Aula 4: Linguagens Regulares.

Prof. Lucio A. Rocha

Engenharia de Computação
Universidade Tecnológica Federal do Paraná, UTFPR
Câmpus Apucarana, Brasil

1º semestre / 2023

Sumário

Seção 1

Introdução

Linguagens Regulares

- Linguagens Regulares (Tipo 3):
  - Gramática Regular ✓
    - Formalismo axiomático (gerador)
    - Gramática com restrições das regras de produção de sentenças
  - Expressão Regular ✓
    - Formalismo denotacional (gerador)
    - Conjuntos básicos, concatenação, alternativa, repetição
  - Autômato Finito
    - Formalismo operacional (reconhecedor)
    - Conjunto de estados finitos.

Linguagens Regulares

- Na hierarquia de Chomsky:
  - Classe de linguagens mais simples
  - Algoritmos de reconhecimento, geração ou conversão
    - Pouca complexidade
    - Grande eficiência
  - Linguagens de programação em geral são não-regulares.
  - Principal aplicação: análise léxica.

Seção 2

Autômato Finito

Autômato Finito

- É um sistema de estados pré-definidos e finitos.
- O autômato possui:
  - Estados
  - Transições
-

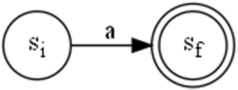


Figura: Autômato Finito.

Autômato Finito

- Autômato Finito é descrito por uma quíntupla:

$$M = (\Sigma, Q, \delta, q_0, F)$$

  - Σ: alfabeto (finito) de entrada.
  - Q: conjunto finito de estados.
  - δ: conjunto de transições (função parcial, função de transição ou programa)

$$\delta : Qx\Sigma \rightarrow Q$$

  - q_0: estado inicial ($q_0 \in Q$)
  - F: conjunto de estados finais. (F $\subseteq$Q)

Autômato Finito

- Função de transição:

$$\delta(p, a) = q$$

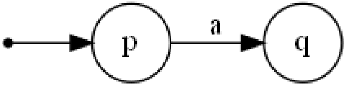


Figura: Autômato Finito.

  - p: estado anterior.
  - a: símbolo lido.
  - q: novo estado.

Autômato Finito

- Função de transição:

$$\delta(p, a) = q$$

-

	p	q
a	q	...
...	...	...

  - p: estado anterior.
  - a: símbolo lido.
  - q: novo estado.

Autômato Finito

- Computação de um autômato finito:
    - Aplicação sucessiva da função de transição para cada símbolo de entrada
    - *Sentença válida*:
        - A função de transição alcançou estado final
    - *Sentença não-reconhecida*:
        - A função de transição terminou a leitura em estado que não é final
        - A função de transição não possui transição para um estado com o símbolo da sentença.
    - Não há sentença inválida: apenas não é reconhecida pela linguagem.

Autômato Finito

- Exemplo: Reconhecer a palavra **010**
  -

δ	S_i	S_n	S_a
0	S_a	S_a	S_a
1	S_n	S_n	S_n

  - Σ: $\{0, 1\}$
  - Q: $\{S_i, S_a, S_n\}$
  - q_0: S_i
  - F: $\{S_a\}$
- Computação da palavra:
  - $\delta(S_i, 0) = S_a$
  - $\delta(S_a, 1) = S_n$
  - $\delta(S_n, 0) = S_a$
  - $S_a \in F$. Então, palavra é válida.

Autômato Finito

- Exemplo: Reconhecer **aa** ou **bb** como subpalavra.
- $L = \{w \mid w \text{ possui } \textbf{aa} \text{ ou } \textbf{bb} \text{ como subpalavra}\}$
- $M = (\{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta, q_0, \{q_f\})$
-

δ	q_0	q_1	q_2	q_f
a	q_1	q_f	q_1	q_f
b	q_2	q_2	q_f	q_f

Autômato Finito

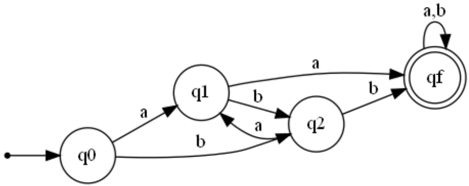
- 

Figura: Autômato Finito.

  - q_1: símbolo anterior é **a**.
  - q_2: símbolo anterior é **b**.
  - q_0: estado inicial.
  - q_f: estado final.
- Pergunta: a palavra **abba** é reconhecida por esse autômato?

Autômato Finito

- Autômato finito sempre pára
  - Palavra é finita
  - Novo símbolo é lido a cada aplicação da função de transição
  - Não existe a possibilidade de loop infinito
- Parada do processamento
  - Palavra válida:
    - A função de transição alcançou estado final
  - Palavra não-reconhecida:
    - Função de transição terminou a leitura em estado que não é final
    - Função de transição não possui transição para um estado com o símbolo da palavra.
    - Não há palavra inválida: apenas não é reconhecida pela linguagem.

Autômato Finito

- Função programa estendida
  - $\delta^* : Qx\Sigma^* \to Q$

  é a função estendida de

  $$\delta : Qx\Sigma \to Q$$

  para reconhecimento de palavras.
  - $\delta^*(q, \varepsilon) = q$

  $$\delta^*(q, aw) = \delta^*(\delta(q, a), w)$$

Autômato Finito

- Exemplo:

$$M = (\{a, b\}, \{q_0, q_1, q_2, q_f\}, \delta, q_0, \{q_f\})$$

- A computação da sentença **abaa**:
  - $\delta^*(q_0, abaa) = \delta^*(\delta(q_0, a), baa) =$
    $\delta^*(q_1, baa) = \delta^*(\delta(q_1, b), aa) =$
    $\delta^*(q_2, aa) = \delta^*(\delta(q_2, a), a) =$
    $\delta^*(q_1, a) = \delta^*(\delta(q_1, a), \varepsilon) =$
    $\delta^*(q_f, \varepsilon) = q_f$
- Portanto, a palavra é aceita.

Seção 3

Linguagens Regulares

Linguagens Regulares

- Σ : alfabeto
- Σ^*: é uma partição de todas as palavras do alfabeto.
ACEITA(M)	REJEITA(M)
- $\Sigma^* = \{\{ACEITA(M)\}, \{REJEITA(M)\}\}$

Linguagens Regulares

- Def.: Linguagem ACEITA pelo autômato finito determinístico (AFD).
- Dado o AFD definido por

$$M = (\Sigma, Q, \delta, q_0, F)$$

, a linguagem ACEITA por M, definida por:

$$ACEITA(M) = L(M)$$

, é o conjunto de todas as palavras de Σ^* aceitas por M a partir do estado inicial q_0, ou seja,

$$ACEITA(M) = L(M) = \{w \mid \delta^*(q_0, w) \in F\}$$

Linguagens Regulares

- Def.: Linguagem REJEITADA pelo autômato finito determinístico (AFD).
- Dado o AFD definido por

$$M = (\Sigma, Q, \delta, q_0, F)$$

, a linguagem REJEITADA por M, definida por:

$$REJEITA(M)$$

, é o conjunto de todas as palavras de Σ^* rejeitadas por M a partir do estado inicial q_0, ou seja,

$$REJEITA(M) = \{w \mid \delta^*(q_0, w) \notin F\}$$
$$REJEITA(M) = \{w \mid \delta^*(q_0, w) \text{ é indefinida } \}$$

Linguagens Regulares

Def.: Linguagem Regular (ou Linguagem Tipo 3)

- L é uma Linguagem Regular se
existe pelo menos um AFD que aceita L, ou seja,
ACEITA(M)=L

- Diferentes autômatos finitos podem aceitar uma mesma linguagem.
- Def.: Autômatos Finitos Equivalentes
M1 e M2 são equivalentes se, e somente se:

$$\text{ACEITA}(M1) = \text{ACEITA}(M2)$$

Linguagens Regulares

- Exemplo: Considere a linguagem L_1 sobre o alfabeto $\Sigma = \{a, b\}$
- $L_1 = \emptyset = \{\}$
- $M_1 = (\{a, b\}, \{q_0\}, \delta_1, q_0, \{\})$
- $\delta_1 = \{$
  $\delta(q_0, a) = q_0,$
  $\delta(q_0, b) = q_0\}$
- $ACEITA(M_1) = L_1$. Portanto, L_1 é uma linguagem regular.

Linguagens Regulares

- Exemplo: Considere a linguagem L_2 sobre o alfabeto $\Sigma = \{a, b\}$
- $L_2 = \Sigma^* = \{\varepsilon, a, aa, ab, bb, aabb, ...\}$
- $M_2 = (\{a, b\}, \{q_0\}, \delta_2, q_0, \{q_0\})$
- $\delta_2 = \{$
  $\delta(q_0, a) = q_0,$
  $\delta(q_0, b) = q_0\}$
- $ACEITA(M_2) = L_2$. Portanto, L_2 é uma linguagem regular.

Linguagens Regulares

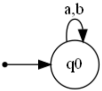
- Pergunta: M_1 é equivalente a M_2?
- 

Figura: M_1.

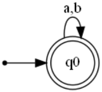


Figura: M_2.

Linguagens Regulares

- Exemplo:
- $L_1 = \{w \mid$ w possui número ímpar de **a** e **b** $\}$
- $M_1 = (\{a, b\}, \{q_0, q_1, q_2, q_3, q_4\}, \delta_1, q_0, \{q_2, q_4\})$
- $\delta_1 = \{$
  $\delta(q_0, a) = q_1,$
  $\delta(q_0, b) = q_3,$
  $\delta(q_1, a) = q_0,$
  $\delta(q_1, b) = q_2,$
  $\delta(q_2, b) = q_1,$
  $\delta(q_3, b) = q_0,$
  $\delta(q_3, a) = q_4,$
  $\delta(q_4, a) = q_3\}$
- $ACEITA(M_1) = L_1$. Portanto, L_1 é uma linguagem regular.

Linguagens Regulares

- Exemplo:
- $L_2 = \{w \mid$ w possui número ímpar de **a** e **b** $\}$
- $M_2 = (\{a, b\}, \{q_0, q_1, q_2, q_3\}, \delta_2, q_0, \{q_2\})$
- $\delta_2 = \{$
  $\delta(q_0, a) = q_1,$
  $\delta(q_0, b) = q_3,$
  $\delta(q_1, a) = q_0,$
  $\delta(q_1, b) = q_2,$
  $\delta(q_2, b) = q_1,$
  $\delta(q_3, b) = q_0,$
  $\delta(q_3, a) = q_2,$
  $\delta(q_2, a) = q_3\}$
- $ACEITA(M_2) = L_2$. Portanto, L_2 é uma linguagem regular.

Linguagens Regulares

- Pergunta: M_1 é equivalente a M_2?
-

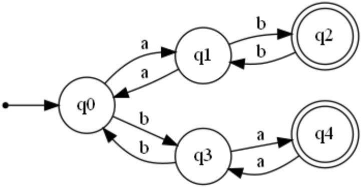


Figura: M_1.

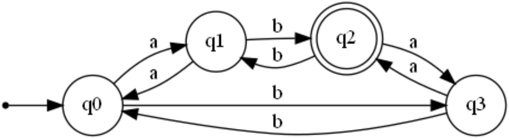
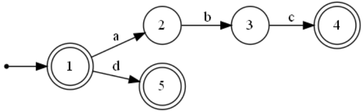


Figura: M_2.

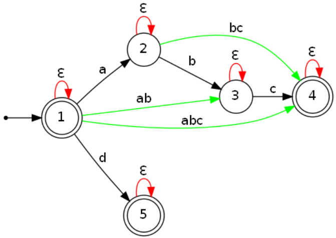
Linguagens Regulares

- Computações x Caminhos de um Grafo
  - Conjunto de arcos: todas as computações possíveis.
  - Subconjunto de arcos:
    - Com origem no estado inicial.
    - Destino em um estado final.
    - Linguagem aceita.

Linguagens Regulares



(a) Autômato M.



(b) Computação sobre M.

Figura: Autômato x Caminhos de um Grafo.

- $ACEITA(M) = \{\varepsilon, d, abc\}$
- $COMPUTACAO(M) = \{\varepsilon, a, b, c, d, ab, bc, abc\}$

Linguagens Regulares

- Linguagem gerada: Seja $G = (V_T, V_N, \mathbb{P}, S_i)$ uma gramática. A linguagem gerada é:

$$L(G) = GERA(G)$$
$$L(G) = \{w \in (V_T)^* \mid S \Rightarrow^+ w\}$$

Linguagens Regulares

- Uma gramática G é regular se G é uma gramática linear.
- Gramática linear: todas as produções são da forma:

$$A \to wB \text{ ou}$$
$$A \to Bw \text{ ou}$$
$$A \to w$$

-

Gramática Linear	Produções
Linear à Esquerda (GLE)	$A \to Bw$ ou $A \to w$
Linear Unitária à Esquerda (GLUE)	$GLE + \|w\| \leq 1$
Linear à Direita (GLD)	$A \to wB$ ou $A \to w$
Linear Unitária à Direita (GLUD)	$GLD + \|w\| \leq 1$

Linguagens Regulares

- Exemplo:
  - A linguagem **a(ba)*** é gerada pelas seguintes gramáticas regulares:
    - GLD:

$$G_1 = (\{a, b\}, \{S, A\}, \mathbb{P}, S)$$
$$\mathbb{P} : S \rightarrow aA$$
$$A \rightarrow baA|\varepsilon$$

    - GLE:

$$G_2 = (\{a, b\}, \{S\}, \mathbb{P}, S)$$
$$\mathbb{P} : S \rightarrow Sba|a$$

    - GLUD:

$$G_3 = (\{a, b\}, \{S, A, B\}, \mathbb{P}, S)$$
$$\mathbb{P} : S \rightarrow aA$$
$$A \rightarrow bB|\varepsilon$$
$$B \rightarrow aA$$

    - GLUE:

$$G_4 = (\{a, b\}, \{S, A\}, \mathbb{P}, S)$$
$$\mathbb{P} : S \rightarrow Aa|a$$
$$A \rightarrow Sb$$