

CIÊNCIAS DA COMPUTAÇÃO

Aspectos Teóricos da Computação

CIÊNCIAS DA COMPUTAÇÃO

Aspectos Teóricos da Computação

ROTEIRO

- Hierarquia de Chomsky
- Máquina de Estado Finito
- Conceitos da Teoria de Autômatos
- Máquina de Turing

3

Hierarquia de Chomsky (1928-...)

- Noam Chomsky, professor emérito MIT, é considerado o Pai da Linguística Moderna.
- Enunciou propriedades fundamentais das linguagens formais e, sobre elas, concebeu uma hierarquia.
- Estas propriedades são fundamentais para o desenvolvimento de compiladores, conhecidos como Linguagem Livre de Contexto.



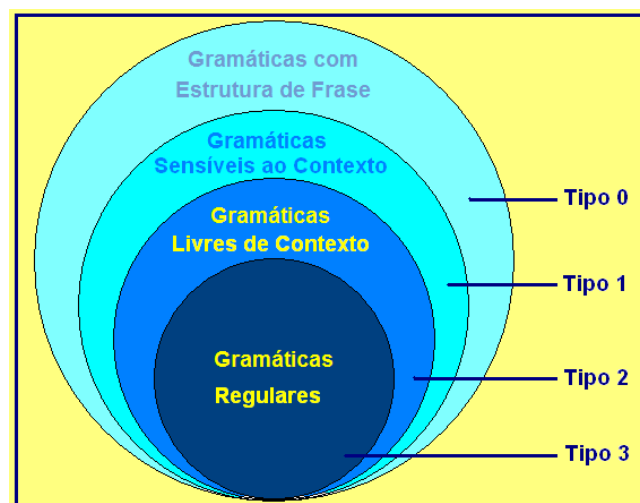
4

Hierarquia de Chomsky (1928-...)

- Gramáticas divididas em quatro classes hierarquicamente organizadas.
- São numeradas de forma decrescentes, da mais simples para a mais complexa: 3, 2, 1 e 0.
- 3 é classe mais restrita e 0 é a classe mais abrangente.
- A classe N gera um conjunto mais amplo de linguagens que a $N+1$ ($0 \leq N \leq 2$).

5

Hierarquia de Chomsky (1928-...)



https://upload.wikimedia.org/wikipedia/commons/a/a3/Hierarquia_de_Chomsky.PNG

6

Hierarquia de Chomsky (1928-...)

- Nomenclatura:
 - Tipo 0: Gramáticas com Estrutura de Frase (GEFs)
 - Tipo 1: Gramáticas Sensíveis ao Contexto (GSCs)
 - Tipo 2: Gramáticas Livres de Contexto (GLCs)
 - Tipo 3: Gramáticas Regulares (GRs)

7

Hierarquia de Chomsky

- Tipo 0: Gramáticas com Estrutura de Frase (GEFs)
 - Nenhuma restrição é imposta à gramática.
 - Linguagens Geradas:
 - Linguagens Recursivamente Enumeráveis (LREs); e
 - Linguagens Recursivas (LRs).
 - Ex. a partir de GEF é possível especificar um subconjunto de um idioma português, inglês, etc.).

8

Hierarquia de Chomsky

- Tipo 1: Gramáticas Sensíveis ao Contexto (GSCs)
 - Restrição:

“ no processo de substituição a ser aplicado sobre uma determinada forma sequencial da gramática, esta substituição não pode reduzir de comprimento da forma sequencial.”
 - As produções da gramática aumentam ou mantém o comprimento da forma sentencial sobre a qual é aplicada.
 - Ex.: Linguagem de todas as cadeias consistindo em n ocorrências do símbolo "a", e n "b"s, e n "c"s (abc, aabbcc, aaabbbccc, etc.).

9

Hierarquia de Chomsky

- Tipo 2: Gramáticas Livres de Contexto (GLCs)
 - Adicionalmente à restrição às GSC:

$A \rightarrow \beta$, onde $A \in (V \cup T)^*$, onde:

 - A : variável da gramática;
 - β : forma sentencial gerada a partir de A ;
 - V : conjunto de variáveis da gramática; e
 - T : conjunto de símbolos terminais da gramática.
 - Ex.: Linguagens de programação são geradas a partir de uma GLC.

10

Hierarquia de Chomsky

- Tipo 3: Gramáticas Regulares (GRs)
 - As produções da gramática devem gerar expressões regulares.
 - As linguagens correspondentes são denominadas de linguagens regulares geradas por gramáticas regulares.
 - Ex.:
 - a) todas cadeias que contenham um número par de 'a';
 - b) todas cadeias formadas por uma quantidade qualquer de 'a' seguido de uma quantidade qualquer de 'b'; e
 - c) todas cadeias de 'a' seguido de 'b', onde o número de 'a' é igual ao de 'b'.

11

Hierarquia de Chomsky

- Tipo 3: Gramáticas Regulares (GRs)
 - As produções da gramática devem gerar expressões regulares.
 - As linguagens correspondentes são denominadas de linguagens regulares geradas por gramáticas regulares.
 - Ex.:
 - a) todas cadeias que contenham um número par de 'a';
 - b) todas cadeias formadas por uma quantidade qualquer de 'a' seguido de uma quantidade qualquer de 'b';
 - c) todas cadeias de 'a' seguido de 'b', onde o número de 'a' é igual ao de 'b';
 - **(c) FALSO!!! - não pode ser considerado uma linguagem regular pois requer a atuação de uma memória para estruturar os elementos de suas cadeias, isto é, quando a frequência de um elemento da cadeia determina a frequência de outro elemento da mesma cadeia.**

12

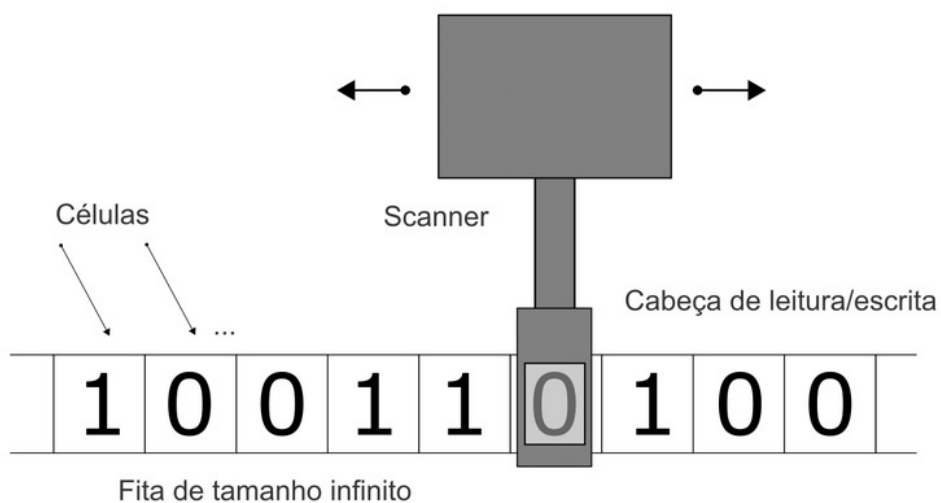
Hierarquia de Chomsky

Tabela 1 – Resumo da Hierarquia de Chomsky.

Hierarquia de Chomsky	Gramática	Classe da Linguagem	Nome do Reconhecedor
Tipo 0	Irrestrita	Recursivamente Enumerável	Máquina de Turing (MT)
—	—	Recursiva	MT que sempre para
Tipo 1	Sensível ao Contexto	Sensível ao Contexto	Autômato Linearmente Limitado
Tipo 2	Livre de Contexto	Livre de Contexto	Autômato de Pilha
Tipo 3	Regular	Regular	Autômato Finito

13

Máquina de Turing



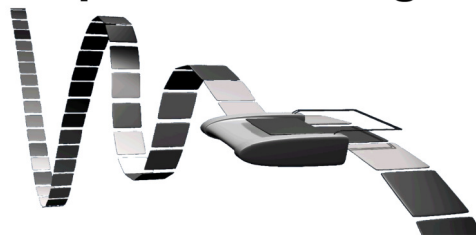
14

Máquina de Turing

- Uma fita que é dividida em células, uma adjacente à outra.
- Cada célula contém um símbolo de algum alfabeto finito.
- O alfabeto contém um símbolo especial branco (aqui escrito como \neg) e um ou mais símbolos adicionais.
- Assume-se que a fita é arbitrariamente extensível para a esquerda e para a direita, isto é, a máquina de Turing possui tanta fita quanto é necessário para a computação.
- Assume-se também que células que ainda não foram escritas estão preenchidas com o símbolo branco.

15

Máquina de Turing

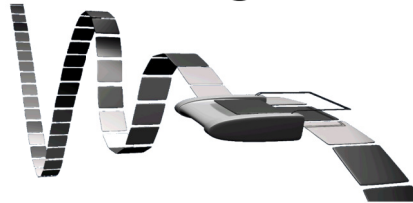


- Um cabeçote, que pode ler e escrever símbolos na fita e mover-se para a esquerda ou para a direita.
- Um registrador de estados, que armazena o estado da máquina de Turing.
 - O número de estados diferentes é sempre finito e há um estado especial denominado estado inicial com o qual o registrador de estado é inicializado.

16

Aspectos Teóricos da Computação

Máquina de Turing



- Uma tabela de ação (ou função de transição) que diz à máquina que símbolo escrever, como mover o cabeçote (\leftarrow para esquerda e \rightarrow para direita) e qual será seu novo estado, dados o símbolo que ele acabou de ler na fita e o estado em que se encontra.
- Se não houver entrada alguma na tabela para a combinação atual de símbolo e estado então a máquina para.

17

Aspectos Teóricos da Computação

Máquina de Turing

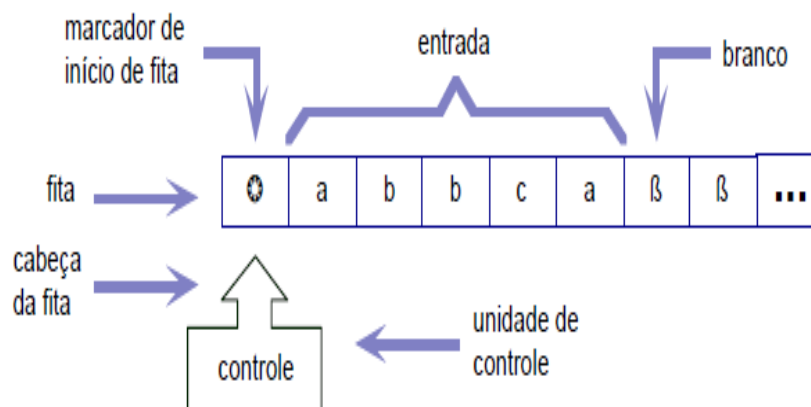


Figura 3.3 Fita e unidade de controle de uma Máquina de Turing

18

Máquina de Turing

Fita

- Usada simultaneamente como dispositivo de **entrada, de saída e de memória de trabalho**;
- É finita à esquerda e infinita (tão grande quanto necessário) à direita, sendo dividida em células, cada uma das quais armazenando um símbolo.
- Os **símbolos** podem pertencer:
 - ⇒ ao alfabeto de entrada,
 - ⇒ ao alfabeto auxiliar,
 - ⇒ **␣** branco, e
 - ⇒ **⦿** marcador de início de fita.
- Inicialmente, a palavra a ser processada ocupa as células mais à esquerda, após o marcador de início de fita, ficando as demais com branco.

19

Máquina de Turing

Unidade de Controle:

- Reflete o estado corrente da máquina.
- Possui um número finito e predefinido de estados.
- Possui uma unidade de leitura e gravação (*cabeça da fita*), a qual acessa uma célula da fita de cada vez.
- A *cabeça da fita* lê o símbolo de uma célula de cada vez e grava um novo símbolo.
- Após a leitura/gravação (a gravação é realizada na mesma célula de leitura), a cabeça move-se uma célula para a direita ou esquerda.

20

Máquina de Turing

• Programa ou Função de Transição

- o programa comanda as leituras e gravações, o sentido de movimento da cabeça e define o estado da máquina.
- programa é uma função que, dependendo do estado corrente da máquina e do símbolo lido, determina o símbolo a ser gravado, o sentido do movimento da cabeça e o novo estado.

21

Máquina de Turing

• Processamento

- O processamento de uma MT para uma palavra de entrada w consiste na sucessiva aplicação da função programa, a partir do estado inicial q_0 e da cabeça posicionada na célula mais à esquerda da fita até ocorrer uma condição de parada.
- O processamento de M para a entrada w pode parar ou ficar em um laço infinito.
- A parada pode ser de duas maneiras: aceitando ou rejeitando a entrada w .

22

Máquina de Turing

▪ Variações

- **Inexistência do Marcador de Início de Fita:**

- É frequente não incluir um marcador de início de fita. Assim, a célula mais à esquerda da fita contém o primeiro símbolo da entrada (ou branco, se a entrada for vazia). Neste caso, ao definir uma função programa, deve-se tomar cuidado especial para controlar quando a cabeça da fita atinge o fim da mesma;

- **Cabeça de Fita não se Move em uma Leitura/Gravação:**

- Na função programa, é possível especificar, adicionalmente ao movimento para esquerda ou direita, que a cabeça permaneça parada (na célula de leitura/gravação). O principal objetivo dessa variação é facilitar a especificação da função programa, bem como reduzir o número de transições necessárias.

23

Conceitos da Teoria de Autômatos

- **Alfabeto: Σ**

- Conjunto de símbolos finito e não vazio.

- Exemplos:

- Alfabeto binário

- ✓ $\Sigma = \{0, 1\}$

- Alfabeto romano

- ✓ $\Sigma = \{a, b, c, \dots, z\}$

24

Conceitos da Teoria de Autômatos

- String (ou palavra):
 - é uma sequência finita de símbolos escolhidos de algum alfabeto.
 - Ex.: Seja o alfabeto $\Sigma = \{0,1\}$. São strings:
 - 01101 e 111.
- String vazio (ϵ):
 - É o string com zero ocorrência de símbolos / tem comprimento zero.
- Tamanho ou comprimento de uma string $w = |w|$:
 - Número de posições para símbolos no string.
 - Ex.: $|0011|=4$ e $|000|=3$ e $|\epsilon|=0$.

25

Conceitos da Teoria de Autômatos

- Prefixo
 - é qualquer sequência inicial de símbolos da palavra.
- Sufixo
 - é qualquer sequência final de símbolos da palavra.
- Subpalavra (substring) de uma palavra
 - é qualquer sequência de símbolos contíguos da palavra.
- Exemplo: abcb é uma palavra sobre o alfabeto $\{a, b, c\}$. Sobre esta palavra:
 - Prefixos:
 - ϵ , a, ab, abc, abcb
 - Sufixos:
 - ϵ , b, cb, bcb, abcb
 - Subpalavra
 - Qualquer prefixo ou sufixo de uma palavra é uma subpalavra.

26

Conceitos da Teoria de Autômatos

- Prefixo próprio:
 - um prefixo que não seja a palavra inteira
 - Exemplo:
 - "abc" é um prefixo próprio de "abcdef"
- Sufixo próprio:
 - um sufixo que não seja a palavra inteira
 - Exemplo:
 - "def" é um sufixo próprio de "abcdef"

27

Máquina de Estado Finito

- Potência de um alfabeto Σ^k :
 - Define-se Σ^k como o conjunto de strings de comprimento k , onde o símbolo de cada um deles está em Σ .
 - Exemplo: Seja $\Sigma = \{0, 1\}$. $\Sigma^1 = ?$; $\Sigma^2 = ?$
 - $\Sigma^1 = \{0, 1\}$.
 - $\Sigma^2 = \{00, 01, 10, 11\}$.
 - $\Sigma^3 = ?$
- Σ^* : conjunto de todos os strings de um alfabeto.
 - $\{0, 1\}^* = \{ \epsilon, 0, 1, 00, 01, 10, 11, 000, \dots \}$
 - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots$

28

Conceitos da Teoria de Autômatos

- Alfabeto Σ^+ :
 - $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \dots$
 - $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$
- Concatenação de Strings
 - Sejam as strings $x = a_1a_2a_3\dots a_i$ e $y = b_1b_2b_3\dots b_j$, onde $|x|=i$ e $|y|=j$. Assim, a string concatenada de x e y , $x \circ y$, de tamanho $i+j$ é: $x \circ y = a_1a_2a_3\dots a_ib_1b_2b_3\dots b_j$.
 - Ex.: $x = 011101$ e $y = 101$. $x \circ y = ?$ e $y \circ x = ?$
 - Resp.: $x \circ y = 011101101$ e $y \circ x = 101011101$.

29

Conceitos da Teoria de Autômatos

- Linguagens: conjunto de palavras
 - Seja Σ um alfabeto e sobre este um conjunto de palavras é escolhido a partir de Σ^* . Diz-se que L é uma linguagem sobre Σ , se $L \subseteq \Sigma^*$.
 - Uma linguagem sobre Σ não precisa incluir palavras com todos os símbolos do alfabeto Σ .
 - Língua portuguesa
 - $\Sigma = \{a, à, á, ã, b, c, d, e, \dots, v, w, x, y \text{ e } z\}$
 - Linguagem C:
 - $L \subseteq \{\text{subconjunto dos caracteres ASCII}\}$
 - Diz-se que uma **linguagem** é **regular** se existir um autômato finito que a reconhece.

30

Conceitos da Teoria de Autômatos

- Exercícios Linguagens:

1) Qual é a linguagem de todas as palavras que consistem em n 0's seguidos por n valores 1's, para algum $n \geq 0$?

Resp.: $L = \{ \epsilon, 01, 0011, 000111, \dots \}$.

2) Qual é a linguagem de todas as palavras que consistem de números binários que sejam primos?

Resp.: $L = \{ 10, 11, 101, 111, 1011, \dots \}$

Conceitos da Teoria de Autômatos

- Exercícios Linguagens:

- 3) Suponha o alfabeto $\Sigma = \{ a, b \}$. Então, o conjunto de palíndromos (palavras que têm a mesma leitura da esquerda para a direita e vice-versa) sobre Σ é um exemplo de linguagem infinita. Assim, são palavras desta linguagem:

- Resp.: $\epsilon, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, \dots$