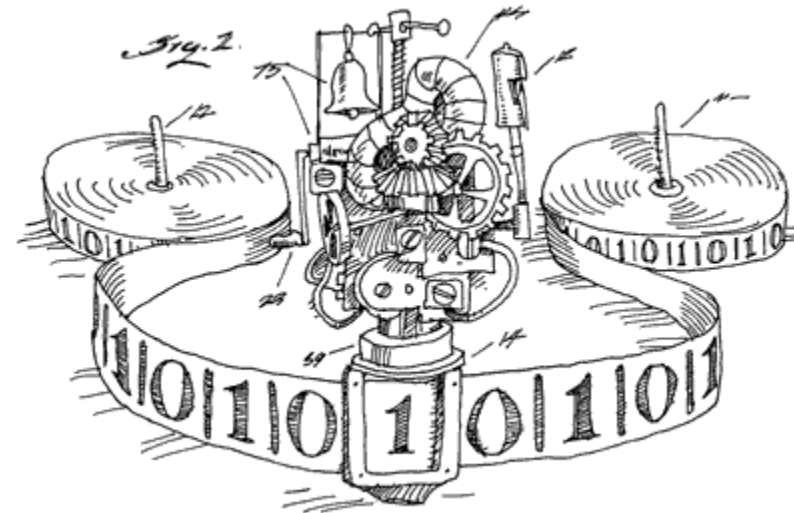


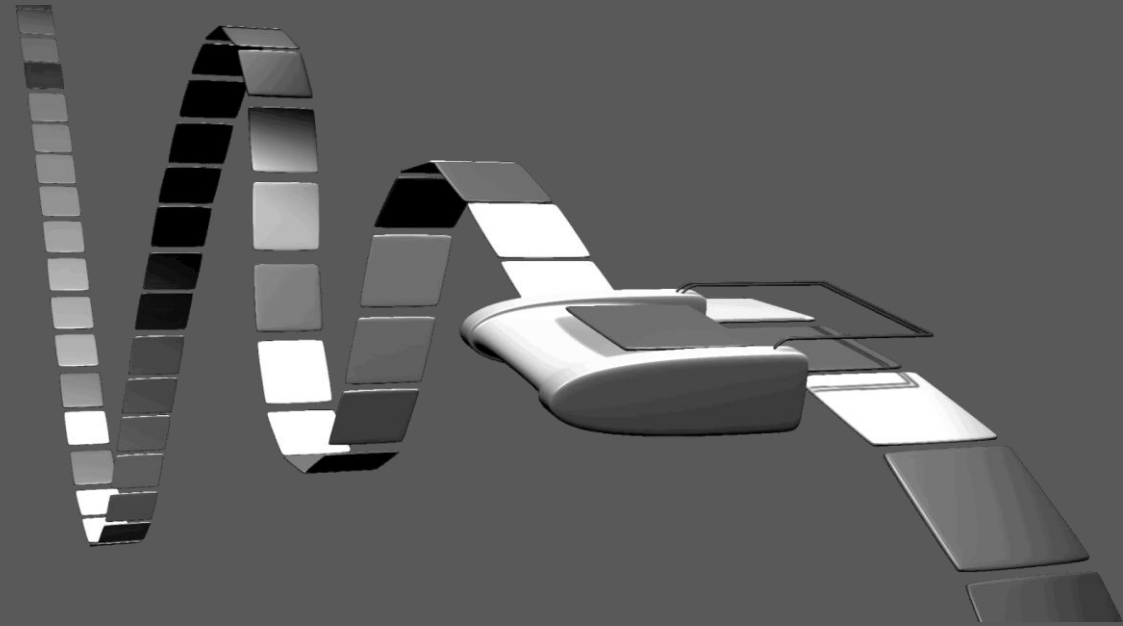
Autômatos Finitos Determinísticos

Introdução à Teoria da Computação

Nicollas Mocelin Sdroievski



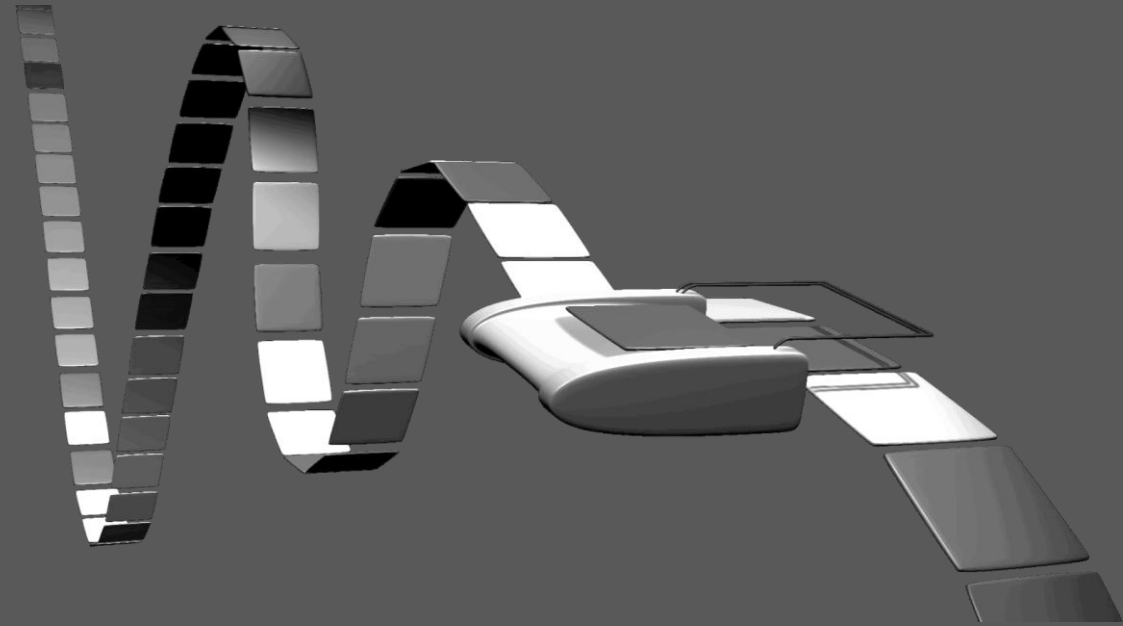
Relembrando



Alfabetos, Strings e Linguagens

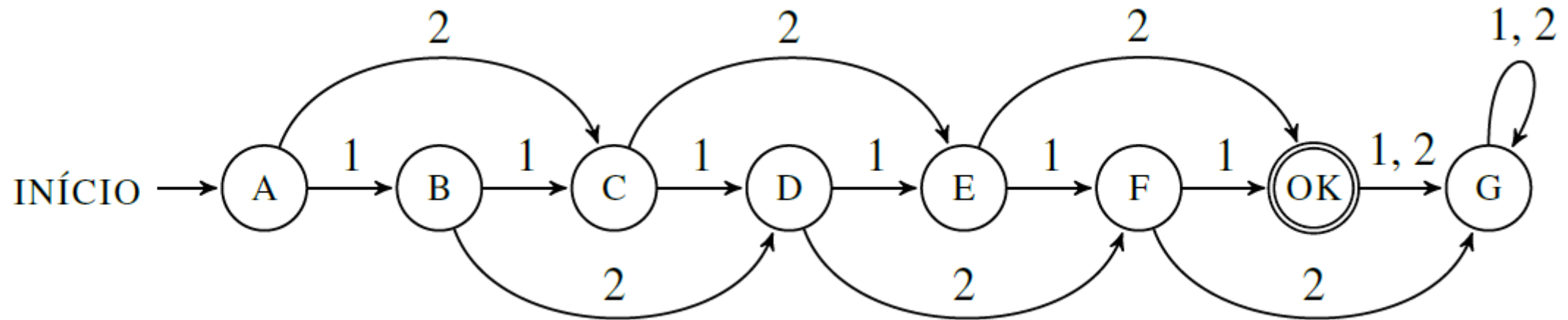
- Alfabeto Σ : conjunto de símbolos
 - $\Sigma = \{0,1\}$, $\Sigma = \{a,b,c,\dots,z\}$
- String w : sequência de símbolos
 - Pode ser vazia (ϵ)
 - 0011 , 1^{10} , $(01)^{100}$, *abacate*
- Linguagem L : conjunto de strings
 - $\{\epsilon, 0,1\}$
 - $\{0,1\}^*$
 - $\{s \mid s \text{ é uma palavra do português}\}$
- Linguagens e problemas de decisão

AFDs



Autômatos Finitos Determinísticos

- Considere uma máquina que vende chocolates de R\$6 e aceita moedas de R\$1 e R\$2



AFDs intuitivamente

- Diagramas como o da figura anterior
- A entrada é uma sequência de moedas
 - Ou uma string cujos símbolos são moedas (1 e 2)
- Indicação de início
- Indicação de “aceitação”

AFDs na teoria

- **Definição.** Um autômato Finito Determinístico (AFD), é uma 5-tupla $D = (Q, \Sigma, \delta, q_0, F)$, tal que:
 - Q é o conjunto de estados
 - Σ é o conjunto de símbolos de entrada
 - $\delta : Q \times \Sigma \rightarrow Q$ é a função de transição
 - $q_0 \in Q$ é o estado inicial
 - $F \subseteq Q$ é o conjunto de estados finais
- Definição versus instância

AFD para a máquina de vender chocolates

- Podemos definir matematicamente o autômato da máquina de chocolates através do AFD $C = \{Q, \Sigma, \delta_c, A, F\}$ a seguir

$Q = \{A, B, C, D, E, F, OK, G\};$

$\Sigma = \{1, 2\};$

$\delta_c : Q \times \Sigma \rightarrow Q$ é a função definida caso a caso a seguir:

$\delta_c(A, 1) = B, \delta_c(A, 2) = C, \delta_c(B, 1) = C, \delta_c(B, 2) = D, \delta_c(C, 1) = D, \delta_c(C, 2) = E,$
 $\delta_c(D, 1) = E, \delta_c(D, 2) = F, \delta_c(E, 1) = F, \delta_c(E, 2) = OK, \delta_c(F, 1) = OK, \delta_c(F, 2) = G,$
 $\delta_c(OK, 1) = G, \delta_c(OK, 2) = G, \delta_c(G, 1) = G, \delta_c(G, 2) = G.$

O estado A é o estado inicial;

O conjunto unitário $\{OK\}$ é o conjunto de estados finais.



Exercício resolvido

Exercício 3.1 (Resolvido) Desenhe o diagrama do autômato A definido a seguir:

$A = (Q, \Sigma, \delta, p, \{r\})$, tal que $Q = \{p, q, r\}$, $\Sigma = \{0, 1\}$ e a função δ é definida abaixo:

$$\delta(p, 1) = p,$$

$$\delta(p, 0) = q,$$

$$\delta(q, 0) = q,$$

$$\delta(q, 1) = r,$$

$$\delta(r, 0) = r,$$

$$\delta(r, 1) = r.$$

Solução:

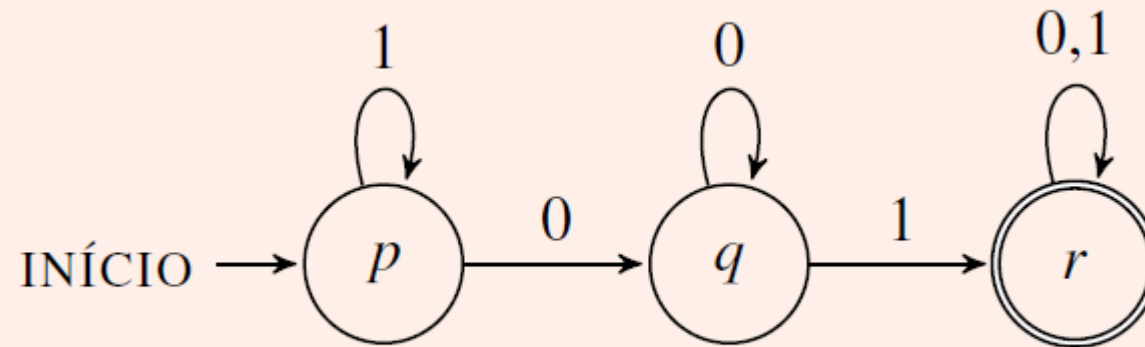
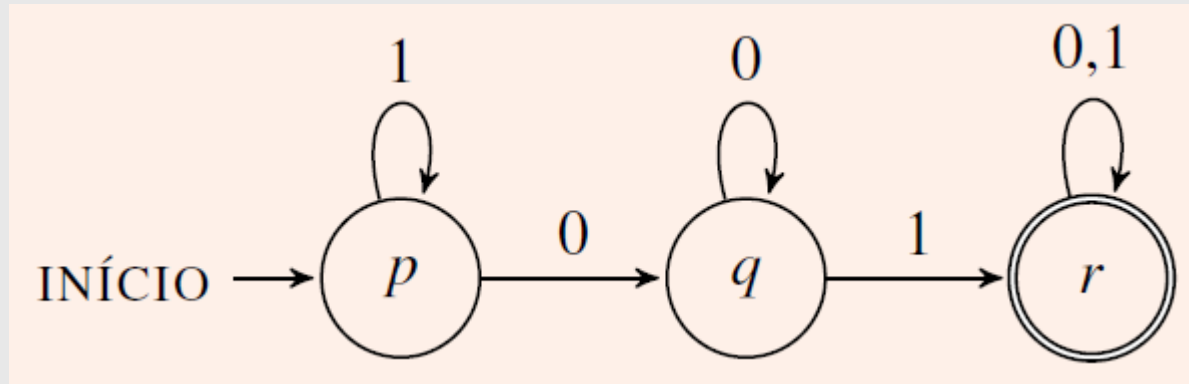


Tabela de transições

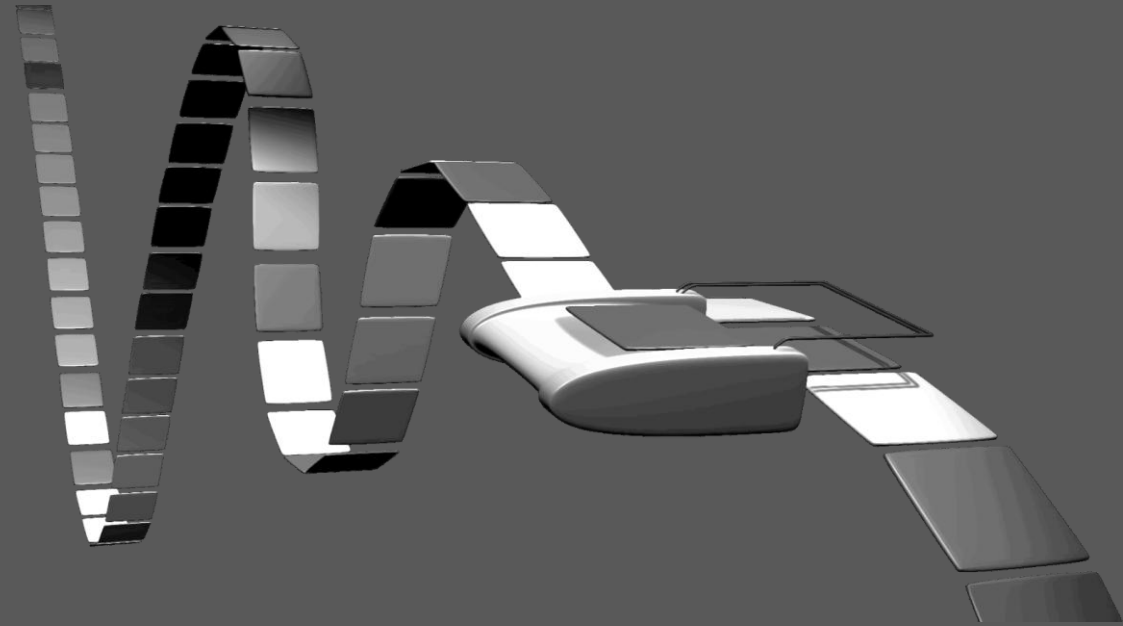
- Facilita a apresentação da função de transição δ

	0	1
$\rightarrow p$	q	p
q	q	r
$*r$	r	r



- Seta ao lado de p indica que é o estado inicial
- Asterisco ao lado de r indica que é um estado final

Aceitação e
rejeição de strings



Intuição

- Interpretação de um AFD (5-tupla) como um modelo de computação
 - Um AFD recebe uma string $w = a_1 a_2 \dots a_n$ como entrada
 - Processa cada um dos símbolos e produz uma saída
- Mais precisamente
 - O primeiro símbolo de w é lido
 - Ocorre uma mudança de estado
 - O símbolo lido é descartado
 - Reposta depende do estado atingido após a última transição
 - Se estado final, aceita. Senão, rejeita

Exercícios resolvidos

Exercício 3.2 (Resolvido) Seja $\Sigma = \{0, 1\}$. Considere a linguagem binária L das strings que contém a substring 01, ou seja, $L = \{w : w \text{ é da forma } x01y, \text{ sendo que } x, y \in \Sigma^*\}$. Construa um AFD que aceite todas e somente as strings de L .

Exercício 3.3 (Resolvido) Forneça um AFD com alfabeto binário que aceite as strings w tal que $N(w)$ é um múltiplo de 3.

Definição formal

- Com a função de transição, podemos responder à pergunta:
 - Dado um estado e um símbolo, qual o próximo estado?
- Para entender a computação do autômato, devemos responder à pergunta:
 - Dado um estado e uma string, qual é o estado resultante?
- Vamos formalizar essa ideia com a função de transição *estendida* $\hat{\delta}$

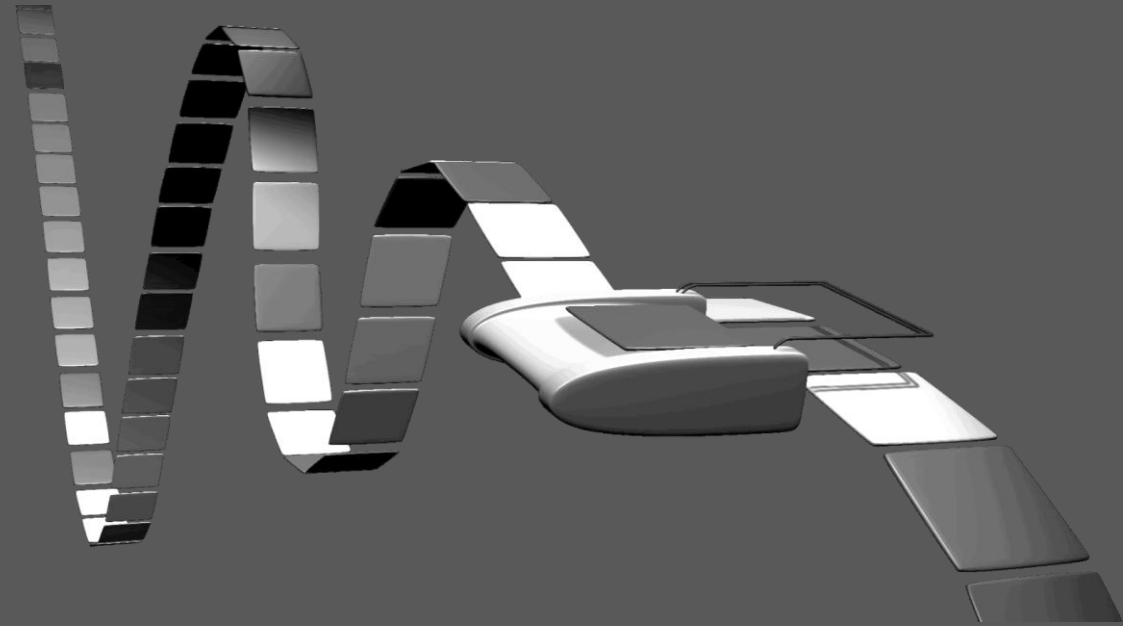
Definição formal

- Lembre que $\delta : Q \times \Sigma \rightarrow Q$
- Queremos $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$
- **Definição.** Dado um AFD $D = (Q, \Sigma, \delta, q_0, F)$, a função de transição estendida $\hat{\delta}$ de D é a função $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$ definida indutivamente:
 - **Caso base** $w = \varepsilon$
 - $\hat{\delta}(q, \varepsilon) = q$
 - **Indução** $|w| > 0$
 - $w = xa, x \in \Sigma^*$ e $a \in \Sigma$. Então $\hat{\delta}(q, w) = \hat{\delta}(q, xa) = \delta(\hat{\delta}(q, x), a)$

Formalizando aceitação/rejeição

- **Definição (aceitação e rejeição).** Seja $D = (Q, \Sigma, \delta, q_0, F)$ um AFD e $w \in \Sigma^*$.
 - Se $\hat{\delta}(q_0, w) \in F$, então dizemos que D aceita a string w .
 - Se $\hat{\delta}(q_0, w) \notin F$, então dizemos que D rejeita a string w .
- **Definição (Linguagem de um AFD).** A linguagem de um AFD D , denotada por $L(D)$, é definida como $L(D) = \{w \mid \hat{\delta}(q_0, w) \in F\}$.
- **Definição (Linguagem regular).** Dada uma linguagem L , se existe um AFD D tal que $L = L(D)$, então L é dita uma linguagem regular.

Exercícios



Exercícios

Exercício 3.5 Seja $\Sigma = \{0, 1\}$. Construa AFDs que aceite a string fornecida como entrada se, e somente se, a string pertence a linguagem L , para cada um dos casos abaixo:

(a) $L = \{w : w \text{ tenha um número ímpar de } 1\text{'s}\}$.

- Recomendo as letras restantes

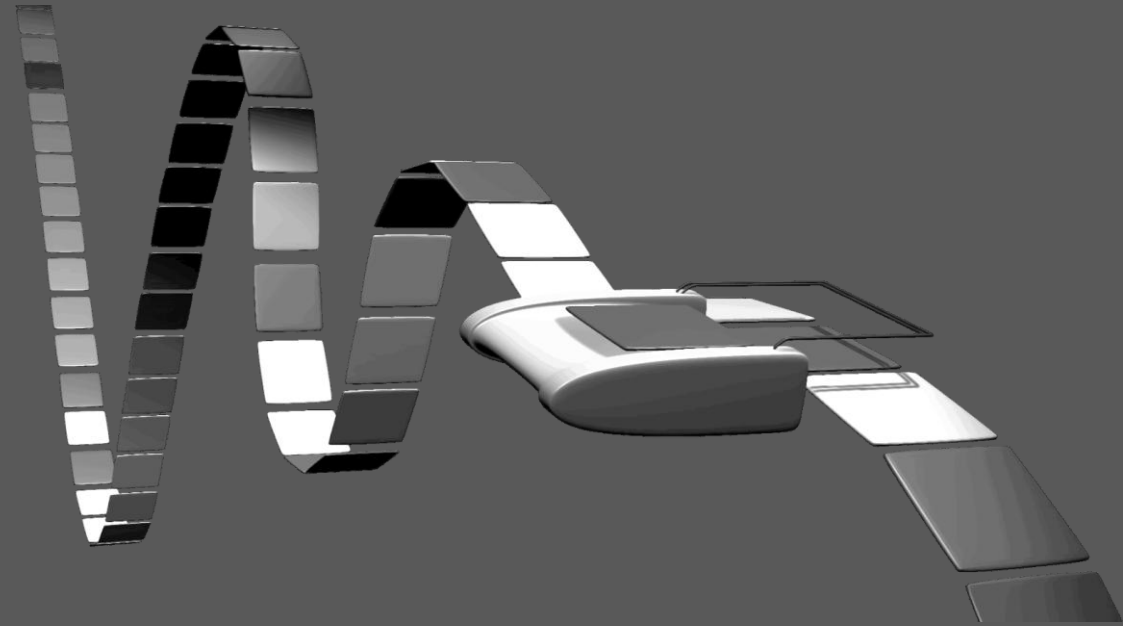
Exercício 3.6 Seja um AFD com alfabeto Σ e conjunto de estados Q . Sejam $x, y \in \Sigma^*$, $a \in \Sigma$ e $q \in Q$ quaisquer.

(a) Mostre que $\hat{\delta}(q, xy) = \hat{\delta}(\hat{\delta}(q, x), y)$.

(b) Mostre que $\hat{\delta}(q, ax) = \hat{\delta}(\delta(q, a), x)$.



Para fechar



Para fechar

- Hoje
 - AFDs
 - Nosso primeiro modelo de computação
 - Capaz de reconhecer as linguagens regulares
- Próxima aula
 - Autômatos finitos não-determinísticos
 - Variação não-determinística de AFDs

Referências

- [SIL25] – SILVA, M.V.G. Autômatos, Computabilidade e Complexidade Computacional , 2025.