Aula 6: Expressão Regular e Autômatos

Prof. Lucio A. Rocha

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

2º semestre / 2023

Sumário

Seção 1

Expressão Regular e Autômatos

Autômato Finito

- (Revisão) Autômato Finito: É uma máquina de estados finitos que aceita símbolos de entrada de uma sentença.
- Ao final da sentença, o autômato indica se a sentença é válida para a gramática ou não.
- O autômato é definido para o conjunto de símbolos que devem ser reconhecidos.

Expressão Regular e Autômatos

- 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)

Expressão Regular e Autômatos

- Estados e transição:

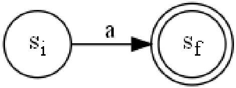


Figura: *AF*.

- Estado inicial e estados finais:

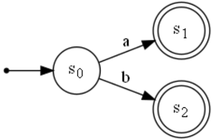


Figura: *AF* com Múltiplos Estados Finais.

Expressão Regular e Autômatos

- Autômato Finito Determinístico (AFD):
  - A partir de um estado:
  - A transição sempre é feita a partir do símbolo de entrada.
  - Não há transição pela sentença vazia.
  - Não há transições alternativas a partir de um dado estado com um determinado símbolo de entrada.

Expressão Regular e Autômatos

- (Revisão) Autômato Finito Não-Determinístico (AFN)
  - A partir de um estado:
  - Pode existir transição para dois ou mais estados diferentes a partir do símbolo de entrada.
  - Uma das transições é escolhida se existir o símbolo na sentença.
  - Pode haver transição para outro(s) estado(s) sem a existência de nenhum símbolo.
    - Transição pela sentença vazia.

Expressão Regular e Autômatos

- Geração de AFD a partir de Expressão Regular (ER):
  - Procedimento sistemático para construir um AFD que reconhece palavras de uma linguagem regular:
  - **Algoritmo de Thompson**: construção de um AFN a partir de uma ER.
  - **Método da construção de subconjuntos**: conversão do AFN para um AFD equivalente.
  - **Minimização de estados**: combinação de estados redundantes para construir o menor AFD que reconhece sentenças da linguagem regular.

Expressão Regular e Autômatos

- Algoritmo de Thompson:
  - Gera um AFN pela combinação de autômatos menores que reconhecem na ER os elementos primitivos:
    - Um símbolo do alfabeto.
    - Concatenação de duas ER.
    - Alternativa de duas ER.
    - Repetição (zero ou mais vezes) de uma ER.

Expressão Regular e Autômatos

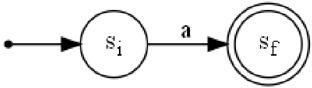
- Algoritmo de Thompson:
  - Autômato que reconhece um símbolo do alfabeto.
  - 

Figura: *AF*.

Expressão Regular e Autômatos

- Algoritmo de Thompson:
  - Para as demais construções, dois autômatos serão combinados: um para a ER A e outro para a ER B.

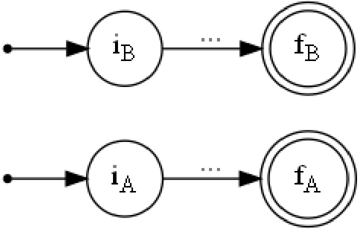


Figura: *AF*.

Expressão Regular e Autômatos

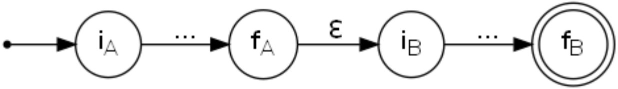
- Algoritmo de Thompson:
  - Concatenação de duas ER: Autômato que reconhece *AB*.
    - 

Figura: $AFN\varepsilon$.

Expressão Regular e Autômatos

- Algoritmo de Thompson:
  - Alternativa de duas ER: Autômato que reconhece $A|B$.

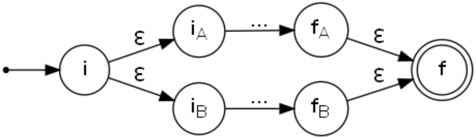
- 

Figura: $AFN\varepsilon$.

Expressão Regular e Autômatos

- Algoritmo de Thompson:
  - Repetição de uma ER: Autômato que reconhece $A*$

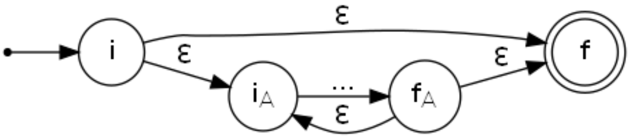


Figura: $AFN\varepsilon$.

Expressão Regular e Autômatos

- Exemplo:
  - AFN para reconhecer palavras da linguagem:

$$(0|1) * 0$$

  - Procedimento:
    1. Concatenação, para reconhecer $(0|1)*$ e 0
    2. Repetição, para reconhecer $(0|1)*$
    3. Alternativa, para reconhecer $0|1$

Expressão Regular e Autômatos

- Concatenação, para reconhecer $(0|1)*$ e 0

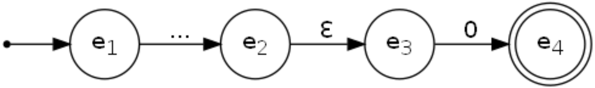
- 

Figura: *AFNε*.

Expressão Regular e Autômatos

- Concatenação, para reconhecer $(0|1)*$ e 0 ✓
- Repetição, para reconhecer $(0|1)*$

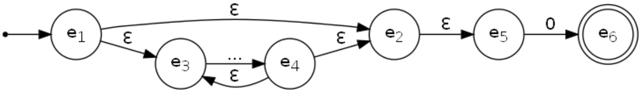


Figura: *AFNε*.

Expressão Regular e Autômatos

- Concatenação, para reconhecer $(0|1)*$ e 0 ✓
- Repetição, para reconhecer $(0|1)*$ ✓
- Alternativa, para reconhecer $0|1$
-

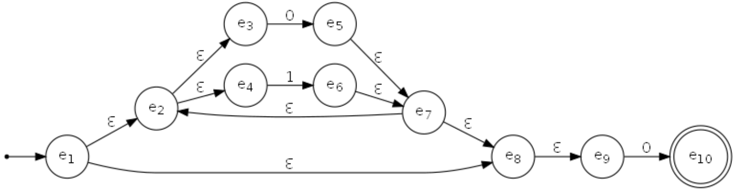


Figura: *AFNε*.

Expressão Regular e Autômatos

- Método da construção de subconjuntos.
  - Procedimento para converter um AFN em um AFD
  - Autômato gerado reconhece sentenças da mesma ER
  - Estados que podem ser alcançados a partir de outro por meio de transições pela string vazia são combinados em um único estado.
  - Conceito de ε^* de um subconjunto de estados do AFN.

Expressão Regular e Autômatos

- Método da construção de subconjuntos
  1. Obter estado inicial.
     - O estado inicial do AFD é a ε^* do conjunto contendo apenas o estado inicial do AFND.
  2. Obter novos estados e transições.
     - Cada estado obtido para o AFD é analisado para descobrir, para cada símbolo do alfabeto, suas transições de saída e novos estados que são gerados.
  3. Marcar estados finais.
     - Cada estado do AFD que contenha em seu subconjunto um estado final do AFN será um estado final do AFD.

Expressão Regular e Autômatos

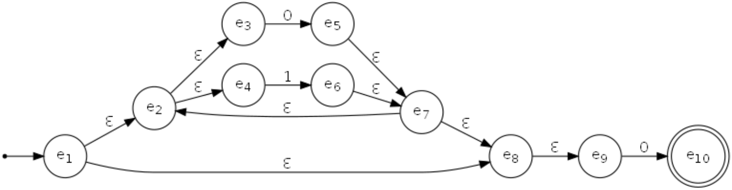
- Exemplo: Conversão do AFN para AFD.
- Expressão regular: **(0|1)*0**
- 

Figura: *AFNε*.

Expressão Regular e Autômatos

- $E^*\{e1\} = S0 = \{e1, e2, e3, e4, e8, e9\}$

S_0	e_1	e_2	e_3	e_4	e_8	e_9
0	–	–	e_5	–	–	e_{10}
1	–	–	–	e_6	–	–

- $E^*\{e5, e10\} = S1 = \{e2, e3, e4, e5, e7, e8, e9, e10\}$

S_1	e_2	e_3	e_4	e_5	e_7	e_8	e_9	e_{10}
0	–	e_5	–	–	–	–	e_{10}	–
1	–	–	e_6	–	–	–	–	–

- $E^*\{e6\} = S2 = \{e2, e3, e4, e6, e7, e8, e9\}$

S_1	e_2	e_3	e_4	e_5	e_7	e_8	e_9	e_{10}	e_{10}
0	–	e_5	–	–	–	–	–	–	e_{10}
1	–	–	e_6	–	–	–	–	–	–

Expressão Regular e Autômatos

- Tabela de Transições para Reconhecer sentenças da ER **(0|1)*0**

	S_0	S_1	S_2
0	S_1	S_1	S_1
1	S_2	S_2	S_2

- Estado inicial: S_0
- Estados finais: S_1

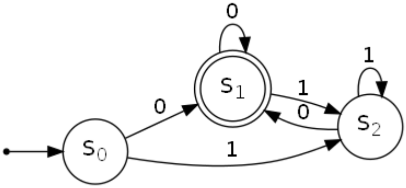
- 

Figura: AFD da ER: $(0|1) * 0$.

Expressão Regular e Autômatos

- Minimização de Estados
  - Eliminar estados redundantes
  - Particionamento do conjunto de estados
    - $P_1 = \{C_1, C_2\} \mid C_1 = F, C_2 = Q - F$
    - F: conjunto de estados finais.
    - Q-F: conjunto de estados não-finais.
    - $C_1 = \{S_1\}$
    - $C_2 = \{S_0, S_2\}$

Expressão Regular e Autômatos

- Minimização de Estados
  - C_1 é unitário, então não possui estados redundantes.
    - $C_1 = \{S_1\}$

	C_1
0	C_1
1	C_2

    - $C_2 = \{S_0, S_2\}$

	S_0	S_2
0	C_1	C_1
1	C_2	C_2

$\rightarrow$

	C_2
0	C_1
1	C_2

Expressão Regular e Autômatos

- Minimização de Estados:

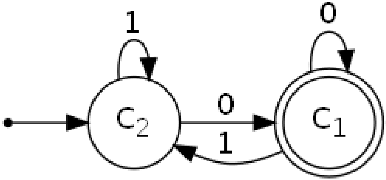


Figura: AFD Minimizado da ER: $(0|1) * 0$