Disciplinas MCTA015-13 / CCM-104 Linguagens Formais e Autômata / 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

UFABC

Conteúdo

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

Como ainda estamos em uma pandemia, começaremos com alguns lembretes importantes.

Bem-vindos de volta!

#DeVoltaAosCampi

COVID

- O uso de máscara é obrigatório na UFABC
- Máscara **não** é *chin diaper*!

COVID

- Não retire a máscara para espirrar/tossir!
- Cubra a boca ao espirrar/tossir.

Ausências

- Haverá cobrança de presença (mín. 75%)
- Se estiver com problemas ou doente, me envie um e-mail
    - A página da disciplina contém material de apoio e o conteúdo das aulas

Motivação

Teoria da Computação

É uma área da Ciência da Computação que tenta responder

> *"quais são as capacidades e limitações fundamentais dos computadores?"*

Teoria da 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 da 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

Teoria da 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
- O que pode ser computado?

Teoria da 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
- O que pode ser computado?
  - Teoria da Computabilidade

Teoria da 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
- O que pode ser computado?
  - Teoria da Computabilidade
- O quão bem pode ser computado?

Teoria da 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
- 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

Tipos de problemas

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?

Tipos de problemas

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?

Tipos de problemas

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.

Tipos de problemas

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 $c : E(G) \to \mathbb{R}^+$.
- **Saída:** Um ciclo hamiltoniano sobre G de custo mínimo.

Tipos de problemas

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 $c : E(G) \to \mathbb{R}^+$.
- **Saída:** Um ciclo hamiltoniano sobre G de custo mínimo.

Caixeiro Viajante – Decisão

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

Tipos de problemas

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 $c : E(G) \to \mathbb{R}^+$.
- **Saída:** Um ciclo hamiltoniano sobre G de custo mínimo.

Caixeiro Viajante – Decisão

- **Entrada:** Grafo G, função de custos $c : E(G) \to \mathbb{R}^+$, 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: $5 \rightarrow$ "5" ou "00110101" ou "101", etc ...

Tudo é ~~número~~ string!

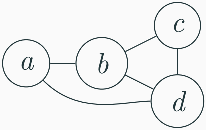
Strings podem representar qualquer objeto:

- Números: $5 \rightarrow$ “`5`” ou “`00110101`” ou “`101`”, etc …
- Polinômios: $4x^2 + 3y - 5 \rightarrow$ “`4x^2 + 3y - 5`” ou “`3,2,4x2+3y1-5y0`” …

Tudo é ~~número~~ string!

Strings podem representar qualquer objeto:

- Números: $5 \rightarrow$ “5” ou “00110101” ou “101”, etc ...
- Polinômios: $4x^2 + 3y - 5 \rightarrow$ “4x^2 + 3y - 5” ou “3,2,4x2+3y1-5y0” ...
- Grafos:

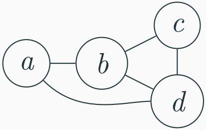


$\rightarrow$ “({a,b,c,d},{ab,bc,cd,bd,ad})”

Tudo é ~~número~~ string!

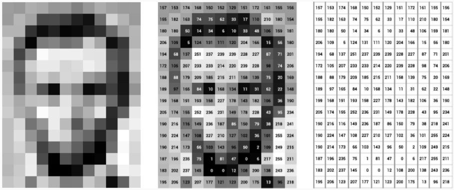
Strings podem representar qualquer objeto:

- Números: $5 \to$ “`5`” ou “`00110101`” ou “`101`”, etc ...
- Polinômios: $4x^2 + 3y - 5 \to$ “`4x^2 + 3y - 5`” ou “`3,2,4x2+3y1-5y0`” ...
- Grafos:



$\to$ “`({a,b,c,d},{ab,bc,cd,bd,ad})`”

- Imagens:



Linguagem formal

Qualquer *conjunto de strings* sobre um alfabeto.

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

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

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

Linguagens formais ...

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

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

Linguagens formais ...

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

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

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

Linguagens formais ...

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

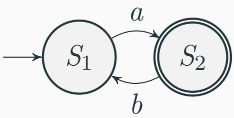
- $\mathcal{P} = \{2, 3, 5, 7, 11, 13, 17, 19, \ldots\}$
- Decidir se um número n é primo $\equiv$ decidir se $n \in \mathcal{P}$

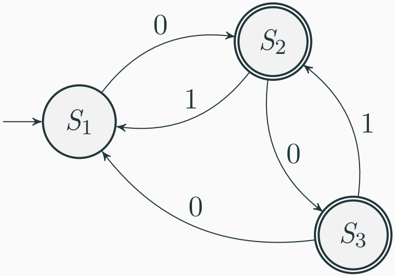
Em outras palavras, **linguagens formalizam problemas**.

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

Autômatos

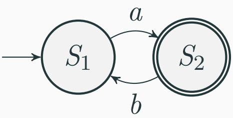
Máquinas (modelos matemáticos) abstratas que reconhecem palavras de uma linguagem.

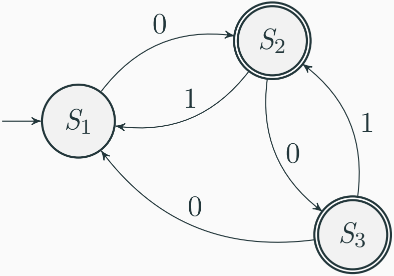




Autômatos

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

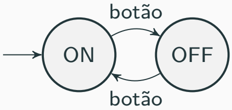


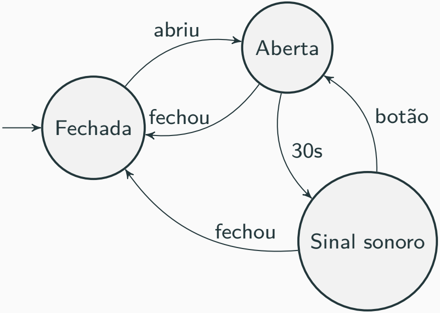


- Quais linguagens podem ser reconhecidas em cada autômato?

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*?

Máquinas de Turing

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*?
- Tratabilidade (quão eficiente algo pode ser computado?)
  - Complexidade
  - $\mathcal{P}$ vs $\mathcal{NP}$

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)

Veja a seção Recomendação no site, com slides de revisão!

Sobre as aulas

- 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 (incluindo videoaulas)

Sobre as aulas

- 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 (incluindo videoaulas)
  - Videoaulas não substituem as aulas presenciais!

Sobre as aulas

- 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 (incluindo videoaulas)
  - Videoaulas não substituem as aulas presenciais!
- Atendimentos logo após as aulas, até 19h
  - Qualquer pergunta e feedback são **sempre** bem-vindos
  - Não deixe dúvidas acumularem

Sobre as aulas

- 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 (incluindo videoaulas)
    - Videoaulas não substituem as aulas presenciais!
- Atendimentos logo após as aulas, até 19h
    - 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 :)

Informações

professor.ufabc.edu.br/~carla.negri/cursos/2022Q3-TC/

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

Exercícios de revisão

Apoio

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

Exercício 1

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

Exercício 2

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

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 $n \geq 1$.