercício e1.12:** Se  $\varepsilon \in L$ , então:

- a)  $L^+ = L^*$
- b)  $L^+ \cap L^* = \{\varepsilon\}$
- c)  $L^+$  é subconjunto próprio de  $L^*$
- d)  $L^*$  é subconjunto próprio de  $L^+$

# Conceitos Centrais da Teoria dos Autômatos

## Exercícios de Implementação

**Exercício i1.1:** Em uma classe pública denominada **GeradorStrings** (arquivo **GeradorStrings.java**) escreva o método **gerarStringsK**, com a assinatura abaixo, que retorne uma lista de strings de comprimento **k** de um alfabeto **a**, ordenadas em ordem alfabética. Alfabetos nulos ou vazios e comprimentos negativos não devem ser aceitos, devendo lançar uma exceção do tipo **IllegalArgumentException**.

```
public static List<String> gerarStringsK( int k, char... a ) throws IllegalArgumentException
```

**Exercício i1.2:** Na mesma classe do exercício anterior, escreva o método **gerarStringsAteK**, com a assinatura abaixo, que retorne uma lista de strings de comprimento 0 até **k** de um alfabeto **a**, ordenadas em ordem alfabética. A ideia desse método é simular a operação do fecho sobre um alfabeto, sendo que o resultado é um conjunto infinito caso não haja limite a **k**. Alfabetos nulos ou vazios e comprimentos negativos não devem ser aceitos, devendo lançar uma exceção do tipo **IllegalArgumentException**.

```
public static List<String> gerarStringsAteK( int k, char... a ) throws IllegalArgumentException
```

# Conceitos Centrais da Teoria dos Autômatos

## Exercícios de Implementação

**Observação:** Nos arquivos do curso, você encontrará o esqueleto da classe **GeradorStrings** no arquivo **GeradorStringsEsqueleto.java**, dentro do projeto **LFOC4Aula01** do NetBeans, com a infraestrutura básica para testar sua implementação. Para:

```
public static void main( String[] args ) {  
    int k = 3;  
    char[] a = { '0', '1' }; //  $\Sigma = \{0, 1\}$   
    testeGerarStringsK( k, a );  
    testeGerarStringsAteK( k, a );  
}
```

A saída esperada é:

```
 $\Sigma^0 = \{\epsilon\}$   
 $\Sigma^1 = \{0, 1\}$   
 $\Sigma^2 = \{00, 01, 10, 11\}$   
 $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$   
 $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \dots\}$ 
```



HOPCROFT, J. E.; ULLMAN, J. D.; MOTWANI, R. **Introdução à Teoria de Autômatos, Linguagens e Computação**. 2. ed. Rio de Janeiro: Elsevier, 2002. 560 p.

RAMOS, M. V. M.; JOSÉ NETO, J.; VEGA, I. S. **Linguagens Formais: Teoria, Modelagem e Implementação**. Porto Alegre: Bookman, 2009. 656 p.

SIPSER, M. **Introdução à Teoria da Computação**. 2. ed. São Paulo: Cengage Learning, 2017. 459 p.