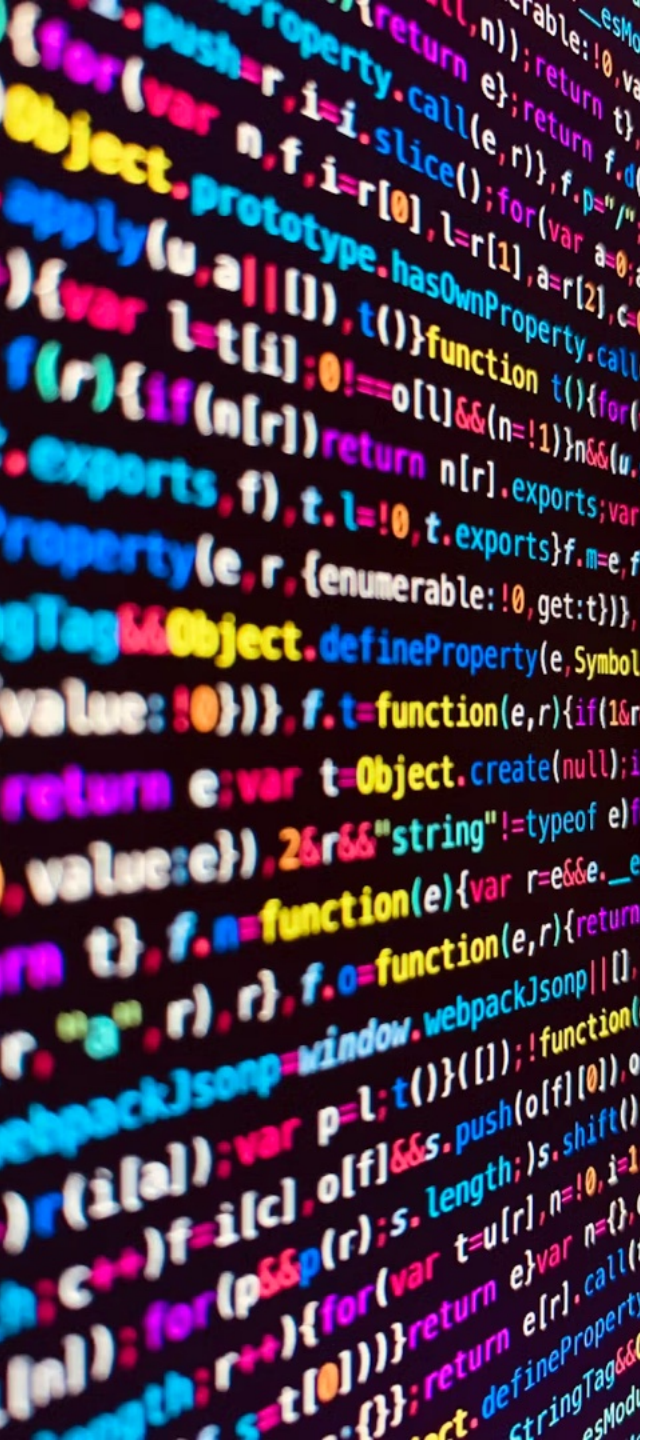


Autômatos, Linguagens e Computação

Introdução à Teoria da Computação

Pontifícia Universidade Católica de Campinas

Prof. Dr. Denis M. L. Martins

Objetivos

- Compreender o escopo e a importância da Teoria da Computação.
- Diferenciar os três pilares principais:
  - Computabilidade
  - Complexidade
  - Autômatos & Linguagens

Contextualizando a Teoria da Computação

Área	Descrição
Algoritmos	Procedimentos finitos que resolvem problemas computacionais.
Arquitetura de Sistemas	Implementação dos algoritmos.
Teoria da Computação	Fundamentos matemáticos que delimitam o que pode ser computado e com que recursos.

A Teoria da Computação não diz **"como"** resolver um problema, mas "se" ele é pode ser **solucionado** e, se puder, o **quão difícil** (complexo) o problema é.

Teoria da Computabilidade

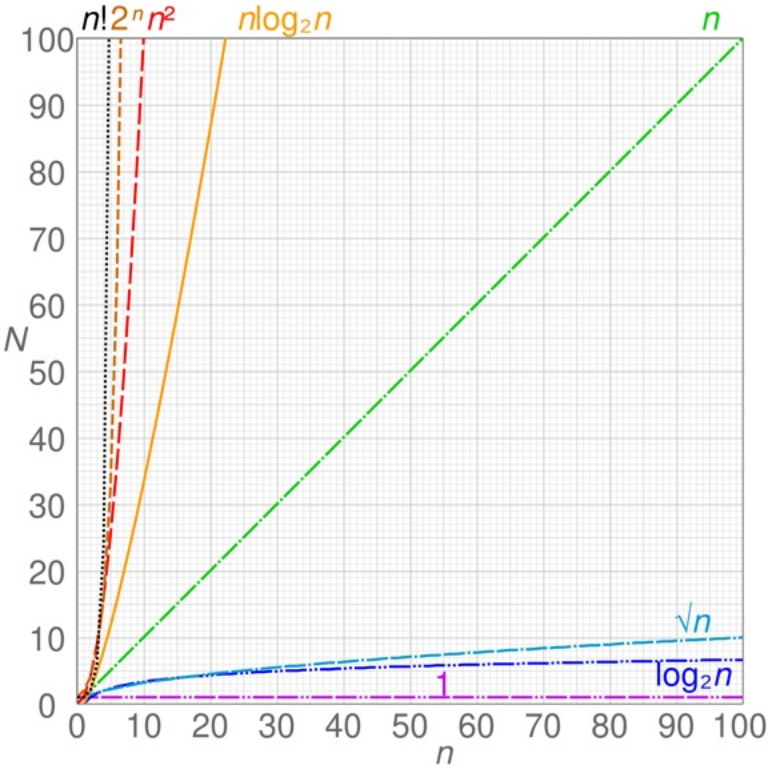
- Pergunta Central: **Existe** algoritmo que resolve problema X?
- Conceitos Chave:
  - Máquina de Turing (MT): modelo universal de computação.
  - Problema Decidível: MT termina em todos os casos, produz resposta correta.
  - Problema Indecidível: não há MT que o resolva em todas as entradas.

Problema	Status
Problema da Parada (Halting/Halteproblem)	Indécidível
Problema da decisão (Entscheidungsproblem)	Indécidível
Equivalência de gramáticas livres de contexto	Decidível
Conjunto dos números primos	Decidível

Tese de Church-Turing

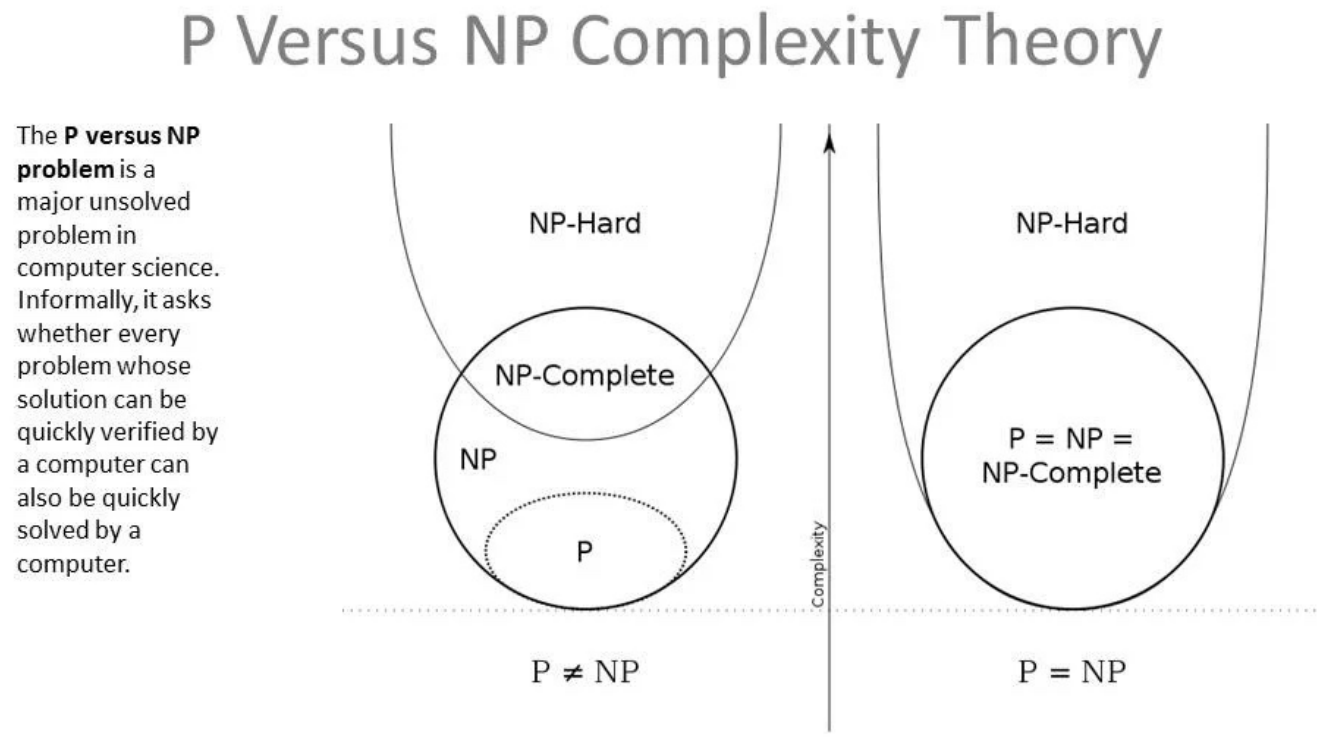
Toda função que seria naturalmente considerada computável pode ser computada por uma Máquina de Turing.

Teoria da Complexidade



- **Pergunta Central**: O que faz alguns problemas computacionalmente difíceis e outros fáceis?
- **Conceitos Chave**:
  - Classes de Complexidade
  - Recursos (tempo/espaco)

Classe	Descrição
P	Problemas resolvíveis em tempo polinomial por MT determinística.
NP	Problemas cujas soluções podem ser verificadas em tempo polinomial.
EXP	Problemas resolvíveis em tempo exponencial.
PSPACE	Classes baseadas no consumo de memória.



P vs NP

- A maioria dos especialistas acredita que $P \neq NP$, mas a prova formal ainda não existe.
- Se $P = NP$: quebra de criptografia baseada em problemas NP-difíceis.
- Se $P \neq NP$: confirma que certos problemas são inerentemente "difíceis" e exigem heurísticas ou aproximações.
- Fonte da imagem: ML Science

Teoria dos Autômatos & Linguagens

- Objetivo: Modelar sistemas e processos de computação de maneira formal (precisa).
- Conceitos Chave:
  - AFD/AFND: Reconhecimento de linguagens regulares.
  - AP (Pushdown Automaton): Reconhecimento de linguagens livres de contexto.
  - MT: Reconhecimento de linguagens recursivamente enumeráveis.
  - Hierarquia de Chomsky: Classes de gramáticas (Tipo-0 a Tipo-3) por restrições crescentes

Perguntas & Discussões

- Quais são as principais limitações que você vê quando tentamos descrever um sistema complexo (ex.: redes sociais, carros autônomos) usando apenas modelos matemáticos?
- Você acha que a teoria da computação pode prever todos os comportamentos de sistemas reais? Por que ou por que não?