1) CONCEITOS ESSENCIAIS

Alfabetos e cadeias

→ Um alfabeto é um conjunto finito de elementos chamados símbolos. Geralmente representados por letras gregas maiúsculas.
Exemplos: $\Sigma = \{0, 1\}$, $\Sigma_1 = \{a, b, \ldots, z\}$, $\Sigma_2 = \{a, b\}$, $\Gamma = \{\#, a1, bd\}$.

→ Dado um alfabeto Σ, uma cadeia (sobre Σ) é uma sequência $x_1 x_2 \ldots x_n$, onde $x_i \in \Sigma$ para cada $1 \leq i \leq n$, ou seja, é uma sequência finita de símbolos.
Geralmente representados por letras gregas minúsculas.
Outros termos: strings ou palavras.

→ O comprimento de uma cadeia w é a quantidade de posições na sequência e é denotado $|w|$.
Exemplos: $w = 011011$ é cadeia sobre $\Sigma = \{0, 1\}$ e $|w| = 6$
$\alpha = carla$ é cadeia sobre $\Sigma_1 = \{a, b, \ldots, z\}$ e $|\alpha| = 5$
$\beta = bd\#a1bd$ é cadeia sobre $\Gamma = \{\#, a1, bd\}$ e $|\beta| = 4$

→ Se $x \in \Sigma$ e w é cadeia sobre Σ, denotamos por $|w|_x$ a quantidade de vezes que o símbolo x aparece em w.
Exemplos: $|\alpha|_a = 2$, $|w|_1 = 4$

→ A concatenação de uma cadeia $\alpha = \alpha_1 \alpha_2 \cdots \alpha_n$ com uma cadeia $\beta = \beta_1 \beta_2 \cdots \beta_m$ é a cadeia $\alpha\beta = \alpha_1 \alpha_2 \cdots \alpha_n \beta_1 \beta_2 \cdots \beta_m$.

→ Denotamos por α^k a concatenação de α com ela mesma k vezes.

→ Exemplos: $\alpha = aba$ e $\beta = caa$
$\alpha\beta = abacaa$ $\quad \beta\alpha = caaaba$ $\quad \alpha^3 = aba\,aba\,aba$
$\alpha^2\beta^3 = aba\,aba\,caa\,caa\,caa$

Cadeias importantes

→ O reverso da cadeia $w = w_1 w_2 \dots w_n$ é a cadeia
$$w^R = w_n w_{n-1} \dots w_1.$$

→ A cadeia vazia, denotada por ε, é a cadeia de comprimento 0.
Obs: para qualquer cadeia w sobre Σ, $w\varepsilon = \varepsilon w = w$.

Subcadeias

→ Uma cadeia β é subcadeia de outra cadeia w se existem cadeias α e γ tais que $w = \alpha\beta\gamma$.
Exemplos: $\beta = 01$ é subcadeia de $w = 011011$ pois $w = \alpha\beta\gamma$ com $\alpha = \varepsilon$ e $\gamma = 1011$ (ou $\alpha = 011$ e $\gamma = 1$).

Potência de alfabeto

→ Dado um alfabeto Σ, denotamos por Σ^k o conjunto de todas as cadeias de comprimento k sobre Σ.

→ Se $\Sigma = \{0,1\}$, então $\Sigma^0 = \{\varepsilon\}$, $\Sigma^1 = \{0,1\}$, $\Sigma^2 = \{00, 01, 10, 11\}$, $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

→ Obs: $|\Sigma^k| = |\Sigma|^k$

Fecho de Kleene e fecho positivo de alfabetos

→ O fecho de Kleene de Σ, denotado Σ^*, é o conjunto de todas as cadeias sobre Σ, isto é,

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \cdots = \bigcup_{i=0}^{\infty} \Sigma^i$$

→ O fecho positivo de Σ, denotado Σ^+, é o conjunto de todas as cadeias não vazias sobre Σ, isto é,

$$\Sigma^+ = \bigcup_{i=1}^{\infty} \Sigma^i$$

$\{\varepsilon\} \neq \{\} = \emptyset$

→ Logo, $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$

→ Seja $\Sigma = \{0,1,2\}$. Então

$$\Sigma^* = \{\varepsilon, 0, 1, 2, 00, 01, 02, 10, 11, 12, 20, 21, 22, 000, 001, \ldots\}$$
$$\Sigma^+ = \{0, 1, 2, 00, 01, 02, 10, 11, 12, 20, 21, 22, 000, 001, \ldots\}$$

OBSERVAÇÃO: agora, ao invés de escrever "w é uma cadeia sobre o alfabeto Σ", basta escrever "$w \in \Sigma^*$".

Exemplo: $\Sigma^k = \{w \in \Sigma^* : |w| = k\}$.

Linguagens

→ Uma linguagem sobre um alfabeto Σ é um subconjunto de Σ^*. geralmente representadas por letras maiúsculas.
Assim, "L é linguagem sobre Σ" equivale a "$L \subseteq \Sigma^*$".

→ Exemplos:
$L = \{010, 11, 0, 1011\}$ é linguagem sobre $\Sigma = \{0, 1\}$.
$A = \{\text{abacate}, \text{uva}, \text{cafe}, \text{banana}, \varepsilon\}$ é linguagem sobre $\Gamma = \{a, \dots, z\}$
$B = \{ w \in \{x, y\}^* : |w| \text{ é par} \}$
$= \{xy, xx, xxxx, yxxy, \varepsilon \dots\}$

$L_1 = \{ w \in \{0,1\}^* : |w|_0 = |w|_1 \}$
$L_2 = \{ w \in \{a,b\}^* : |w|_a \text{ é par} \}$
$L_3 = \{ 0^n 1^n : n \geqslant 1 \}$
$L_4 = \{ a^i b^j : 0 \leqslant i \leqslant j \}$
$L_5 = \{\varepsilon\}$
$L_6 = \{ \}$
$L_7 = \{ w \in \{0,1,2\}^* : \text{o } 5^{\text{o}} \text{ símbolo de } w \text{ é } 2 \}$
$L_8 = \{ w : w \text{ é um programa sintaticamente correto em C} \}$

Linguagens × Problemas

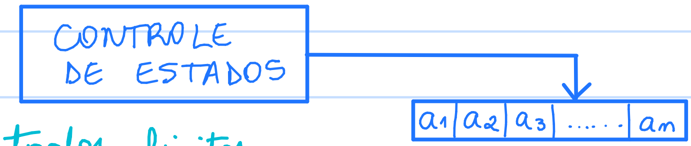
→ Vamos focar muito na questão
"Dados $L \subseteq \Sigma^*$ e $w \in \Sigma^*$, w pertence a L?"

→ Acontece que qualquer problema pode ser expresso com linguagens.
$P = \{ w \in \{0, 1, \dots, 9\}^* : \text{se vista como número, } w \text{ é primo} \}$
$P_1 = \{ w0 : w \in \{0,1\}^* \}$
$S = \{ a^i b^j c^k : i, j, k \in \mathbb{Z}_{\geqslant 1} \text{ e } i + j = k \}$

Autômatos Finitos

→ São modelos computacionais com quantidade limitada (finita) de memória.

→ São utilizados em controladores simples e em reconhecimento de padrões em dados.

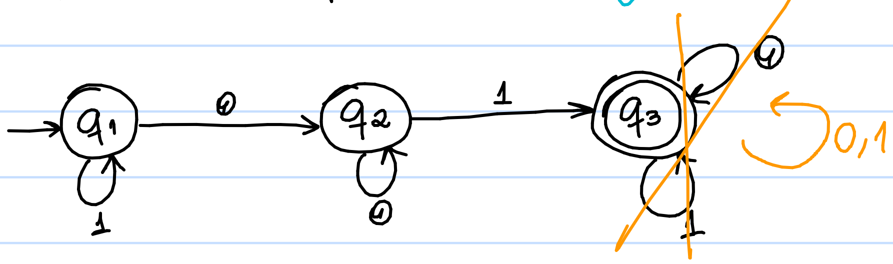


→ Também chamados máquinas de estados finitos.
O propósito de um estado é lembrar coisas importantes.
Por ser em quantidade finita, precisa ser bem projetado.
Pode ser implementado em sistemas com poucos recursos.

→ São dispositivos reconhecedores de linguagens: dada $w \in \Sigma^*$, aceita w se $w \in L$ e rejeita caso contrário, sendo L a linguagem que ele foi projetado para reconhecer.

→ Possuem estados e transições entre os estados.
A ideia é processar w um símbolo por vez, começando no estado inicial, seguindo as transições, e eventualmente aceitar ($w \in L$) ou rejeitar ($w \notin L$).

→ Em geral são representados por um diagrama de estados



Autômatos Finitos Determinísticos

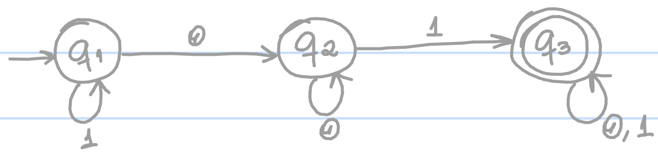
DEFINIÇÃO: Um autômato finito determinístico (AFD) é uma 5-upla $(Q, \Sigma, \delta, q_0, F)$ em que

- Q é um conjunto finito de estados
- Σ é um alfabeto
- $\delta : Q \times \Sigma \to Q$ é uma função de transição
- $q_0 \in Q$ é o estado inicial
- $F \subseteq Q$ é o conjunto de estados finais
  ↳ $\emptyset \subseteq Q$

Exemplo

→ $M = (\{q_1, q_2, q_3\}, \{0, 1\}, \delta, q_1, \{q_3\})$ em que δ é dada por

δ	0	1
q_1	q_2	q_1
q_2	q_2	q_3
q_3	q_3	q_3



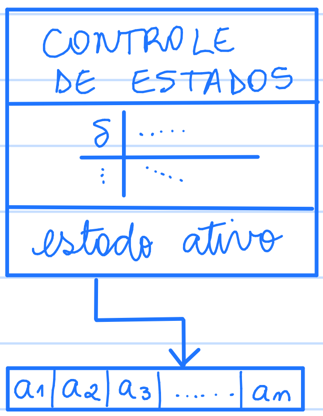
é um AFD.

Dissecando um AFD

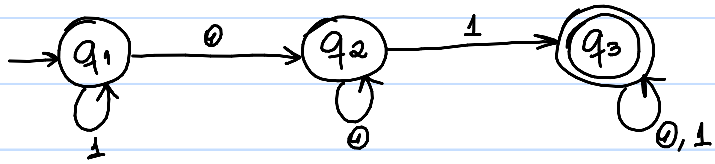
→ Autômato ⇒ máquina automática

→ Finito ⇒ com memória limitada (estados)

→ Determinístico ⇒ para cada símbolo do alfabeto existe exatamente um estado para o qual o autômato pode transitar a partir do único estado ativo.



Computação - informal



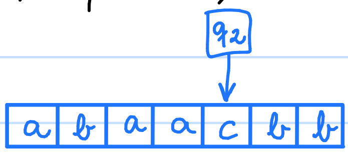
Vamos computar as cadeias:

- 00110
- 010101
- 100

$q_1 00110 \vdash 0 q_2 0110 \vdash 00 q_2 110 \vdash 001 q_3 10$
$\vdash 0011 q_3 0 \vdash 00110 q_3$

Configuração e movimento

DEFINIÇÃO: Uma configuração de um AFD $M = (Q, \Sigma, \delta, q_0, F)$ é descrita por $\alpha q \beta$ onde $\alpha, \beta \in \Sigma^*$, $\alpha\beta$ é a cadeia da entrada, $q \in Q$, e a cabeça está sobre o primeiro símbolo de β.



$\equiv$ $a\, b\, a\, a\, q_2\, c\, b\, b$

DEFINIÇÃO: Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD e sejam $x \in \Sigma$, $\alpha, \beta \in \Sigma^*$ e $q_i, q_j \in Q$. Se $\delta(q_i, x) = q_j$, então $\alpha q_i x \beta \vdash \alpha x q_j \beta$ é um movimento de M.

Usamos $\vdash^*$ para representar uma sequência de zero ou mais movimentos em M.

Computação em AFD

DEFINIÇÃO: Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD e $w \in \Sigma^*$. Dizemos que M aceita w se $q_0 w \vdash^* w q_F$ com $q_F \in F$.

Se $q_F \notin F$, então M rejeita w.

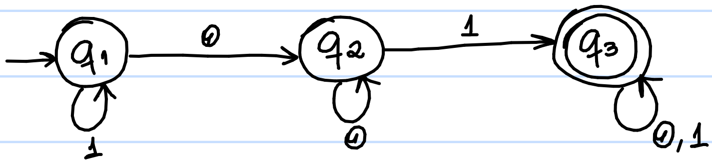
Função de transição estendida

DEFINIÇÃO: Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD. A *função de transição estendida* de M é a função $\hat{\delta} : Q \times \Sigma^* \to Q$ definida da seguinte forma: para $q \in Q$ e $w \in \Sigma^*$,

$$\hat{\delta}(q, w) = \begin{cases} q & \text{se } w = \varepsilon \\ \delta(\hat{\delta}(q, \alpha), x) & \text{se } w = \alpha x, \text{ com } x \in \Sigma \text{ e } \alpha \in \Sigma^* \end{cases}$$

→ Ou seja: $\hat{\delta}(q, w)$ é o estado ativo em M após computar toda uma cadeia w a partir do estado q.

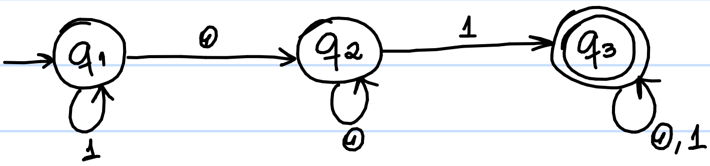
Exemplos



- $\hat{\delta}(q_1, 1100) = q_2$
- $\hat{\delta}(q_1, 001010) = q_3$
- $\hat{\delta}(q_1, \varepsilon) = q_1$
- $\hat{\delta}(q_2, 0000) = q_2$
- $\hat{\delta}(q_2, 1100) = q_3$

Computação em AFD - Definição alternativa

DEFINIÇÃO: Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD e seja $w \in \Sigma^*$. Dizemos que *M aceita w* se $\hat{\delta}(q_0, w) \in F$. Caso contrário, *$M$ rejeita w*.



Computação em AFD - Definição alternativa 2

DEFINIÇÃO: Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD e seja $w = w_1 w_2 \cdots w_n$ uma cadeia sobre Σ ($w \in \Sigma^*$).

Dizemos que M aceita w se existe uma sequência de estados $(r_0, r_1, \dots, r_n)$ tal que

- $r_0 = q_0$
- $r_{i+1} = \delta(r_i, w_{i+1}) \quad \forall\, i = 0, \dots, n-1$
- $r_n \in F$

A linguagem de um AFD

DEFINIÇÃO: Seja $M = (Q, \Sigma, \delta, q_0, F)$ um AFD. Dizemos que

$$X = \{ w \in \Sigma^* : M \text{ aceita } w \}$$

é a linguagem reconhecida por M, ou simplesmente a linguagem de M.

Também dizemos que M reconhece X.

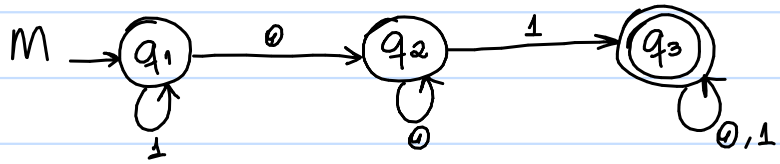
Denotamos tal linguagem por $L(M)$.

Importante

Um AFD aceita cadeias (0 ou mais).

Ele reconhece uma única linguagem.

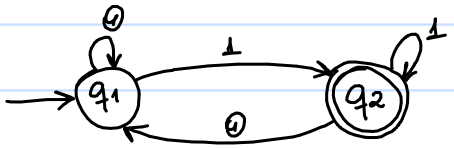
Exemplo



$$L(M) = \{ w \in \{0,1\}^* : 01 \text{ é subcadeia de } w \}$$

Mais exemplos

M_2



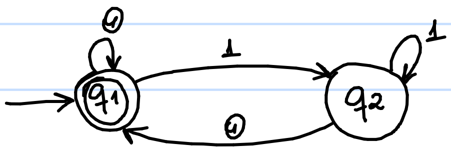
$L(M_2) = \{ w \in \{0,1\}^* : \quad w \text{ termina em } 1 \}$

pois : $\hat{\delta}(q_1, w) = q_1 \iff w$ termina em 0 $(w = \alpha 0$ para $\alpha \in \{0,1\}^*)$
ou $w = \varepsilon$

$\hat{\delta}(q_1, w) = q_2 \iff w$ termina em 1 $(w = \alpha 1$ para $\alpha \in \{0,1\}^*)$

Mais exemplos

M_2



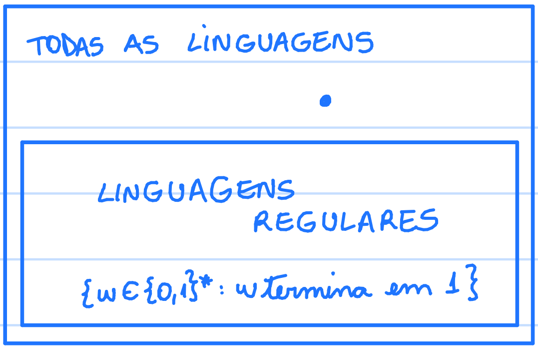
$L(M_2) = \{ w \in \{0,1\}^* : w = \varepsilon \text{ ou } w \text{ termina em } 0 \}$

pois : $\hat{\delta}(q_1, w) = q_1 \iff w$ termina em 0 $(w = \alpha 0$ para $\alpha \in \{0,1\}^*)$
ou $w = \varepsilon$

$\hat{\delta}(q_1, w) = q_2 \iff w$ termina em 1 $(w = \alpha 1$ para $\alpha \in \{0,1\}^*)$

Linguagens regulares

DEFINIÇÃO: Uma linguagem é *regular* se algum AFD a reconhece.



Problemas de interesse

1) Dado um AFD M, determine $L(M)$.

2) Dada $L \subseteq \Sigma^*$, faça um AFD que a reconheça.

3) Dada $L \subseteq \Sigma^*$, determine se L é regular.

Observe:

- Pode existir mais de um AFD que reconhece uma ling.
- não encontrar um AFD não implica na sua inexistência.

Mais exemplos

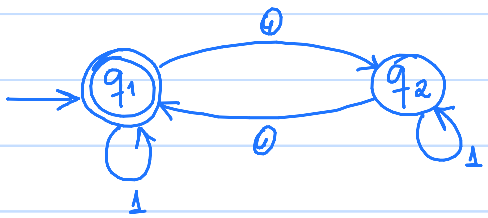
Dê um AFD para $L_1 = \{ w \in \{0,1\}^* : |w|_0 \text{ é par} \}$

→ O que é importante lembrar?
Quais eventos são importantes?
Quais estados criar?

→ Se a string x já foi lida e você sabe algo sobre ela, qual decisão tomar ao ler um 0? E um 1?

→ A todo momento, a cadeia pode acabar...

→ Qual a menor cadeia que ele deve aceitar?



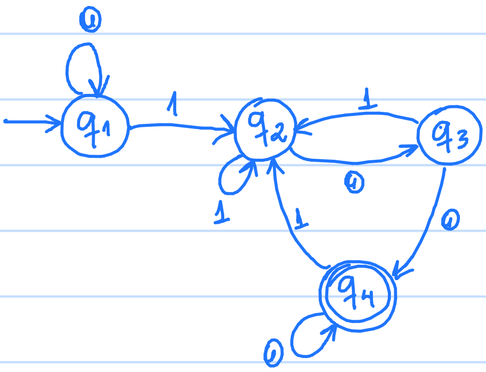
$$\hat{\delta}(q_1, w) = q_1 \iff |w|_0 \text{ é par}$$
$$\hat{\delta}(q_1, w) = q_2 \iff |w|_0 \text{ é ímpar}$$

Mais exemplos

Dê um AFD para $L_2 = \{w \in \{0,1\}^* : w$ termina em 00 e contém pelo menos um $1\}$

→ Eventos importantes:

- ler 0 (começo de um padrão importante)
- ler 0 depois de ter lido um 0 (pode ser o final)
- já ter lido um 1 em algum momento



$\hat{\delta}(q_1,w) = q_1 \iff w = 0^k$ com $k \geq 0$

$\hat{\delta}(q_1,w) = q_2 \iff w = \alpha 1$, com $\alpha \in \Sigma^*$

$\hat{\delta}(q_1,w) = q_3 \iff w = \alpha 10$, com $\alpha \in \Sigma^*$

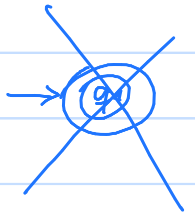
$\hat{\delta}(q_1,w) = q_4 \iff w = \alpha 1000^k$, com $\alpha \in \Sigma^*$ e $k \geq 0$

Erros comuns: $q_3 \iff w = \alpha 0$

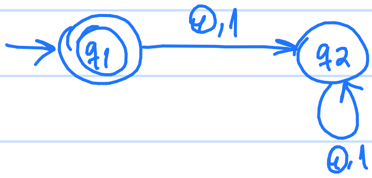
$q_2 \iff w = 0^k 1$

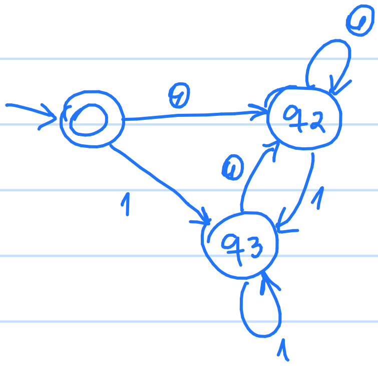
Mais exemplos

Dê um AFD para $L_3 = \{\varepsilon\}$, $\Sigma = \{0,1\}$



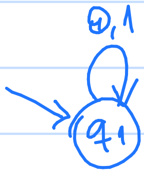
não é AFD

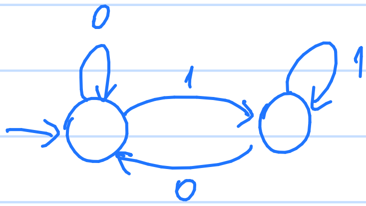




Mais exemplos

Dê um AFD para $L_4 = \emptyset$, $\Sigma = \{0,1\}$

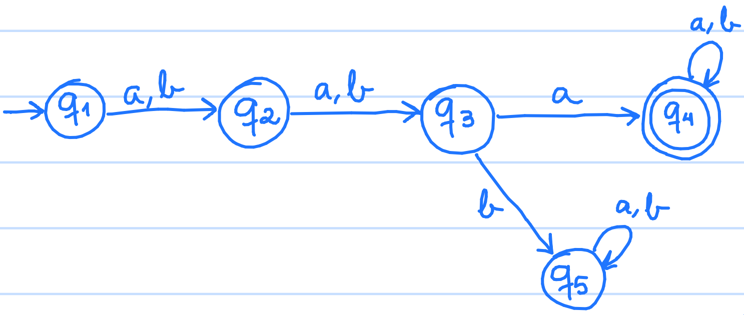




Mais exemplos

Dê um AFD para $L_5 = \{ w \in \{a,b\}^* : |w| \geq 3$ e o 3º símbolo de w é $a \}$

$\rightarrow w = \alpha a \beta$, $\alpha \in \{a,b\}^2$, $\beta \in \{a,b\}^*$



$\hat{\delta}(q_1, w) = q_1 \iff w = \varepsilon$

$\hat{\delta}(q_1, w) = q_2 \iff w \in \Sigma^1$

$\hat{\delta}(q_1, w) = q_3 \iff w \in \Sigma^2$

$\hat{\delta}(q_1, w) = q_4 \iff w = \alpha a \beta$, $\alpha \in \Sigma^2$ e $\beta \in \Sigma^*$

$\hat{\delta}(q_1, w) = q_5 \iff w = \alpha b \beta$, $\alpha \in \Sigma^2$ e $\beta \in \Sigma^*$