Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologias
Engenharia da Computação

Thales L. A. Valente

Disciplina: Linguagens Formais e Autômatos
Código: EECP0020

27 de junho de 2024

Conteúdo programático

- Elementos de matemática discreta
- Conceitos básicos de linguagens
- Linguagens regulares e autômatos finitos
- Linguagens livres de contexto e autômatos de pilha
- Linguagens sensíveis ao contexto e Máquinas de Turing com fita limitada
- Linguagens recursivas e Máquinas de Turing com finta infinita
- Linguagens recursivamente enumeráveis

Hierarquia de Chomsky

- De acordo com a complexidade relativa das linguagens, Chomsky definiu uma classificação que permite antecipar as propriedades fundamentais das linguagens e vislumbrar os modelos de implementação mais adequados. A **Hierarquia de Chomsky** é ilustrada abaixo:

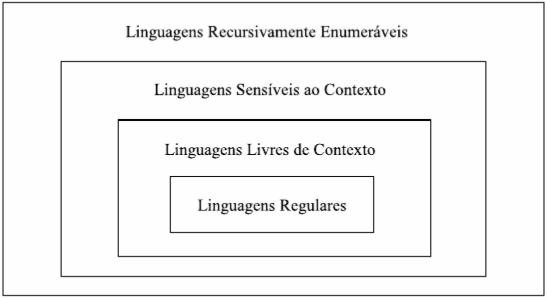


Figura: Hierarquia de Chomsky

Hierarquia de Chomsky

- Nesta apresentação, vai-se fazer uma inversão na ordem de estudo das classes de linguagens, para um melhor entendimento. A nova ordem é:

  1. Linguagens Recursivamente Enumeráveis ou do tipo 0
  2. Linguagens Recursivas
  3. Linguagens Sensíveis ao Contexto ou do tipo 1

Linguagens Recursivamente Enumeráveis ou do tipo 0

- O estudo de Linguagens Recursivamente Enumeráveis pode ser abordado a partir de dois formalismos básicos:
  - **Operacional ou reconhecedor**: representado pelas Máquinas de Turing com Fita Ilimitada.
  - **Axiomático ou gerador**: representado por Gramáticas Irrestritas.

Máquina de Turing com Fita Infinita

- Em 1936, Alan Turing propôs um modelo conhecido como **Máquina de Turing**, aceito hoje como uma formalização de algoritmos ou funções computáveis.
- Para tanto, algumas propriedades devem ser satisfeitas, tais como:
  - A descrição do algoritmo deve ser finita.
  - Os passos do algoritmo devem ser:
    - Discretos;
    - Executáveis mecanicamente e;
    - De tempo finito.

Máquina de Turing com Fita Infinita

- A Máquina de Turing possui três componentes básicos:

  1. **Fita**: usada simultaneamente como dispositivo de entrada, de saída e memória de trabalho. É infinita à direita.
  2. **Unidade de controle**: reflete o estado corrente da máquina. Possui cabeça de leitura/gravação que acessa uma célula da fita de cada vez e movimenta-se para a esquerda e para a direita.
  3. **Programa ou função de transição**: define as leituras/escritas, o sentido do movimento da cabebça e o estado da máquina.

Máquina de Turing com Fita Infinita

- Formalmente, uma **Máquina de Turing com Fita Infinita M** é uma 8-tupla:

$$M = \{\Sigma, Q, \delta, q_0, F, V, \beta, \circledast\}$$

onde:

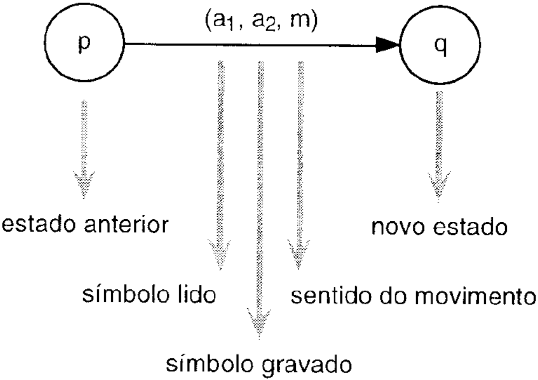
  - Σ: alfabeto de símbolos de entrada.
  - Q: conjunto finito de estados do autômato.
  - δ: função programa da forma

$$\delta : Q \times (\Sigma \cup V \cup \{\beta, \circledast\}) \rightarrow Q \times (\Sigma \cup V \cup \{\beta, \circledast\}) \times \{E, D\}$$

  - q_0: estado inicial, tal que $q_0 \in Q$.
  - F: conjunto de estados finais, tal que $F \subset Q$.
  - V: alfabeto auxiliar (pode ser vazio).
  - β: símbolo especial branco.
  - $\circledast$: símbolo ou marcador de início da fita.

Máquina de Turing com Fita Infinita

- Graficamente, a função programa pode ser representada como:

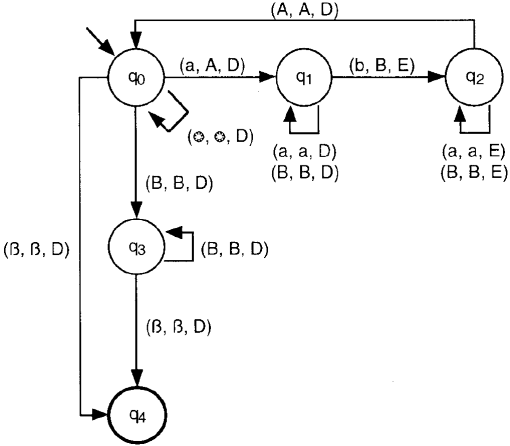


Máquina de Turing com Fita Infinita

- Uma Máquina de Turing com Fita Infinita pode parar aceitando ou rejeitando a entrada ou ficar em um *loop* infinito. As condições de paradas são as seguintes:
  - A máquina assume um estado final: a máquina pára e a palavra é **aceita**.
  - A função programa é indefinida para o argumento (símoblo lido e estado corrente): a máquina pára e a palavra é **rejeitada**.
  - O argumento corrente da função define u m movimento à esquerda quando a cabeça já está na célula mais à esquerda: a máquina pára e a palavra é **rejeitada**.

Máquina de Turing com Fita Infinita

- **Exemplo**: A linguagem $L = \{a^n b^n | n \geq 0\}$ é reconhecida pela seguinte Máquina de Turing:



Máquina de Turing com Fita Infinita

- A **Hipótese de Church** ou **Hipótese de Turing-Church** pode ser assim enunciada:
  - A capacidade de computação representada pela Máquina de Turing é o limite máximo que pode ser atingido por qualquer dispositivo de computação.

Máquina de Turing com Fita Infinita

- Corroboram com a Hipótese de Turing o fato de todos os demais modelos de computação propostos possuem, no máximo, o mesmo poder computacional da Máquina de Turing. Exemplos de modelos incluem:
  - Autômato do Múltiplas Pilhas;
  - Máquina de Turing não-determinísitica;
  - Máquina de Turing com Fita Infinita à Esquerda e à Direita;
  - Máquina de Turing com Múltiplas Fitas;
  - Máquina de Turing Multidimensional;
  - Máquina de Turing com Múltiplas Cabeças e;
  - Combinações de Modificações sobre a Máquina de Turing.

Máquina de Turing com Fita Infinita

- **Definição**: Uma linguagem aceita por uma Máquina de Turing é dita **Liguagem Recursivamente Enumerável** ou do **Tipo 0**.
    - O termo enumerável indica que qualquer palavra da linguagem pode ser listada por uma Máquina de Turing. E essa enumeração é feita de forma recursiva.
- **Exemplos**:
    - $\{a^n b^n | n \geq 0\}$
    - $\{w | w$ tem o mesmo número de símbolos a e $b\}$
    - $\{a^i b^j c^k | i = j$ ou $j = k\}$

Máquina de Turing com Fita Infinita

- Existem Linguagens Recursivamente Enumeráveis para as quais alguma palavra fica em *loop* ao ser processada por uma Máquina de Turing.
- As linguagens desta classe que não apresentam tal característica são chamadas de **Linguagens Recursivas**. Ou seja, L é uma Linguagem Recursiva se:
  - $ACEITA(M) = L$
  - $REJEITA(M) = \Sigma^* - L$
- **Exemplos**:
  - $\{a^n b^n | n \geq 0\}$
  - $\{a^n b^n c^n | n \geq 0\}$
  - $\{w | w \in \{a, b\}^*$ e tem o dobro de símbolos a que $b\}$

Gramática Irrestrita

- Ao contrário das gramáticas GR e GLC apresentadas anteriormente, uma **Gramática Irrestrita ou do Tipo 0** é aquele em que não há restrições sobre o formato de suas regras.
- **Exemplo**: A liniguagem $L = \{a^n b^n c^n\}$ é gerada pela seguinte Gramática Irrestrita:

$$G = (\{S < C\}, \{a, b, c\}, P, S),$$

onde $P = \{S \rightarrow abc|\epsilon, ab \rightarrow aabbC, Cb \rightarrow bC, Cc \rightarrow cc\}$

  - Como seria a derivação da palavra *aaabbbccc*?

Gramática Irrestrita

- Como teorema, tem L é um Linguagem Recursivamente Enumerável se, e somente se, L é gerada por uma Gramática Irrestrita.

Linguagens Sensíveis ao Contexto ou do Tipo 1

- O estudo de Linguagens Sencíveis ao Contexto pode ser abordado a partir de dois formalismos básicos:
  - **Operacional ou reconhecedor**: representado pelas Máquinas de Turing com Fita Limitada.
  - **Axiomático ou gerador**: representado por Gramáticas Sensíveis ao Contexto.

Gramáticas Sensíveis ao Contexto

- **Uma Gramática Sensível ao Contexto $\boldsymbol{G}$** é uma gramática $G = \{V, T, P, S\}$ com a restriição de que qualquer regra de P é da forma $\alpha \to \beta$, onde:
  - α é uma palavra de $(V \cup T)^+$
  - β é uma palavra de $(V \cup T)^*$
  - $|\alpha| \leq |\beta|$, excetuando-se, eventualmente, para $S \to \epsilon$.

- Uma linguagem gerada por uma Gramática Sensível ao Contexto é dita **Linguagem Sensível ao Contexto ou do Tipo 1**.

Gramáticas Sensíveis ao Contexto

- **Exemplo**: A linguagem $L = \{ww | w$ é palavra de $\{a, b\}^*\}$ pode ser gerada pela seguinte gramática:

$$G = (\{S, X, Y, A, B, \langle aa\rangle, \langle ab\rangle, \langle ba\rangle, \langle bb\rangle\}, \{a, b\}, P, S), \text{ onde:}$$

$$
\begin{aligned}
P = \{\, & S \to XY \mid aa \mid bb \mid \varepsilon, \\
& X \to XaA \mid XbB \mid aa\langle aa\rangle \mid ab\langle ab\rangle \mid ba\langle ba\rangle \mid bb\langle bb\rangle, \\
& Aa \to aA,\ Ab \to bA,\ AY \to Ya, \\
& Ba \to aB,\ Bb \to bB,\ BY \to Yb, \\
& \langle aa\rangle a \to a\langle aa\rangle,\ \langle aa\rangle b \to b\langle aa\rangle,\ \langle aa\rangle Y \to aa, \\
& \langle ab\rangle a \to a\langle ab\rangle,\ \langle ab\rangle b \to b\langle ab\rangle,\ \langle ab\rangle Y \to ab, \\
& \langle ba\rangle a \to a\langle ba\rangle,\ \langle ba\rangle b \to b\langle ba\rangle,\ \langle ba\rangle Y \to ba, \\
& \langle bb\rangle a \to a\langle bb\rangle,\ \langle bb\rangle b \to b\langle bb\rangle,\ \langle bb\rangle Y \to bb\,\}
\end{aligned}
$$

Máquina de Turing com Fita Limitada

- Uma **Máquina de Turing com Fita Limitada** é basicamente uma Máquina de Turing com a fita limitada ao tamanho da palavra de entrada mais duas células contendo marcadores de início e fim da fita.
- Formalmente, uma Máquina de Turing com Fita Limitada $\boldsymbol{M}$ é uma 8-tupla:

$$M = \{\Sigma, Q, \delta, q_0, F, V, \circledast, \dagger\}$$

onde:

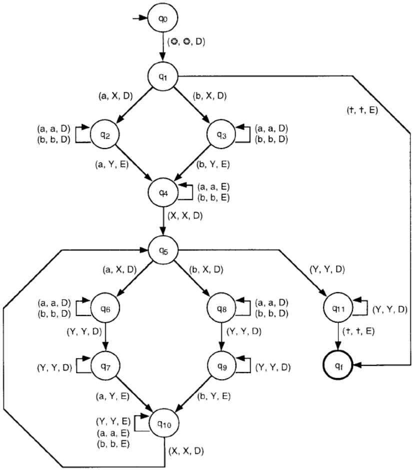
- Σ: alfabeto de símbolos de entrada.
- Q: conjunto finito de estados do autômato.
- δ: função programa da forma

$$\delta : Q \times (\Sigma \cup V \cup \{\circledast, \dagger\}) \rightarrow 2^{Q \times (\Sigma \cup V \cup \{\circledast, \dagger\}) \times \{E, D\}}$$

- q_0: estado inicial, tal que $q_0 \in Q$.
- F: conjunto de estados finais, tal que $F \subset Q$.
- V: alfabeto auxiliar (pode ser vazio).
- $\circledast$: símbolo ou marcador de início da fita.
- $\dagger$: símbolo ou marcador de fim da fita.

Máquina de Turing com Fita Infinita

- **Exemplo**: A linguagem $L = \{ww | w \text{ é palavra de } \{a, b\}^*\}$ é reconhecida pela seguinte Máquina de Turing com Fita Limitada:



Resumo do Curso

- A associação entre linguagens, gramáticas e reconhecedores é destacada na tabela abaixo:

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

Figura: Linguagens, gramáticas e reconhecedores

Bibliografia

1. MENEZES, Paulo B. Linguagens formais e autômatos. 6ª ed. Porto Alegre: Bookman, 2011.
    - **Capítulos 6**.

Dúvidas?