[Aula 18] Máquina de Turing

Prof. João F. Mari
joaof.mari@ufv.br

BIBLIOGRAFIA

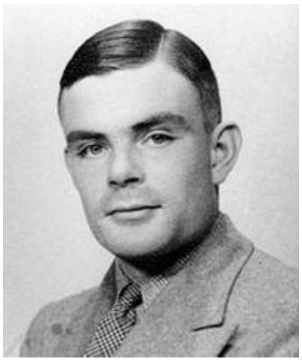
- MENEZES, P. B. **Linguagens formais e autômatos**, 6. ed., Bookman, 2011.
  - Capítulo 8.
  - + Slides disponibilizados pelo autor do livro.

Linguagens recursivamente enumeráveis

- **Ciência da Computação:**
  - Conhecimento sistematizado relativo à computação
- A origem da Ciência da Computação é remota:
  - Grécia antiga(III a.C):
    - Euclides desenvolve um algoritmo para encontrar o MDC.
  - Babilônia:
    - Estudos sobre complexidade e redutibilidade de problemas.
  - Início do século XX:
    - Pesquisas com o objetivo de definir:
      - Modelo computacional suficientemente genérico;
      - Capaz de implementar qualquer função computável.

Linguagens recursivamente enumeráveis

- **Máquina de Turing:**
  - Proposto por Alan Turing (1936) :
  - Aceito como uma formalização de:
    - Procedimento efetivo;
    - Algoritmo;
    - Função computável.
  - Algoritmo:
    - Sequência finita de instruções;
    - Podem ser realizadas mecanicamente em um tempo finito.



https://pt.wikipedia.org/wiki/Alan_Turing

https://pt.wikipedia.org/wiki/Máquina_de_Turing

Linguagens recursivamente enumeráveis

- **Hipótese de Church (ou Tese de Church-Turing):**
  - Proposto por Alonzo Church (1936)

"Qualquer função computável pode ser processada por uma máquina de Turing."

  - Existe um procedimento expresso na forma de uma máquina de Turing capaz de processar a função.

https://en.wikipedia.org/wiki/Alonzo_Church
https://pt.wikipedia.org/wiki/Tese_de_Church-Turing

Linguagens recursivamente enumeráveis

- **Máquina de Turing:**
  - É um **autômato**;
  - A fita não possui tamanho máximo;
  - Pode ser usada simultaneamente como:
    - Dispositivo de entrada;
    - Dispositivo de saída;
    - Memória de trabalho.

- **Linguagens Recursivamente Enumeráveis (Tipo 0):**
  - Classe de linguagens **aceitas** por uma **máquina de Turing**;
  - De acordo com a **Hipótese de Church**:
    - A classe das Linguagens Recursivamente Enumeráveis é:
      - O conjunto de todas as linguagens que podem ser reconhecidas mecanicamente em um tempo finito por uma máquina de Turing.

Linguagens recursivamente enumeráveis

- **Gramática Irrestrita:**
  - Gramáticas sem restrições sobre a forma das produções
  - Possui o mesmo poder computacional que o formalismo Máquina de Turing
- Consequência importante do estudo das linguagens recursivamente enumeráveis
  - **"Existem mais problemas não-solucionáveis do que problemas solucionáveis."**

Linguagens recursivamente enumeráveis

- A classe das Linguagens Recursivamente Enumeráveis:
  - Inclui linguagens para as quais é **impossível** determinar mecanicamente se uma palavra não pertence à linguagem.
    - Se L é uma destas linguagens, então:
      - Existe pelo menos uma palavra w não pertencente a L que;
      - Qualquer máquina de Turing M que aceita L entra em loop infinito.
  - Ou seja:
    - Se w **pertence** a L:
      - M para e aceita a entrada.
    - Se w **não pertence** a L:
      - (A) M para rejeitando a palavra ou
      - (B) M permanecer processando indefinidamente (loop).

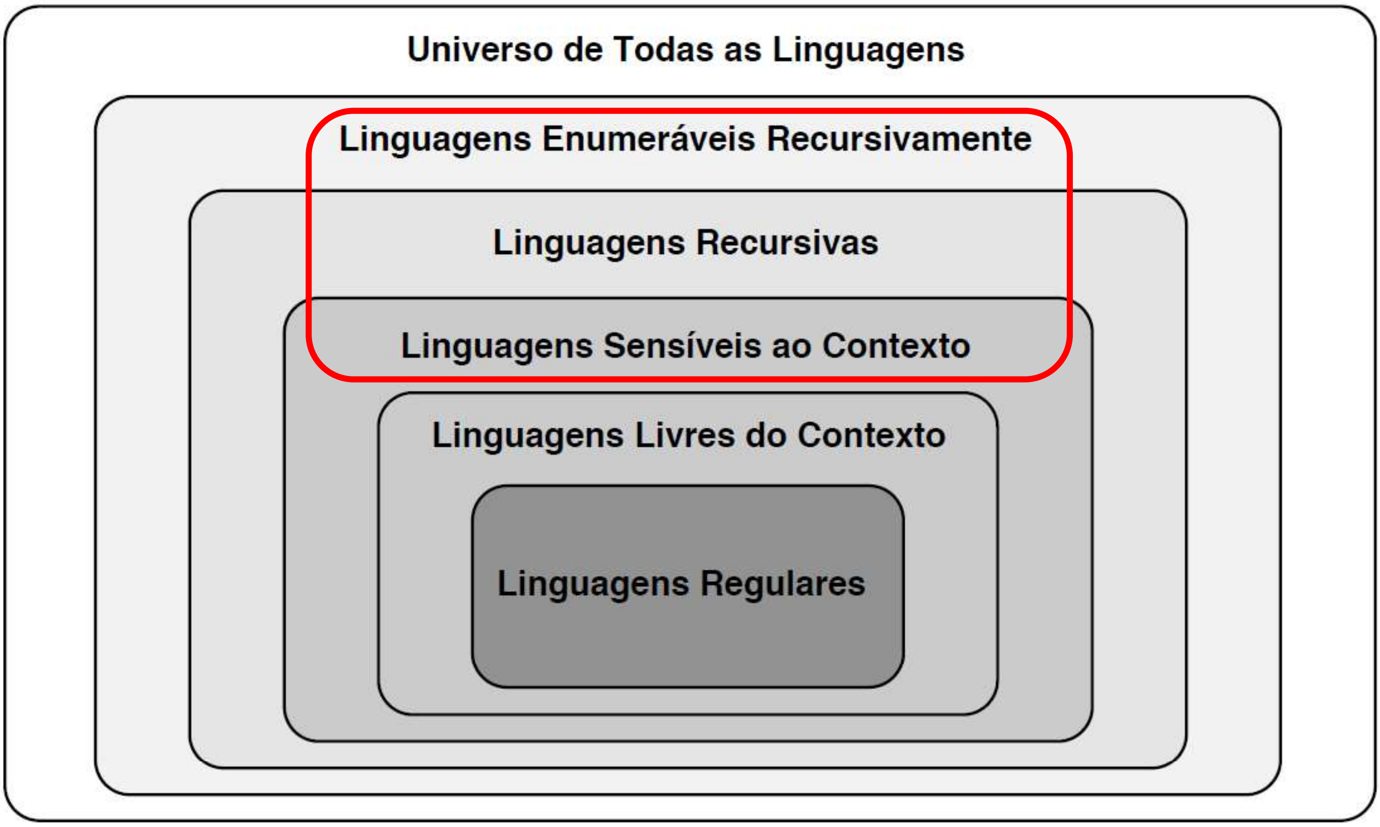
Linguagens recursivamente enumeráveis

- A classe das Linguagens Recursivamente Enumeráveis:
  - **Linguagens recursivas:**
    - Subclasse da classe das Linguagens Recursivamente Enumeráveis.
    - Existe pelo menos uma máquina de Turing que **para** qualquer entrada w:
      - Seja aceitando ou rejeitando w.

Linguagens recursivamente enumeráveis

- A classe das Linguagens Recursivamente Enumeráveis:
  - **Linguagens Sensíveis ao Contexto (Tipo 1):**
    - Aceitas por uma Máquina de Turing com Fita Limitada
      - Limitação no tamanho da fita (finita).
  - Gramática Sensível ao Contexto:
    - Diferente das Linguagens Livres de Contexto:
    - Possuem produções na forma:
      - $yAz \rightarrow ywz$, para $A \in V$; y e $z \in (V \cup T)^*$; e $w \in (V \cup T)^+$;
        - » yAz : Só é possível derivar a partir de A de acordo com o contexto.
  - A Classe das Linguagens Sensíveis ao Contexto:
    - Está contida propriamente na Classe das Linguagens Recursivas;
    - Inclui a grande maioria das linguagens aplicadas.

Linguagens recursivamente enumeráveis



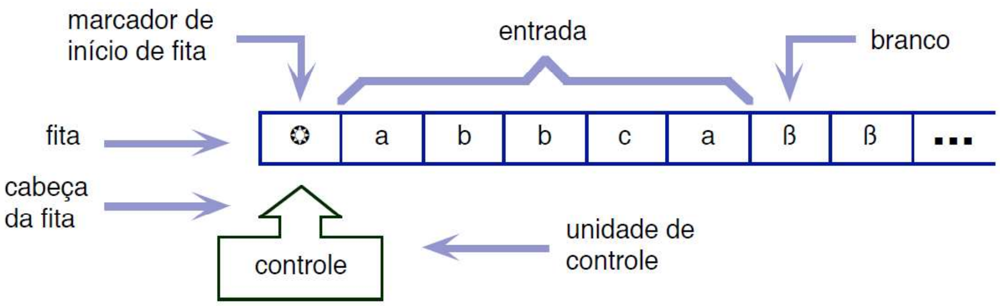
Máquina de Turing

- **Três partes:**
  - <u>Fita:</u>
    - Dispositivo de entrada, saída e memória de trabalho.
      - Simultaneamente
    - Infinita: “tão grande quanto necessário”
    - Cada célula uma armazenando um símbolo.
    - Símbolos podem ser:
      - Do alfabeto de entrada, do alfabeto auxiliar, “branco” ou "marcador de início de fita“.
  - <u>Unidade de Controle:</u>
    - Estado corrente da máquina;
    - Possui uma unidade de leitura e gravação (cabeça da fita);
    - Acessa uma célula da fita de cada vez;
    - Movimenta para a esquerda ou para a direita.
  - <u>Programa, Função Programa ou Função de Transição:</u>
    - Define o estado da máquina;
    - Comanda leituras, gravações e sentido de movimento (cabeça).

Máquina de Turing

- Inicialmente:
  - Palavra a ser processada:
    - Células mais à esquerda (após “marcador de início de fita”)
  - Demais células: “branco”.

- Unidade de controle
  - Número finito e predefinido de estados
  - A cabeça da fita:
    - Lê um símbolo de cada vez e grava um novo símbolo
    - Move uma célula para a direita ou para a esquerda
  - O programa define o símbolo gravado e o sentido do movimento.



[DEF] Máquina de Turing

- Máquina de Turing M:
  - $M = (\Sigma, Q, \delta, q_0, F, V, \beta, ❂)$

  - Σ - Alfabeto (de símbolos) de entrada;
  - Q - Conjunto de estados possíveis da máquina (finito);
  - δ - Função programa ou função de transição (função parcial);
    - Suponha que $\Sigma \cup V$ e { β, ❂ } são conjuntos disjuntos;

      δ: $Q \times (\Sigma \cup V \cup \{ \beta,$ ❂ $\}) \rightarrow Q \times (\Sigma \cup V \cup \{ \beta,$ ❂ $\}) \times \{ E, D \}$
    - Transição da máquina: $\delta(p, x) = (q, y, m)$

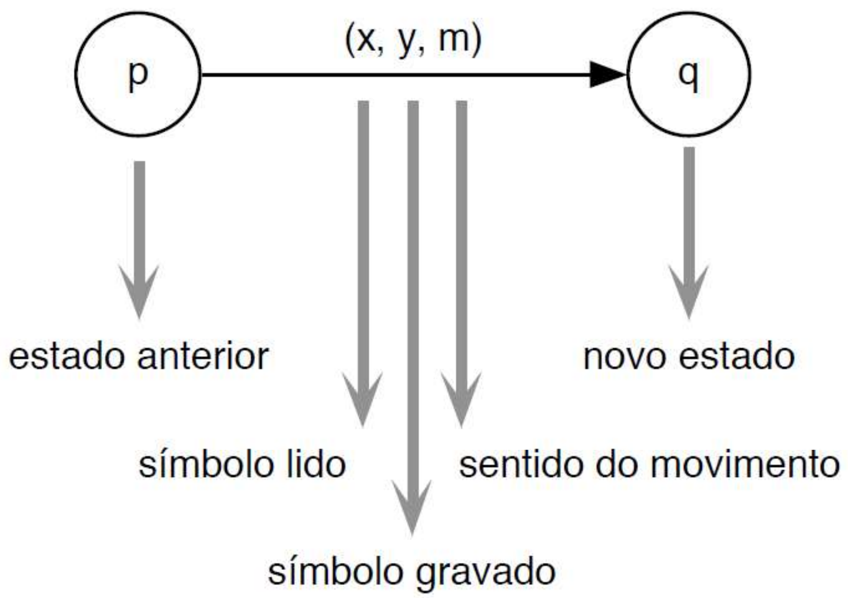
  - q_0 - Estado inicial: elemento distinguido de Q;
  - F - Conjunto de estados finais: subconjunto de Q;
  - V - Alfabeto auxiliar (pode ser vazio);
  - β - Símbolo especial branco;
  - ❂ - Símbolo de início ou marcador de início da fita.

[DEF] Máquina de Turing

- Símbolo de início de fita
  - • ocorre exatamente uma vez e na célula mais à esquerda da fita
- A Função programa...
  - Considera:
    - O estado corrente;
    - O símbolo lido da fita.
  - Determina:
    - O novo estado;
    - O símbolo a ser gravado;
    - Sentido de movimento da cabeça (E e D).

[DEF] Máquina de Turing

- Estados inicial e finai são como nos autômatos finitos;
  - Transição: δ(p, x) = (q, y, m))



Computação de uma entrada w por uma máquina de Turing

- Consiste da sucessiva aplicação da função programa:
  - A partir do estado inicial;
  - A cabeça posicionada na célula mais à esquerda da fita
    - Até ocorrer uma condição de parada.
  - Processamento pode parar ou ficar processando indefinidamente (ciclo ou *loop* infinito).
- **Aceita** a entrada w:
  - Atinge um estado final;
    - Máquina para e w é aceita
- **Rejeita** a entrada w:
  - Função programa é indefinida para o argumento (símbolo lido e estado corrente):
    - Máquina **para** e w é **rejeitada**.
  - Argumento define um movimento à esquerda, e a cabeça da fita já se encontra na célula mais à esquerda
    - Máquina **para** e w é **rejeitada**.

[DEF] Linguagens Aceita, Rejeitada, Loop

- Linguagem Aceita ou Linguagem Reconhecida por M:

  ACEITA(M) ou L(M)

  - Conjunto de todas as palavras de Σ^* aceitas por M, a partir de q_0.
- Linguagem Rejeitada por M:

  REJEITA(M)

  - Conjunto de todas as palavras de Σ^* rejeitadas por M, a partir de q_0.
- Linguagem Loop de M:

  LOOP(M)

  - Conjunto de todas as palavras de Σ^* para as quais M fica processando indefinidamente a partir de q_0.

[DEF] Linguagens Aceita, Rejeitada, Loop

- Cada máquina de Turing M sobre Σ
  - Induz uma partição de Σ* em:
    - ACEITA(M), REJEITA(M) e LOOP(M)

[EX] Máquina de Turing – Duplo balanceamento

- L = { $a^n b^n$ | n ≥ 0 }
- A Máquina de Turing M:

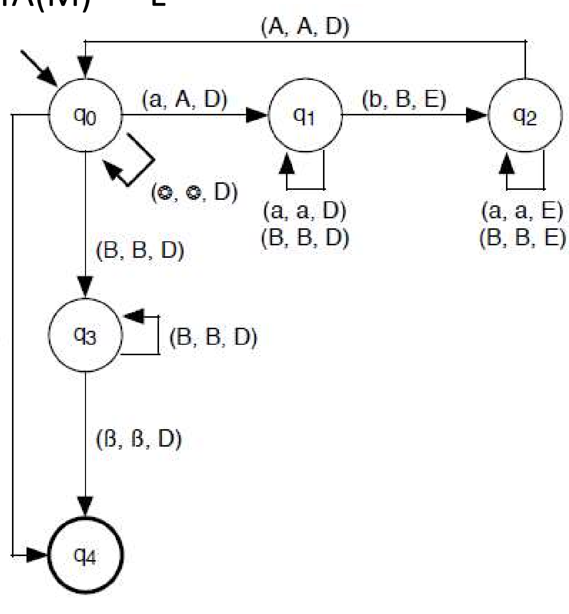
$$M = (\{ a, b \}, \{ q_0, q_1, q_2, q_3, q_4 \}, \delta, q_0, \{ q_4 \}, \{ A, B \}, \beta, \circledast)$$

- É tal que:

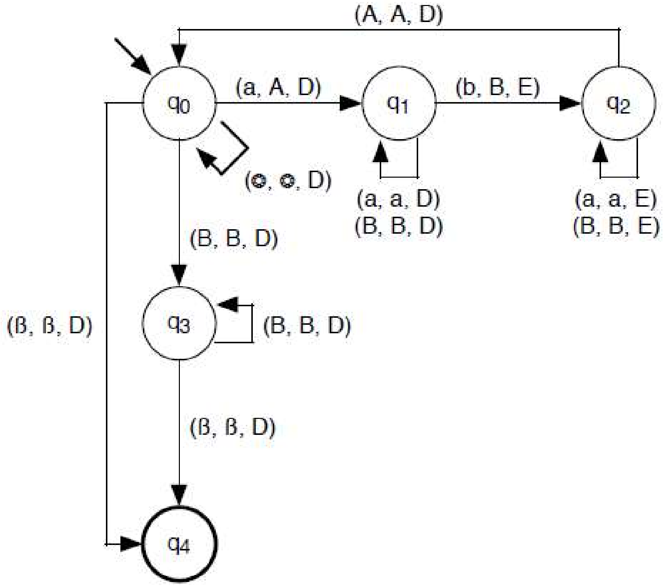
$$ACEITA(M) = L \text{ e } REJEITA(M) = \sim L$$

- Portanto, LOOP(M) = ∅
  - Qualquer palavra que não esteja na forma $a^x b^x$ é rejeitada

δ	$\circledast$	a	b	A	B	β
q_0	$(q_0, \circledast, D)$	(q_1, A, D)			(q_3, B, D)	(q_4, β, D)
q_1		(q_1, a, D)	(q_2, B, E)		(q_1, B, D)	
q_2		(q_2, a, E)		(q_0, A, D)	(q_2, B, E)	
q_3					(q_3, B, D)	(q_4, β, D)
q_4						



[EX] Máquina de Turing – Duplo balanceamento



[EX] Máquina de Turing – Duplo balanceamento

[OBS] Máquina de Turing Vs. Algoritmo

- Foi afirmado que Máquina de Turing:
  - É aceita como uma formalização do conceito de algoritmo.
- Entretanto, também é usual considerar como conceito de algoritmo:
  - A máquina de Turing que sempre para para qualquer entrada.
- Nesse caso, uma máquina que eventualmente fica em *loop* infinito:
  - *Não* seria considerada um algoritmo.

[FIM]

- FIM:
  - **[AULA 18]** Linguagens recursivamente enumeráveis – Máquina de Turing
- Próxima aula:
  - **[AULA 19]** Máquina de Turing – Modelos equivalentes