

# Disciplina CCM-104 – Teoria da Computação

Primeira aula: introdução e exercícios de revisão

---

**Profa. Carla Negri Lintzmayer**

`carla.negri@ufabc.edu.br`

`www.professor.ufabc.edu.br/~carla.negri`

Centro de Matemática, Computação e Cognição – Universidade Federal do ABC



Estes slides contêm uma motivação (Seção 1) para o estudo desta disciplina e alguns exercícios de revisão (Seção 2), sobre o conteúdo recomendado.

# Motivação

---

É uma área da Ciência da Computação que tenta responder  
*“quais são as capacidades e limitações fundamentais dos computadores?”*

É uma área da Ciência da Computação que tenta responder

*“quais são as capacidades e limitações fundamentais dos computadores?”*

- O que é computação?

É uma área da Ciência da Computação que tenta responder  
*“quais são as capacidades e limitações fundamentais dos computadores?”*

- O que é computação?
  - Teoria dos Autômatos

É uma área da Ciência da Computação que tenta responder  
*“quais são as capacidades e limitações fundamentais dos computadores?”*

- O que é computação?
  - Teoria dos Autômatos
- O que pode ser computado?

É uma área da Ciência da Computação que tenta responder  
*“quais são as capacidades e limitações fundamentais dos computadores?”*

- O que é computação?
  - Teoria dos Autômatos
- O que pode ser computado?
  - Teoria da Computabilidade

É uma área da Ciência da Computação que tenta responder  
*“quais são as capacidades e limitações fundamentais dos computadores?”*

- O que é computação?
  - Teoria dos Autômatos
- O que pode ser computado?
  - Teoria da Computabilidade
- O quão bem pode ser computado?

É uma área da Ciência da Computação que tenta responder  
*“quais são as capacidades e limitações fundamentais dos computadores?”*

- O que é computação?
  - Teoria dos Autômatos
- O que pode ser computado?
  - Teoria da Computabilidade
- O quão bem pode ser computado?
  - Teoria da Complexidade

# O que é Computação (informal)?

O que é Computação (informal)?

# O que é Computação (informal)?

O que é Computação (informal)?

- É o processo de solução de um **problema** por meio de um **algoritmo**

# Tipos de problemas

Problemas vêm em vários sabores:

- Decisão
- Otimização
- Contagem
- Busca
- etc



**Decisão:** problema cuja resposta é sim ou não.

- Dado um número, ele é par?
- Dado um número, ele é primo?
- Dados dois vértices em um grafo, existe caminho entre eles?
- Dados dois vértices em um grafo, existe caminho com no máximo  $k$  arestas entre eles?

**Otimização:** problema que procura a melhor resposta dentre todas as possíveis.

- Dados dois números, qual o maior divisor comum de ambos?
- Qual o menor caminho entre dois vértices?
- Qual o maior caminho entre dois vértices?
- Dado um conjunto de itens valorados e um recipiente com capacidade finita, qual o maior valor obtido ao escolher itens que caibam no recipiente?

**Contagem:** problema cuja resposta é o número de soluções possíveis.

- Dado um número, quantos divisores ele possui?
- Dado um número, quantos fatores primos ele possui?
- Dados dois vértices em um grafo, quantos caminhos existem entre ambos?

## Tipos de problemas

Qualquer tipo de problema pode ser transformado em (reduzido a) um problema de decisão equivalente.

Qualquer tipo de problema pode ser transformado em (reduzido a) um problema de decisão equivalente.

### Caixeiro Viajante – Otimização

- **Entrada:** Grafo  $G$  e função de custos  $c : E(G) \rightarrow \mathbb{R}^+$ .
- **Saída:** Um ciclo hamiltoniano sobre  $G$  de custo mínimo.

Qualquer tipo de problema pode ser transformado em (reduzido a) um problema de decisão equivalente.

### Caixeiro Viajante – Otimização

- **Entrada:** Grafo  $G$  e função de custos  $c : E(G) \rightarrow \mathbb{R}^+$ .
- **Saída:** Um ciclo hamiltoniano sobre  $G$  de custo mínimo.

### Caixeiro Viajante – Decisão

- **Entrada:** Grafo  $G$ , função de custos  $c : E(G) \rightarrow \mathbb{R}^+$ , e número  $r \in \mathbb{R}^+$ .
- **Saída:** SIM se existe ciclo hamiltoniano em  $G$  de custo menor do que  $r$ ; NÃO caso contrário.

Qualquer tipo de problema pode ser transformado em (reduzido a) um problema de decisão equivalente.

### Caixeiro Viajante – Otimização

- **Entrada:** Grafo  $G$  e função de custos  $c : E(G) \rightarrow \mathbb{R}^+$ .
- **Saída:** Um ciclo hamiltoniano sobre  $G$  de custo mínimo.

### Caixeiro Viajante – Decisão

- **Entrada:** Grafo  $G$ , função de custos  $c : E(G) \rightarrow \mathbb{R}^+$ , e número  $r \in \mathbb{R}^+$ .
- **Saída:** SIM se existe ciclo hamiltoniano em  $G$  de custo menor do que  $r$ ; NÃO caso contrário.

Por isso, podemos nos restringir a problemas de decisão.

## Tudo é número string!

Strings podem representar qualquer objeto:

- Números: 5 → "5" ou "101" ou "11111" ...

# Tudo é número string!

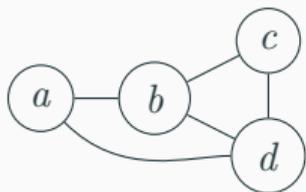
Strings podem representar qualquer objeto:

- Números:  $5 \rightarrow "5"$  ou  $"101"$  ou  $"11111"$  ...
- Polinômios:  $4x^2 + 3y - 5 \rightarrow "4x^2 + 3y - 5"$  ou  $"3,2,4x^2+3y1-5y0"$  ...

# Tudo é número string!

Strings podem representar qualquer objeto:

- Números:  $5 \rightarrow "5"$  ou  $"101"$  ou  $"11111"$  ...
- Polinômios:  $4x^2 + 3y - 5 \rightarrow "4x^2 + 3y - 5"$  ou  $"3,2,4x^2+3y1-5y0"$  ...
- Grafos:

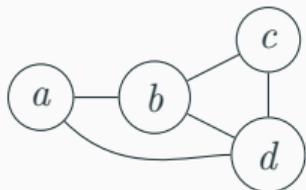


$\rightarrow "(\{a,b,c,d\}, \{ab, bc, cd, bd, ad\})"$  ou  
 $"a(b,d); b(a,c,d); c(b,d); d(a,b,c)"$

# Tudo é número string!

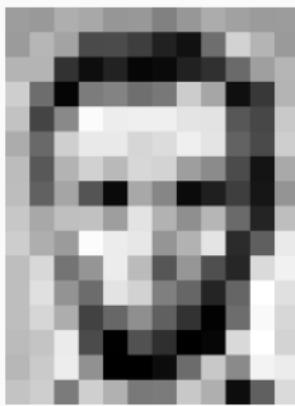
Strings podem representar qualquer objeto:

- Números:  $5 \rightarrow "5"$  ou  $"101"$  ou  $"11111"$  ...
- Polinômios:  $4x^2 + 3y - 5 \rightarrow "4x^2 + 3y - 5"$  ou  $"3,2,4x^2+3y1-5y0"$  ...
- Grafos:



$\rightarrow "(\{a, b, c, d\}, \{ab, bc, cd, bd, ad\})"$  ou  
 $"a(b, d); b(a, c, d); c(b, d); d(a, b, c)"$

- Imagens:



|     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 157 | 153 | 174 | 168 | 150 | 152 | 129 | 151 | 172 | 161 | 155 | 156 |
| 155 | 182 | 163 | 74  | 75  | 62  | 33  | 17  | 110 | 210 | 180 | 154 |
| 180 | 180 | 50  | 14  | 34  | 6   | 10  | 33  | 48  | 106 | 159 | 181 |
| 206 | 109 | 5   | 124 | 131 | 111 | 120 | 204 | 165 | 15  | 56  | 180 |
| 194 | 68  | 137 | 251 | 237 | 239 | 239 | 228 | 227 | 87  | 71  | 201 |
| 172 | 105 | 207 | 233 | 233 | 214 | 220 | 230 | 228 | 94  | 74  | 206 |
| 188 | 88  | 179 | 209 | 185 | 215 | 211 | 158 | 139 | 75  | 20  | 169 |
| 189 | 97  | 169 | 84  | 10  | 168 | 134 | 11  | 31  | 62  | 22  | 148 |
| 199 | 168 | 191 | 193 | 158 | 227 | 178 | 143 | 182 | 156 | 96  | 190 |
| 205 | 174 | 155 | 252 | 236 | 231 | 149 | 178 | 228 | 43  | 95  | 234 |
| 190 | 216 | 116 | 149 | 236 | 187 | 85  | 150 | 79  | 38  | 218 | 241 |
| 190 | 224 | 147 | 108 | 227 | 210 | 127 | 102 | 96  | 101 | 255 | 224 |
| 190 | 214 | 173 | 66  | 103 | 143 | 96  | 50  | 2   | 109 | 249 | 215 |
| 187 | 196 | 225 | 75  | 1   | 81  | 47  | 9   | 6   | 217 | 255 | 211 |
| 183 | 202 | 237 | 145 | 0   | 0   | 12  | 108 | 200 | 138 | 243 | 236 |
| 195 | 206 | 123 | 207 | 177 | 121 | 123 | 200 | 175 | 19  | 96  | 218 |

|     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 157 | 153 | 174 | 168 | 150 | 152 | 129 | 151 | 172 | 161 | 155 | 156 |
| 155 | 182 | 163 | 74  | 75  | 62  | 33  | 17  | 110 | 210 | 180 | 154 |
| 180 | 180 | 50  | 14  | 34  | 6   | 10  | 33  | 48  | 106 | 159 | 181 |
| 206 | 109 | 5   | 124 | 131 | 111 | 120 | 204 | 165 | 15  | 56  | 180 |
| 194 | 68  | 137 | 251 | 237 | 239 | 239 | 228 | 227 | 87  | 71  | 201 |
| 172 | 105 | 207 | 233 | 233 | 214 | 220 | 230 | 228 | 94  | 74  | 206 |
| 188 | 88  | 179 | 209 | 185 | 215 | 211 | 158 | 139 | 75  | 20  | 169 |
| 189 | 97  | 169 | 84  | 10  | 168 | 134 | 11  | 31  | 62  | 22  | 148 |
| 199 | 168 | 191 | 193 | 158 | 227 | 178 | 143 | 182 | 156 | 96  | 190 |
| 205 | 174 | 155 | 252 | 236 | 231 | 149 | 178 | 228 | 43  | 95  | 234 |
| 190 | 216 | 116 | 149 | 236 | 187 | 85  | 150 | 79  | 38  | 218 | 241 |
| 190 | 224 | 147 | 108 | 227 | 210 | 127 | 102 | 96  | 101 | 255 | 224 |
| 190 | 214 | 173 | 66  | 103 | 143 | 96  | 50  | 2   | 109 | 249 | 215 |
| 187 | 196 | 225 | 75  | 1   | 81  | 47  | 9   | 6   | 217 | 255 | 211 |
| 183 | 202 | 237 | 145 | 0   | 0   | 12  | 108 | 200 | 138 | 243 | 236 |
| 195 | 206 | 123 | 207 | 177 | 121 | 123 | 200 | 175 | 19  | 96  | 218 |

## Linguagem formal

Qualquer *conjunto de strings* sobre um alfabeto.

$$L = \{010, 11, 0, 100, 01\}$$

$$T = \{\textit{banana}, \textit{melancia}, \textit{abacate}, \textit{cereja}\}$$

$$C = \{\textit{if}, \textit{while}, \textit{int}, \textit{main}\}$$

Qualquer problema de decisão pode ser representado por um conjunto que contém strings que representam suas instâncias sim.

- $\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, \dots\}$
- Decidir se um número  $n$  é primo  $\equiv$  decidir se  $n \in \mathcal{P}$

Qualquer problema de decisão pode ser representado por um conjunto que contém strings que representam suas instâncias sim.

- $\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, \dots\}$
- Decidir se um número  $n$  é primo  $\equiv$  decidir se  $n \in \mathcal{P}$

Em outras palavras, **linguagens formalizam problemas**.

Qualquer problema de decisão pode ser representado por um conjunto que contém strings que representam suas instâncias sim.

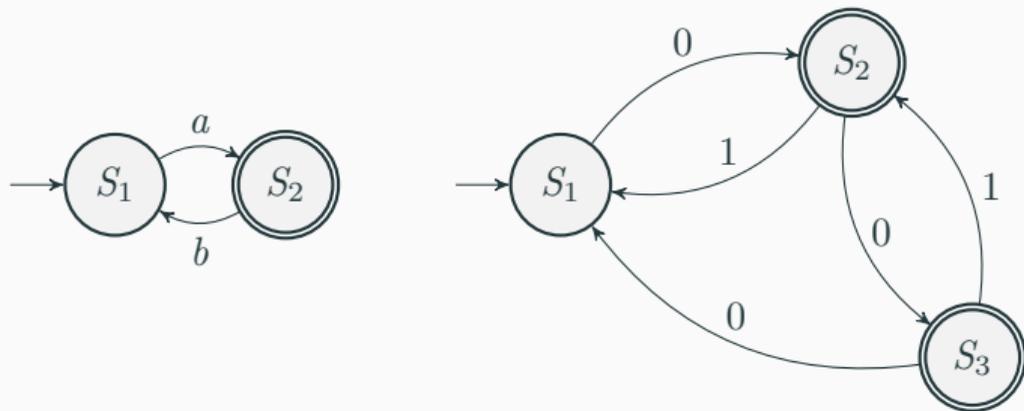
- $\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, \dots\}$
- Decidir se um número  $n$  é primo  $\equiv$  decidir se  $n \in \mathcal{P}$

Em outras palavras, **linguagens formalizam problemas**.

**Computação (formal)**: decidir se uma string pertence à linguagem.

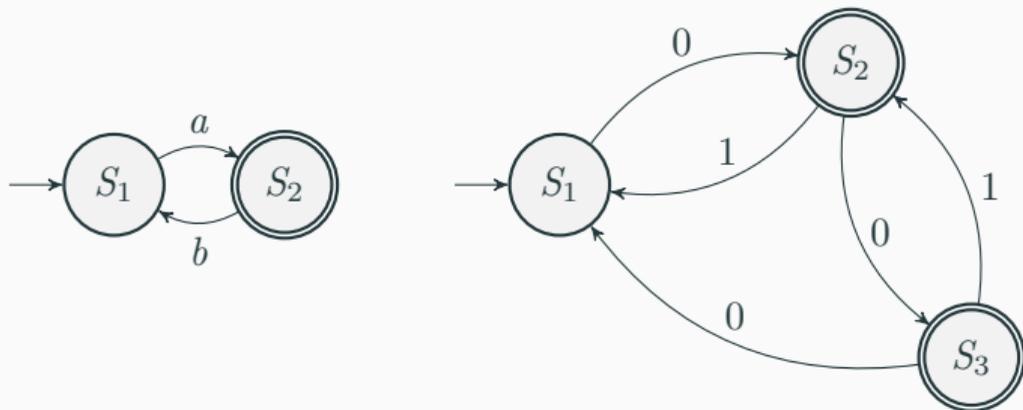
## Autômatos

Máquinas (modelos matemáticos) abstratas que reconhecem palavras de uma linguagem.



## Autômatos

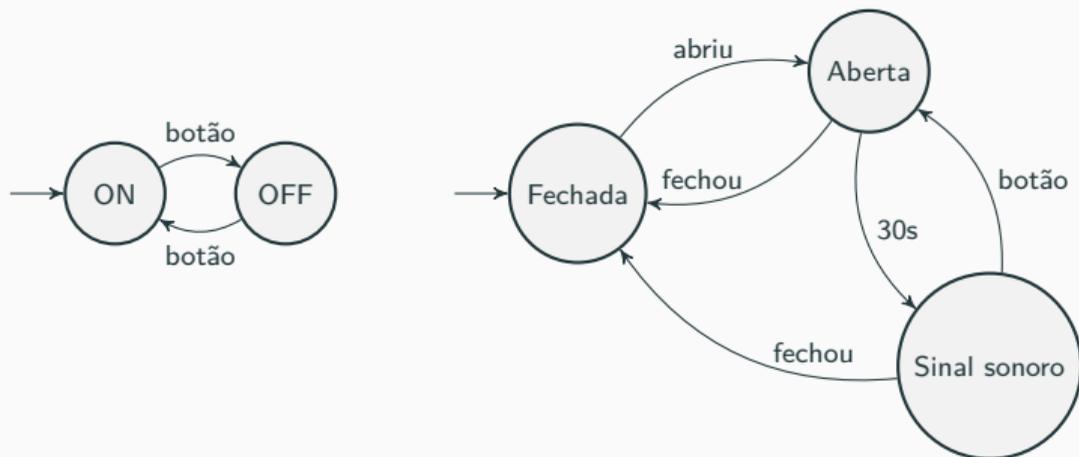
Máquinas (modelos matemáticos) abstratas que reconhecem palavras de uma linguagem.



- Quais linguagens podem ser reconhecidas por cada autômato?

# Autômatos (ou máquinas de estados)

- Abstraem detalhes de implementação dos computadores.
- Introduzem conceitos relevantes a outras áreas e aplicações:
  - Projeto de linguagens de programação (compiladores)
  - Processamento de linguagem natural
  - Análise de texto (Parsing)
  - Projeto e verificação de circuitos e sistemas digitais



## **Máquinas de Turing**

Autômatos com a mesma capacidade de computação dos computadores de hoje.

## Máquinas de Turing

Autômatos com a mesma capacidade de computação dos computadores de hoje.

Explorar os limites da computação:

## Máquinas de Turing

Autômatos com a mesma capacidade de computação dos computadores de hoje.

Explorar os limites da computação:

- Decidibilidade (o que pode ser computado?)

## Máquinas de Turing

Autômatos com a mesma capacidade de computação dos computadores de hoje.

Explorar os limites da computação:

- Decidibilidade (o que pode ser computado?)
  - Problema da Parada: “Dado um programa  $P$  e uma entrada  $I$  para  $P$ , o programa  $P$  quando alimentado com a entrada  $I$  para ou entre em *loop infinito*?”

## Máquinas de Turing

Autômatos com a mesma capacidade de computação dos computadores de hoje.

Explorar os limites da computação:

- Decidibilidade (o que pode ser computado?)
  - Problema da Parada: “Dado um programa  $P$  e uma entrada  $I$  para  $P$ , o programa  $P$  quando alimentado com a entrada  $I$  para ou entre em *loop infinito*?”
- Tratabilidade (quão eficiente algo pode ser computado?)
  - Complexidade de tempo e espaço
  - P vs. NP

## Recomendação

- Conhecimentos de programação (em qualquer linguagem imperativa)
- Familiaridade com estruturas de dados básicas (vetores, pilhas)
- Familiaridade com linguagem matemática (conjuntos, sequências, relações, funções)
- Capacidade para reconhecer argumentos lógicos em uma demonstração matemática (indução)
- Ideal: disciplina “Análise de algoritmos e estruturas de dados” ou equivalente

Veja a seção [Recomendação](#) no site, com slides de revisão!

- Conteúdo dado no quadro, com alguns slides de apoio eventualmente
  - Seguem o livro do Sipser, “Introdução à teoria da computação”
  - Recursos extras disponíveis no site

- Conteúdo dado no quadro, com alguns slides de apoio eventualmente
  - Seguem o livro do Sipser, “Introdução à teoria da computação”
  - Recursos extras disponíveis no site
  - **Videoaulas não substituem as aulas presenciais!**

- Conteúdo dado no quadro, com alguns slides de apoio eventualmente
  - Seguem o livro do Sipser, “Introdução à teoria da computação”
  - Recursos extras disponíveis no site
  - **Videoaulas não substituem as aulas presenciais!**
- Atendimentos logo após as aulas de quinta-feira, até 18h
  - Qualquer pergunta e feedback são **sempre** bem-vindos
  - Não deixe dúvidas acumularem

- Conteúdo dado no quadro, com alguns slides de apoio eventualmente
  - Seguem o livro do Sipser, “Introdução à teoria da computação”
  - Recursos extras disponíveis no site
  - **Videoaulas não substituem as aulas presenciais!**
- Atendimentos logo após as aulas de quinta-feira, até 18h
  - Qualquer pergunta e feedback são **sempre** bem-vindos
  - Não deixe dúvidas acumularem
- Espero que você
  - Participe dos atendimentos
  - Seja autor das suas atividades
  - Se divirta :)

<http://professor.ufabc.edu.br/~carla.negri/cursos/2024Q3-TC/>

- Estude bem o conteúdo do site.
- Verifique-o com frequência!

## Exercícios de revisão

---

- Existem **materiais de revisão** sobre os conteúdos necessários, principalmente sobre **indução**.

## Exercício 1

Descreva em português o conjunto  $\{(a_1, a_2): a_1 = 2a_2 \text{ e } a_2 \in \mathbb{Z}\}$ .

## Exercício 2

Sejam  $A = \{a, b, c\}$ ,  $B = \{x, y\}$  e  $C = \{1, 2, 3, 4\}$ . Forneça um exemplo de função  $f$  tal que  $f: A \times B \rightarrow \mathcal{P}(C)$ .

## Exercício 3

Para quaisquer dois conjuntos  $A$  e  $B$ , mostre que  $\overline{A \cup B} = \overline{A} \cap \overline{B}$ .

## Exercício 4

Prove que a soma dos  $n$  primeiros cubos perfeitos é  $\left(\frac{n(n+1)}{2}\right)^2$  para todo  $n \geq 1$ .