MAC 414

Autômatos, Computabilidade e Complexidade

aula 3 — 21/09/2020

Autômatos

Definição

Um autômato determinístico consiste de 5 componentes,

$$\mathscr{A} = (K, \Sigma, \delta, s, F),$$

onde:

- K *é um conjunto finito, de estados.*
- Σ *é um alfabeto*
- $\delta : K \times \Sigma \to K$ *é a função de transição (ou programa)*
- $s \in K$ *é o estado inicial*
- $F \subseteq K$ *são os estados finais (ou de aceitação).*

Característica fundamental

Memória finita

Característica fundamental

Memória finita

Pré-fixada no algoritmo, independente do tamanho dos dados.

Característica fundamental

Memória finita

Pré-fixada no algoritmo, independente do tamanho dos dados.

Nada de `malloc` ou uso da pilha. Sem recursão ou aninhamento de funções.

Computação em $\mathscr{A}$

Computação em $\mathscr{A}$

Intuição:

Computação em $\mathscr{A}$

Intuição:

$\mathscr{A}$ a cada instante está em um dos estados
(começando em s)

Computação em $\mathscr{A}$

Intuição:

$\mathscr{A}$ a cada instante está em um dos estados (começando em s)

A cada letra que lê, muda de estado, conforme δ

Computação em $\mathscr{A}$

Intuição:

$\mathscr{A}$ a cada instante está em um dos estados (começando em s)

A cada letra que lê, muda de estado, conforme δ

```
while ((c=getchar()) != EOF)
    estado = delta(estado, c);
```

Insira coisas aqui

Computação em $\mathscr{A}$

Intuição:

$\mathscr{A}$ a cada instante está em um dos estados (começando em s)

A cada letra que lê, muda de estado, conforme δ

```
while ((c=getchar()) != EOF)
    estado = delta(estado, c);
```

F vai controlar a saída.

Computação em $\mathscr{A}$

Intuição:

$\mathscr{A}$ a cada instante está em um dos estados (começando em s)

A cada letra que lê, muda de estado, conforme δ

```
while ((c=getchar()) != EOF)
    estado = delta(estado, c);
```

F vai controlar a saída.

(K, Σ, δ, s) é um semiautômato; existem outras especificações de saída.

Computação em $\mathscr{A}$

δ tem uma extensão natural

$$\delta : K \times \Sigma^*$$

definida recursivamente, por:

$$\begin{cases} \delta(q, \lambda) = q \\ \delta(q, x\sigma) = \delta(\delta(q, x), \sigma) \quad q \in K, x \in \Sigma^*, \sigma \in \Sigma. \end{cases}$$

Computação em $\mathscr{A}$

δ tem uma extensão natural

$$\delta : K \times \Sigma^*$$

definida recursivamente, por:

$$\begin{cases} \delta(q, \lambda) = q \\ \delta(q, x\sigma) = \delta(\delta(q, x), \sigma) \quad q \in K, x \in \Sigma^*, \sigma \in \Sigma. \end{cases}$$

Proposição: Para $q \in K, x, y \in \Sigma^*$,

$$\delta(q, xy) = \delta(\delta(q, x), y)$$

Notação melhor

Notação melhor

Escrever simplesmente qx em vez de $\delta(q, x)$

Notação melhor

Escrever simplesmente qx em vez de $\delta(q, x)$

Assim, a proposição fica:

$$q(xy) = (qx)y.$$

A linguagem de um autômato

A linguagem de um autômato

A linguagem reconhecida (ou *aceita*) por um autômato $\mathscr{A}$ é

$$
\begin{aligned}
L(\mathscr{A}) &= \{x \in \Sigma^* \mid \delta(q, x) \in F\} \\
&= \{x \in \Sigma^* \mid s\,x \in F\}
\end{aligned}
$$

A linguagem de um autômato

A linguagem reconhecida (ou *aceita*) por um autômato $\mathscr{A}$ é

$$\begin{aligned} L(\mathscr{A}) &= \{x \in \Sigma^* \mid \delta(q,x) \in F\} \\ &= \{x \in \Sigma^* \mid sx \in F\} \end{aligned}$$

Uma linguagem A é reconhecível se existe um AD $\mathscr{A}$ tal que $L(\mathscr{A}) = A$

A linguagem de um autômato

A linguagem reconhecida (ou *aceita*) por um autômato $\mathscr{A}$ é

$$
\begin{aligned}
L(\mathscr{A}) &= \{x \in \Sigma^* \mid \delta(q, x) \in F\} \\
&= \{x \in \Sigma^* \mid s\, x \in F\}
\end{aligned}
$$

Uma linguagem A é reconhecível se existe um AD $\mathscr{A}$ tal que $L(\mathscr{A}) = A$

Este nome é temporário. É parte do Teorema de Kleene o fato *reconhecível = regular*.

Exemplos triviais

Exemplos triviais

$\mathscr{A} = (\{\bullet\}, \Sigma, \delta, \bullet, \emptyset)$

Exemplos triviais

$\mathscr{A} = (\{\bullet\}, \Sigma, \delta, \bullet, \emptyset)$

$L(\mathscr{A}) = \emptyset$

Exemplos triviais

$$\mathscr{A} = (\{\bullet\}, \Sigma, \delta, \bullet, \emptyset)$$

$$L(\mathscr{A}) = \emptyset$$

$$\mathscr{A} = (\{\bullet\}, \Sigma, \delta, \bullet, \{\bullet\})$$

Exemplos triviais

$$\mathscr{A} = (\{\bullet\}, \Sigma, \delta, \bullet, \emptyset)$$

$$L(\mathscr{A}) = \emptyset$$

$$\mathscr{A} = (\{\bullet\}, \Sigma, \delta, \bullet, \{\bullet\})$$

$$L(\mathscr{A}) = \Sigma^*$$

Outro exemplo

$K = \{1, 2\}$

$\Sigma = \{a, b, c\}$

$\delta: \quad 1a = 1, 2a = 2, \quad 1b = 2, 2b = 1,$

$\quad 1c = 1, 2c = 1$

$s = 1$

$F = \{2\}$

δ	a	b	c
1	1	2	1
2	2	1	1

Outro exemplo

$K = \{1, 2\}$

$\Sigma = \{a, b, c\}$

$\delta : \quad 1a = 1, 2a = 2, \quad 1b = 2, 2b = 1,$

$\quad 1c = 1, 2c = 1$

$s = 1$

$F = \{2\}$

aabaccaba

1 1 1 2 2 1 1 1 2 2

Yes!

O grafo de um autômato

O grafo de um autômato

$G_{\mathscr{A}}$ é um grafo dirigido, com rótulos nas arestas.

O grafo de um autômato

$G_{\mathscr{A}}$ é um grafo dirigido, com rótulos nas arestas.

$V(G_{\mathscr{A}}) = K$

O grafo de um autômato

$G_{\mathscr{A}}$ é um grafo dirigido, com rótulos nas arestas.

$V(G_{\mathscr{A}}) = K$

Arestas: para cada $q \in K, \sigma \in \Sigma$, existe uma aresta α de q a $\delta(q,\sigma)$, cujo rótulo $\rho(\alpha)$ é σ.

O grafo de um autômato

$G_{\mathscr{A}}$ é um grafo dirigido, com rótulos nas arestas.

$V(G_{\mathscr{A}}) = K$

Arestas: para cada $q \in K, \sigma \in \Sigma$, existe uma aresta α de q a $\delta(q,\sigma)$, cujo rótulo $\rho(\alpha)$ é σ.

Desenho:

O grafo de um autômato

$G_{\mathscr{A}}$ é um grafo dirigido, com rótulos nas arestas.

$V(G_{\mathscr{A}}) = K$

Arestas: para cada $q \in K, \sigma \in \Sigma$, existe uma aresta α de q a $\delta(q, \sigma)$, cujo rótulo $\rho(\alpha)$ é σ.

Desenho:

- Cada estado é um círculo, com nome dentro
- Uma flecha aponta s
- Estados de F são círculos duplos
- Se várias arestas têm mesmo início e fim, desenha só um arco, com vários rótulos.

Exemplo

$K = \{1, 2\}$

$\Sigma = \{a, b\}$

$\delta : \quad 1a = 1, 2a = 2, \quad 1b = 2, 2b = 1,$

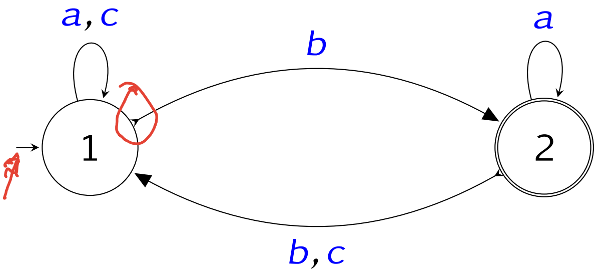
$\quad 1c = 1, 2c = 1$

$s = 1$

$F = \{2\}$

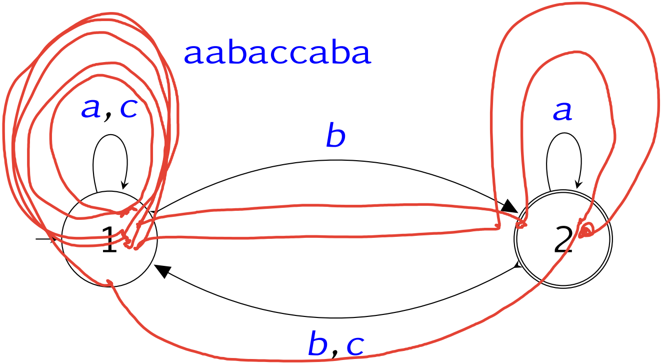
Exemplo

$K = \{1, 2\}$
$\Sigma = \{a, b\}$
$\delta: \quad 1a = 1, 2a = 2, \quad 1b = 2, 2b = 1,$
$\quad 1c = 1, 2c = 1$
$s = 1$
$F = \{2\}$



Exemplo

$K = \{1, 2\}$
$\Sigma = \{a, b\}$
$\delta : \quad 1a = 1, 2a = 2, \quad 1b = 2, 2b = 1,$
$\quad 1c = 1, 2c = 1$
$s = 1$
$F = \{2\}$



Reconhecimento via grafo

Reconhecimento via grafo

caminho

Se P é um passeio em $G_{\mathscr{A}}$, seu rótulo é $\rho(P)$, o produto dos rótulos de suas arestas.

Reconhecimento via grafo

Se P é um passeio em $G_{\mathscr{A}}$, seu rótulo é $\rho(P)$, o produto dos rótulos de suas arestas.

Prop: Se $P = P_1 P_2$ são passeios em em $G_{\mathscr{A}}$,

$$\rho(P) = \rho(P_1)\rho(P_2).$$

Reconhecimento via grafo

Se P é um passeio em $G_{\mathscr{A}}$, seu rótulo é $\rho(P)$, o produto dos rótulos de suas arestas.

Prop: Se $P = P_1 P_2$ são passeios em em $G_{\mathscr{A}}$,

$$\rho(P) = \rho(P_1)\rho(P_2).$$

Prop: Para cada $p \in K$ e $x \in \Sigma^*$, existe um único passeio P em $G_{\mathscr{A}}$ com rótulo x, e ele termina em $\delta(p, x)$.

Reconhecimento via grafo

Se P é um passeio em $G_{\mathscr{A}}$, seu rótulo é $\rho(P)$, o produto dos rótulos de suas arestas.

Prop: Se $P = P_1 P_2$ são passeios em em $G_{\mathscr{A}}$,

$$\rho(P) = \rho(P_1)\rho(P_2).$$

Prop: Para cada $p \in K$ e $x \in \Sigma^*$, existe um único passeio P em $G_{\mathscr{A}}$ com rótulo x, e ele termina em $\delta(p, x)$.

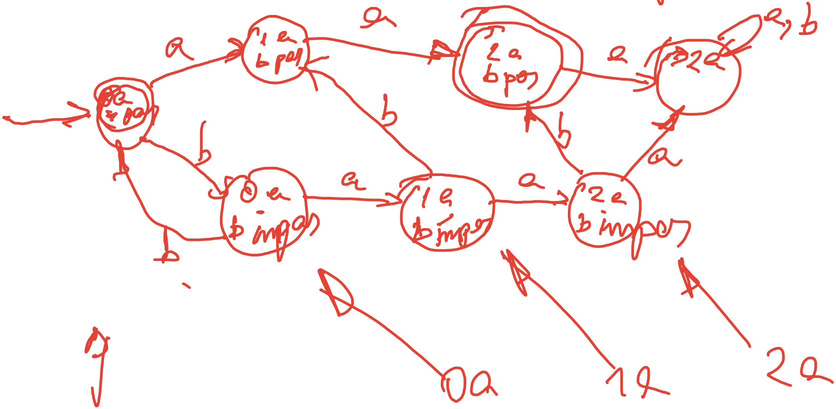
px

Prop: $L(\mathscr{A})$ é o conjunto de todos os rótulos de passeios em $G_{\mathscr{A}}$ de s a F.

$sx \in F$

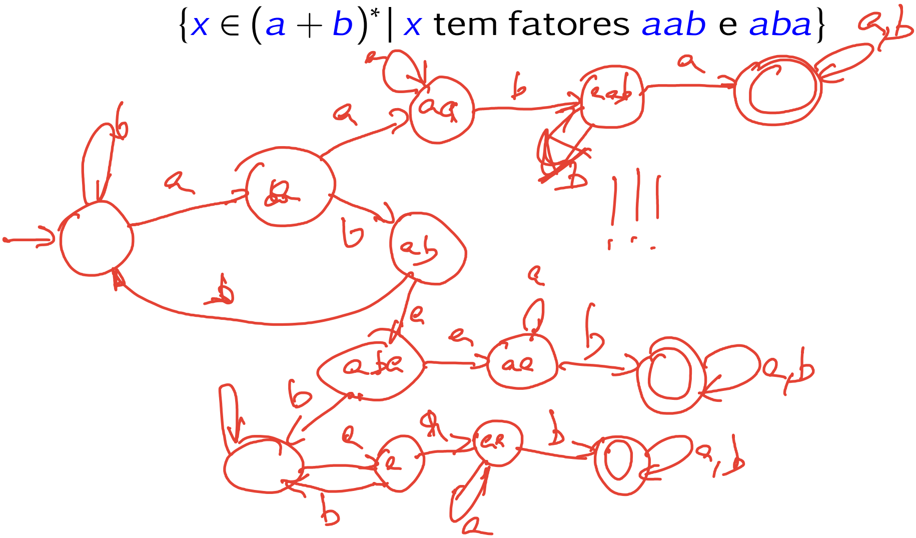
Exemplo

$\{x \in (a+b)^* \mid |x|_a = 0 \text{ ou } 2, |x|_b \text{ é par}\}$



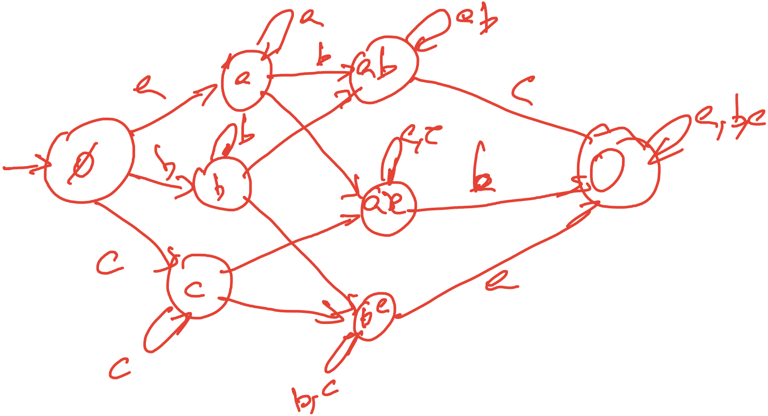
Exemplo

$\{x \in (a+b)^* \mid x$ tem fatores aab e $aba\}$



Exemplo

$\{x \in (a + b + c)^* \mid x \text{ contém todas as letras}\}$



Operações com linguagens

Operações com linguagens

Complemento

Operações com linguagens

Complemento

Prop: O complemento (em relação a Σ^*) de uma linguagem reconhecível é reconhecível.

Operações com linguagens

Complemento

Prop: O complemento (em relação a Σ^*) de uma linguagem reconhecível é reconhecível.

Dem: Seja $L \subseteq \Sigma^*$ uma linguagem reconhecível e $\mathscr{A} = (K, \Sigma, \delta, s, F)$ um AD reconhecendo L. Então $\hat{\mathscr{A}} = (K, \Sigma, \delta, s, K \setminus F)$ reconhece $\bar{L}$. Com efeito,

$$x \in L(\hat{\mathscr{A}}) \iff sx \in K \setminus F$$
$$\iff sx \notin F$$
$$\iff x \notin L(\mathscr{A}) \iff x \in \Sigma^* \setminus L(\mathscr{A})$$

Operações com linguagens

Complemento

Prop: O complemento (em relação a Σ^*) de uma linguagem reconhecível é reconhecível.

Dem: Seja $L \subseteq \Sigma^*$ uma linguagem reconhecível e $\mathscr{A} = (K, \Sigma, \delta, s, F)$ um AD reconhecendo L. Então $\hat{\mathscr{A}} = (K, \Sigma, \delta, s, K \setminus F)$ reconhece $\bar{L}$. Com efeito,

União e interseção

União e interseção

Sejam $L_1 = L(\mathscr{A}_1), L_2 = L(\mathscr{A}_2)$, onde $\mathscr{A}_1 = (K_1, \Sigma, \delta_1, s_1, F_1), \mathscr{A}_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$. Vamos mostrar que $L_1 \cup L_2$ e $l_1 \cap L_2$ são reconhecíveis.

União e interseção

Sejam $L_1 = L(\mathscr{A}_1), L_2 = L(\mathscr{A}_2)$, onde $\mathscr{A}_1 = (K_1, \Sigma, \delta_1, s_1, F_1), \mathscr{A}_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$. Vamos mostrar que $L_1 \cup L_2$ e $l_1 \cap L_2$ são reconhecíveis.

Idéia: processamento paralelo.

União e interseção

Sejam $L_1 = L(\mathscr{A}_1), L_2 = L(\mathscr{A}_2)$, onde $\mathscr{A}_1 = (K_1, \Sigma, \delta_1, s_1, F_1), \mathscr{A}_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$. Vamos mostrar que $L_1 \cup L_2$ e $l_1 \cap L_2$ são reconhecíveis.

Idéia: processamento paralelo.

Implementação: sejam $\mathscr{S}_i = (K_i, \Sigma, \delta_i), i = 1, 2$. Construa

$$\mathscr{S} = (K_1 \times K_2, \Sigma, \delta, (s_1, s_2)),\ (q_1, q_2)\sigma = (q_1\sigma, q_2\sigma).$$

União e interseção

Sejam $L_1 = L(\mathscr{A}_1), L_2 = L(\mathscr{A}_2)$, onde $\mathscr{A}_1 = (K_1, \Sigma, \delta_1, s_1, F_1), \mathscr{A}_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$. Vamos mostrar que $L_1 \cup L_2$ e $l_1 \cap L_2$ são reconhecíveis.

Idéia: processamento paralelo.

Implementação: sejam $\mathscr{S}_i = (K_i, \Sigma, \delta_i)$, $i = 1, 2$. Construa

$$\mathscr{S} = (K_1 \times K_2, \Sigma, \delta, (s_1, s_2)),\ (q_1, q_2)\sigma = (q_1\sigma, q_2\sigma).$$

Por indução, $(q_1, q_2)x = (q_1x, q_2x)$ para toda palavra x.

União e interseção

Sejam $L_1 = L(\mathscr{A}_1), L_2 = L(\mathscr{A}_2)$, onde
$\mathscr{A}_1 = (K_1, \Sigma, \delta_1, s_1, F_1), \mathscr{A}_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$. Vamos mostrar que $L_1 \cup L_2$ e $l_1 \cap L_2$ são reconhecíveis.

Idéia: processamento paralelo.

Implementação: sejam $\mathscr{S}_i = (K_i, \Sigma, \delta_i), i = 1, 2$.
Construa

$$\mathscr{S} = (K_1 \times K_2, \Sigma, \delta, (s_1, s_2)),\ (q_1, q_2)\sigma = (q_1\sigma, q_2\sigma).$$

Por indução, $(q_1, q_2)x = (q_1x, q_2x)$ para toda palavra x.
Considere $(\mathscr{S}, F)$. Como definir F

União e interseção

Sejam $L_1 = L(\mathscr{A}_1), L_2 = L(\mathscr{A}_2)$, onde $\mathscr{A}_1 = (K_1, \Sigma, \delta_1, s_1, F_1), \mathscr{A}_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$. Vamos mostrar que $L_1 \cup L_2$ e $l_1 \cap L_2$ são reconhecíveis.

Idéia: processamento paralelo.

Implementação: sejam $\mathscr{S}_i = (K_i, \Sigma, \delta_i), i = 1, 2$. Construa

$$\mathscr{S} = (K_1 \times K_2, \Sigma, \delta, (s_1, s_2)), \; (q_1, q_2)\sigma = (q_1\sigma, q_2\sigma).$$

Por indução, $(q_1, q_2)x = (q_1x, q_2x)$ para toda palavra x.

Considere $(\mathscr{S}, F)$. Como definir F

para $\cap$: $F_1 \times F_2$

União e interseção

Sejam $L_1 = L(\mathscr{A}_1), L_2 = L(\mathscr{A}_2)$, onde $\mathscr{A}_1 = (K_1, \Sigma, \delta_1, s_1, F_1), \mathscr{A}_2 = (K_2, \Sigma, \delta_2, s_2, F_2)$. Vamos mostrar que $L_1 \cup L_2$ e $l_1 \cap L_2$ são reconhecíveis.

Idéia: processamento paralelo.

Implementação: sejam $\mathscr{S}_i = (K_i, \Sigma, \delta_i), i = 1, 2$. Construa

$$\mathscr{S} = (K_1 \times K_2, \Sigma, \delta, (s_1, s_2)),\ (q_1, q_2)\sigma = (q_1\sigma, q_2\sigma).$$

Por indução, $(q_1, q_2)x = (q_1 x, q_2 x)$ para toda palavra x.

Considere $(\mathscr{S}, F)$. Como definir F

para $\cap$: $F_1 \times F_2$

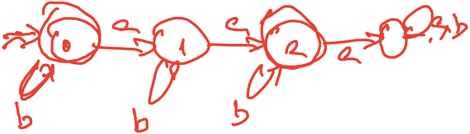
para $\cup$: $F_1 \times K_2 \cup K_1 \times F_2$

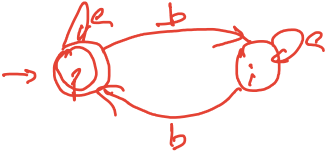
Exemplo

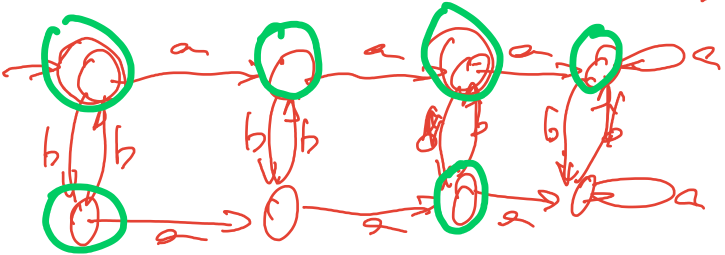
Exemplo

$\Sigma = \{a, b\}$

$L_1 = \{x \mid |x|_a = 0 \text{ ou } 2\}$ $\cap$ $L_2 = \{x \mid |x|_b \text{ é par}\}$







$\cup$

Produto

~~f_1~~ bool f_1(string x) bool f_2(string x)

bool f(string x) devolve 1 sse $x \in L_1 \cdot L_2$

$L_i = \{ x \in \Sigma^* \mid f_i(x) = 1 \}$

Tente escrever $x = yz$ de todo jeito
se alguma der certo, aceite
senão, rejeite

Não-determinismo