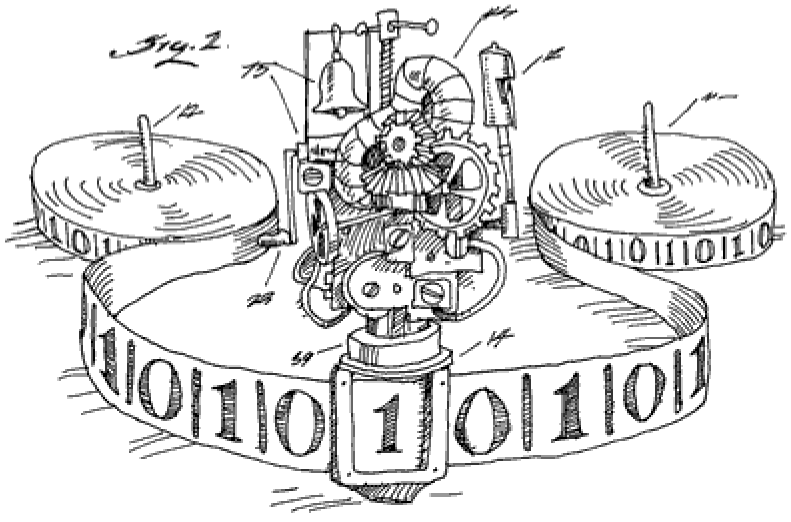
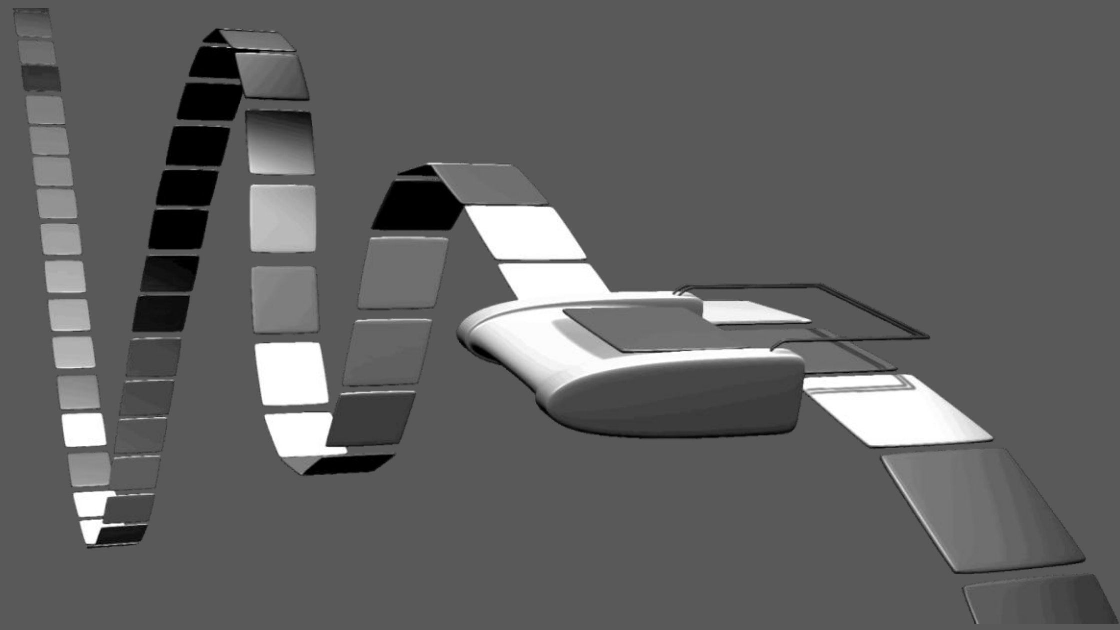
Máquinas de Turing

Introdução à Teoria da Computação

Nicollas Mocelin Sdroievski



Relembrando



Problemas de decisão

- Linguagens = Problemas de decisão

■ **Exemplo 2.10** $L_{01} = \{\varepsilon, 01, 0011, 000111, \ldots\} = \{w \in \Sigma^* : w$ é da forma $0^n 1^n$, para $n \geq 0\}$. ■

■ **Exemplo 2.11** $L_{\mathrm{PAL}} = \{\varepsilon, 0, 1, 00, 11, 010, 101, 000, 111, \ldots\} = \{w \in \Sigma^* : w$ é um palíndromo$\}$. ■

■ **Exemplo 2.12** $L_{\mathrm{P}} = \{10, 11, 101, 111, 1011, \ldots\} = \{w \in \Sigma^* : w$ é um número primo escrito em binário$\}$. ■

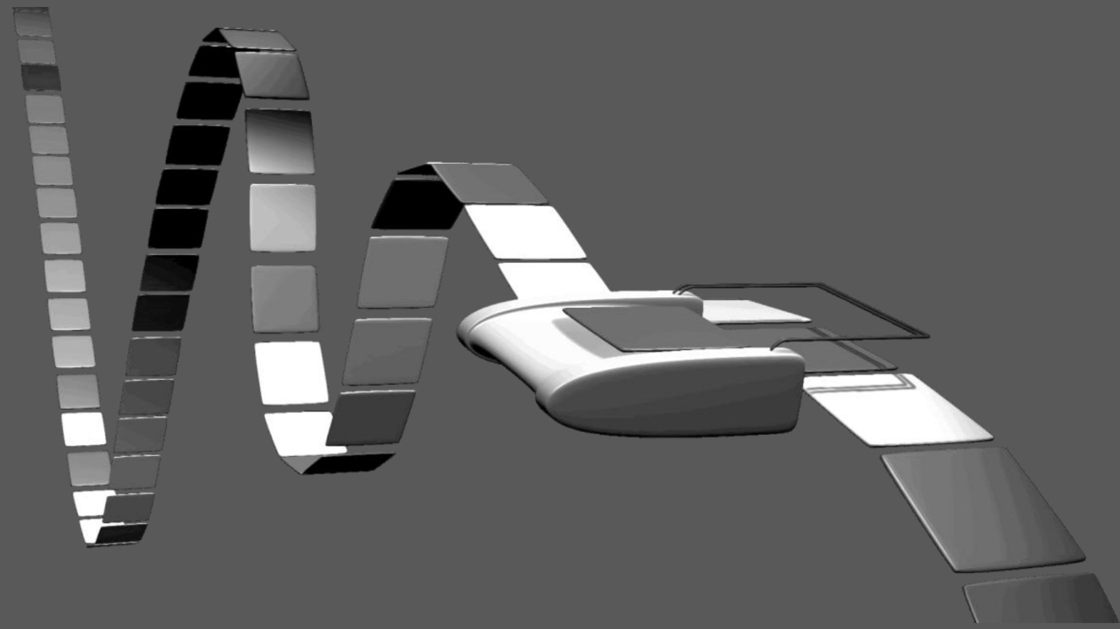
■ **Exemplo 2.13** $L_{\mathrm{SQ}} = \{0, 1, 100, 1001, 10000, \ldots\} = \{w \in \Sigma^* : w$ é um número quadrado perfeito escrito em binário$\}$. ■

- Logo, veremos alguns problemas que não são de decisão também

Expressividade de modelos de computação

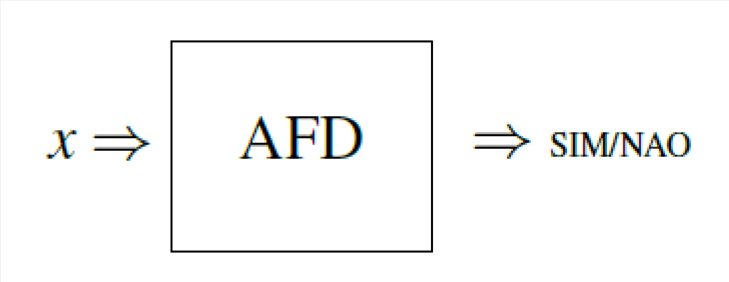
- AFDs e amigos aceitam as linguagens regulares
- APs aceitam as linguagens livres de contexto
- Vimos que existem linguagens que são livres de contexto porém não são regulares
- A linguagem $L = \{a^n b^n c^n \mid n \geq 1\}$ não é uma linguagem livre de contexto (exercício 4.13)
- Como existe um algoritmo simples que resolve o problema computacional de L, precisamos ir além para capturar computação como um todo

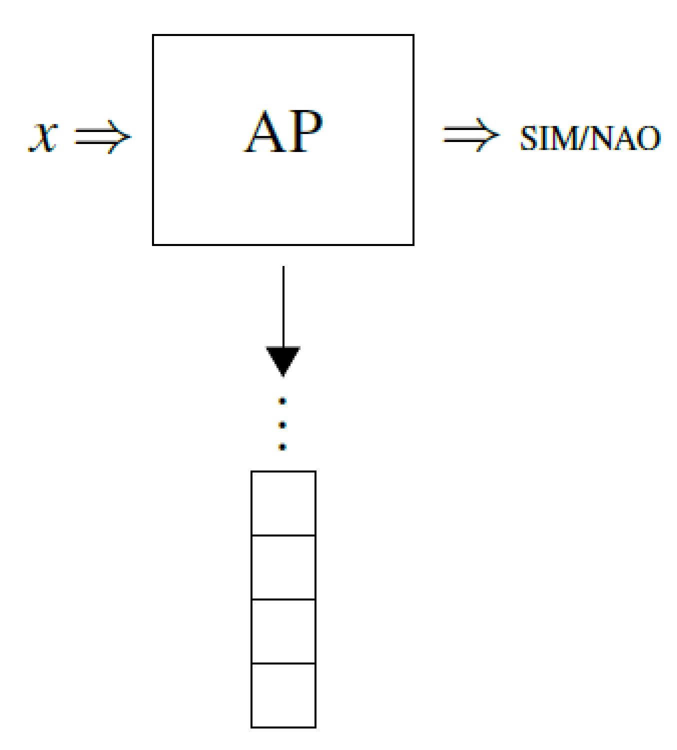
Máquina de Turing

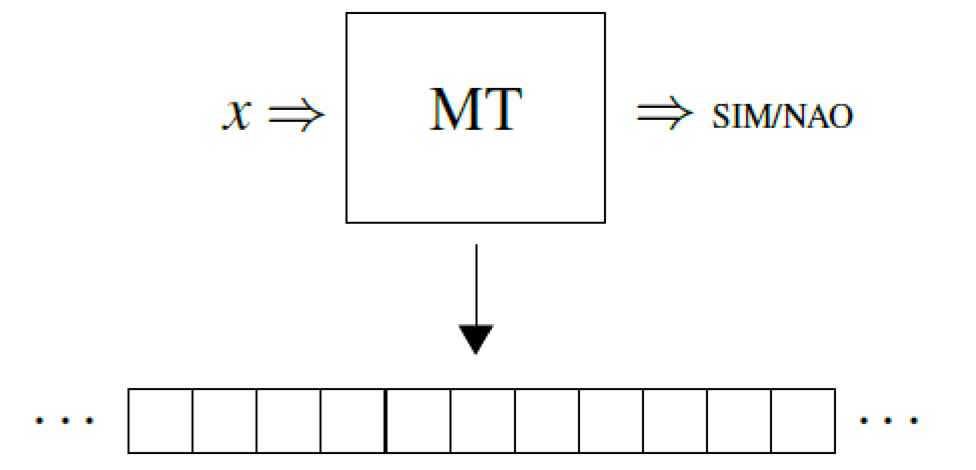


Intuição

- AFD: Máquina de estados
- AP: Máquina de estados com uma pilha infinita
- MT: Máquina de estados com uma fita infinita







String de entrada

- Autômatos finitos e com pilha
  - Processam a string de entrada símbolo por símbolo
- Uma Máquina de Turing começa com a string de entrada na sua fita de dados
- A cada passo, pode
  - Mudar de estado
  - Trocar o símbolo na fita por outro
  - “Olhar” para o símbolo da esquerda ou direita (ou continuar “olhando” para o mesmo símbolo)

Definição

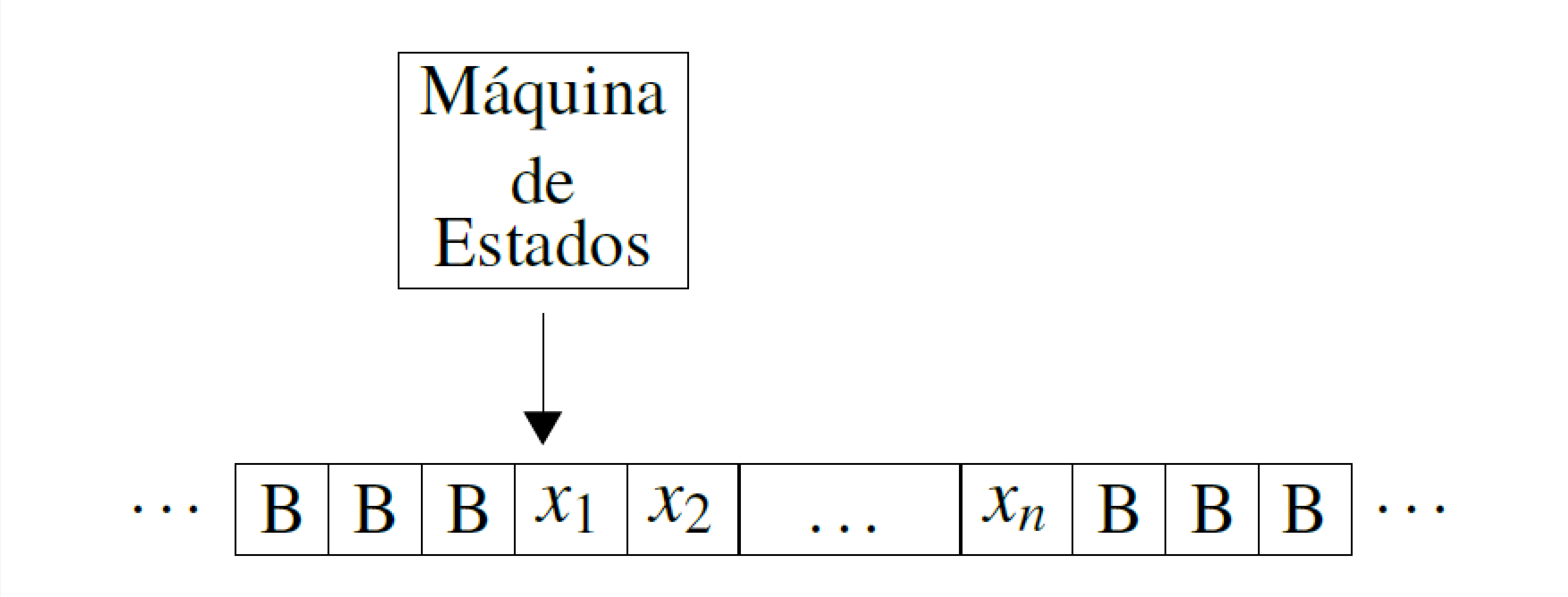
- **Definição.** Uma máquina de Turing (MT) é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, tal que
  - Q é o conjunto finito de estados
  - Σ é o alfabeto de entrada. Quando não especificado, $\Sigma = \{0,1\}$
  - Γ é o alfabeto da fita, tal que $\Sigma \subseteq \Gamma$.
  - $\delta: (Q \setminus \mathrm{F}) \times \Gamma \rightarrow Q \times \Gamma \times D$ é uma função parcial e $D = \{L, R, S\}$.
  - q_0 é o estado inicial.
  - B é o símbolo especial chamado de *símbolo branco*.
  - $F \subseteq Q$ é o conjunto de estados finais

Funcionamento de uma MT

- Suponha que $\delta(q, X) = (q', X', d)$, então
  - A MT estava no estado q, e o símbolo X está sendo escaneado na fita
  - A MT mudará para o estado q'
  - A MT sobrescreve o símbolo X pelo símbolo X'
  - A MT move sua cabeça de leitura/gravação para a direção indicada por d
    1. Se $d = L$, para a esquerda
    2. Se $d = R$, para a direita
    3. Se $d = S$, fica na posição atual

Início da computação

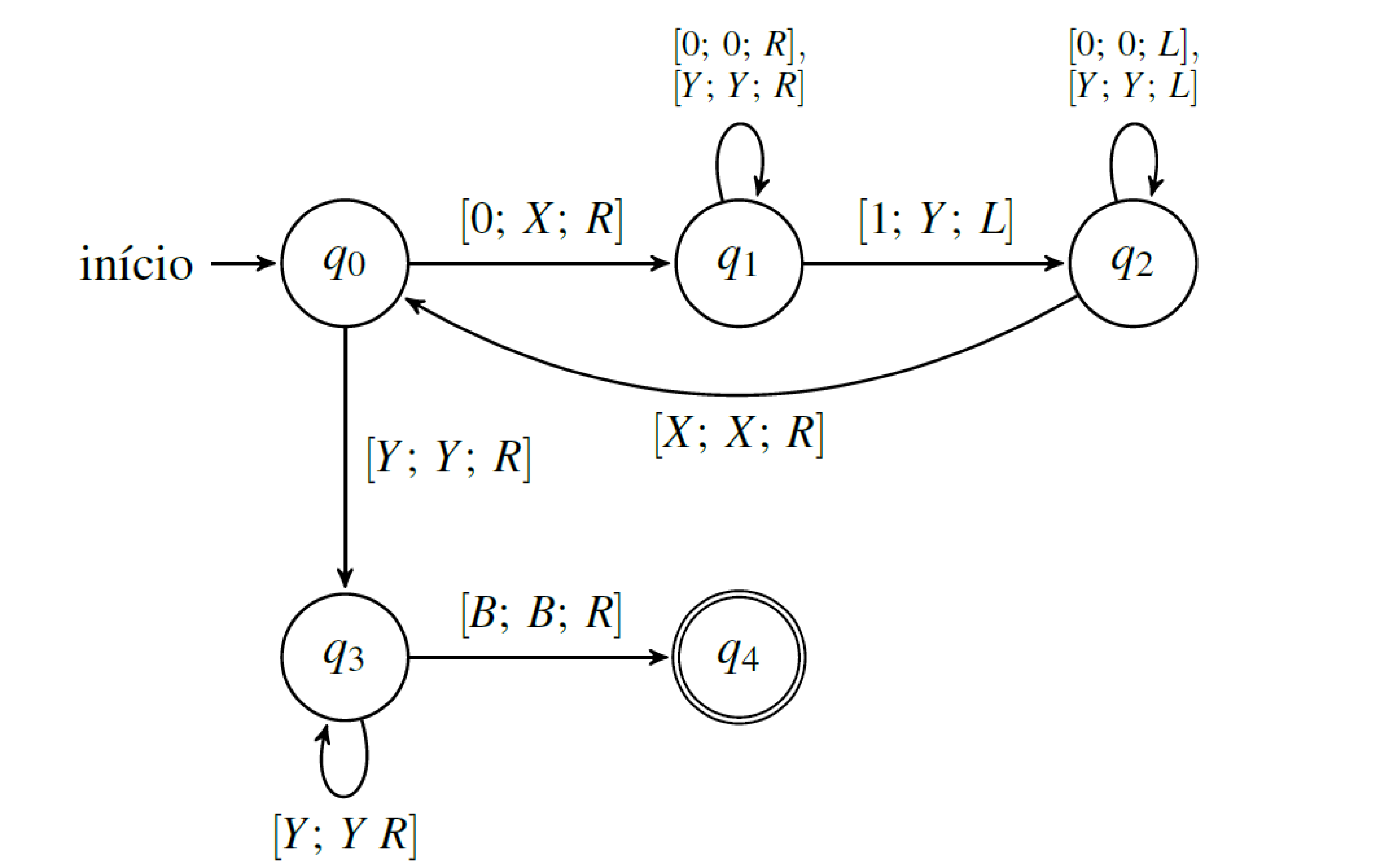
- Se a entrada é $x = x_1 x_2 \dots x_n$, então, no início da computação.
  - A MT está no estado inicial
  - x se encontra na fita e a cabeça de leitura/gravação está posicionada sobre x_1



Determinismo

- A definição de MT que vimos é determinística
- Não é necessário (nem desejável) que δ esteja definida para todos os pares possíveis (q, X)
- Se δ não está definida para (q, X), então a máquina *para* sua execução
  - Se parou em um estado final, aí a máquina aceita a string
  - Se parou em um estado não final, a máquina rejeita

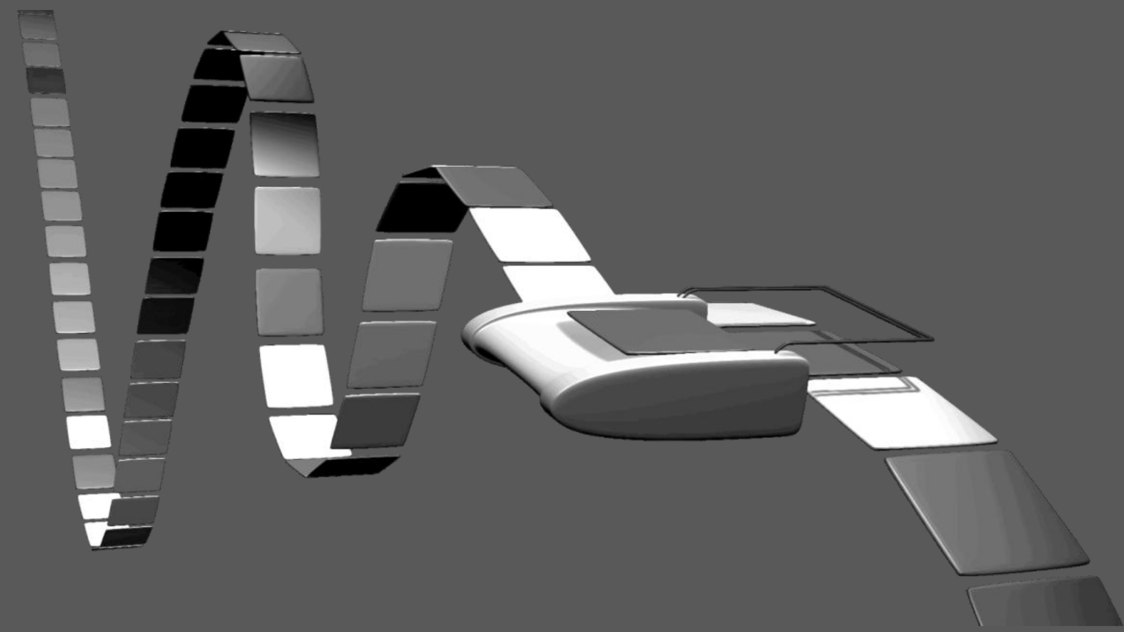
Diagrama de estados



Exercício

- Para a MT do slide anterior,
  - $\Sigma = \{0,1\}$
  - $\Gamma = \{0,1,X,Y,B\}$
- Qual a linguagem aceita pela MT?
- Estado q_0: Lê um símbolo 0, marca a posição com X e vai para q_1
- Estado q_1: Move a cabeça de leitura/gravação para a direita até encontrar um símbolo 1. Quando encontra, marca com Y e vai para q_2
- Estado q_2: Retorna até o último X marcado, move para a direita e recomeça o processo no estado q_0. Quando não houver mais 0's para marcar, vai para q_3
- Estado q_3: Move a cabeça de leitura/gravação à direita para verificar que não restaram símbolos 1 na fita. Se for verdade, vai para o estado de aceitação q_4
- A linguagem é $\{0^n1^n \mid n \geq 1\}$

Definição formal de computação



Configuração

- Assim como para APs, configurações são úteis para definir a computação de uma máquina de Turing
- **Definição.** Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, uma configuração de M é uma tripla (α, q, β) tal que $\alpha, \beta \in \Gamma^*$ e $q \in Q$.
- Para uma MT M, a configuração $(X_1X_2 \dots X_{i-1}, q, X_iX_{i+1} \dots X_n)$ indica que M, depois de processar parte da string de entrada
  - Está no estado q
  - A fita contém os símbolos $\dots BX_1X_2 \dots X_nB \dots$
  - A cabeça de leitura está posicionada sobre o símbolo X_i

Computação

- Vamos definir os símbolos $\vdash_M$ e $\vdash_M^*$, que representam um e uma sequência de passos computacionais, respectivamente
- **Definição (Passo computacional à esquerda).** Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1X_2 \dots X_{i-1}, q, X_iX_{i+1} \dots X_n)$ uma configuração de M. Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1X_2 \dots X_{i-1}, q, X_iX_{i+1} \dots X_n) \vdash_M^{\ell} (X_1X_2 \dots X_{i-2}, p, X_{i-1}YX_{i+1} \dots X_n)$$

Exceto nos casos (1) $i = 1$ e (2) $i = n$ e $Y = B$, casos em que

- (1) Se $i = 1$, escrevemos $(\varepsilon, q, X_1 \dots X_n) \vdash_M^{\ell} (\varepsilon, p, BYX_2 \dots X_n)$
- (2) Se $i = n$ e $Y = B$, escrevemos $(X_1X_2 \dots X_{n-1}, q, X_n) \vdash_M^{\ell} (X_1X_2 \dots X_{n-2}, p, X_{n-1})$

Computação

Exercício 5.1 Apresente definições para os símbolos $\vdash_M^r$ e $\vdash_M^s$ de forma que eles se refiram a passos computacionais à esquerda e estático, para os casos em que $\delta(q, X_i) = (p, Y, R)$ e $\delta(q, X_i) = (p, Y, S)$, respectivamente. ■

- **Definição (Passo computacional de uma MT).** Dada uma MT M, um *passo computacional de* M, denotado por $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^\ell, \vdash_M^S$ ou $\vdash_M^r$
- **Definição.** Dada uma MT M, o símbolo $\vdash_M^*$ é definido indutivamente
  - **Base:** $I \vdash_M^* I$ para qualquer configuração I de M.
  - **Indução:** $I \vdash_M^* J$ se existe K tal que $I \vdash_M K$ e $K \vdash_M^* J$.

Aceitação/Rejeição

- Para todas as definições, Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $w \in \Sigma^*$.

- **Definição.** A configuração (ε, q_0, w) é chamada de *configuração inicial* de M com w. Se $p \in F$, então qualquer configuração da forma (α, p, β) onde $\alpha, \beta \in \Gamma^*$, é chamada de *configuração final* de M com w.

- **Definição.** Dizemos que w é *aceita* por M se $(\varepsilon, q_0, w) \vdash_M^* I_F$, onde I_F é uma configuração final de M. Caso contrário, dizemos que M *não aceita* w.

- **Definição.** Dizemos que w é *rejeitada* por M se $(\varepsilon, q_0, w) \vdash_M^* I_N$, tal que I_N não é uma configuração final de M e M para ao atingir a configuração I_N.

Linguagens de uma MT

- **Definição (Linguagem de MTs).** Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$L(M) = \{w \in \Sigma^* \mid (\varepsilon, q_0, w) \vdash_M^* \ C_F,$

onde C_F é uma configuração final de $M\}$, é chamada de linguagem de M

- **Definição (Linguagem decididas por MTs).** Seja L uma linguagem. Se existe uma *MT que sempre para* M tal que $L(M) = L$, dizemos que M ***decide*** L.

- **Definição (Linguagem aceitas por MTs).** Seja L uma linguagem. Se existe uma *MT* M tal que $L(M) = L$, dizemos que M ***aceita*** L.

Mais tipos de linguagens

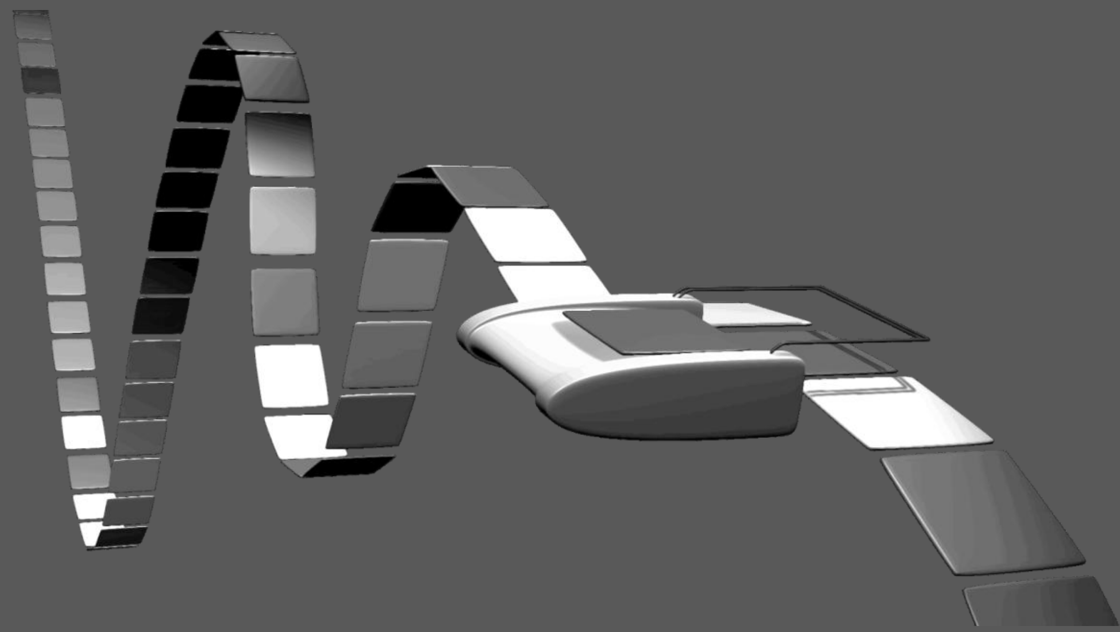
- Uma linguagem que pode ser *decidida* por uma MT é dita uma linguagem ***Recursiva***, ou ***Decidível***

- Uma linguagem que pode ser *aceita* por uma MT é dita uma linguagem ***Recursivamente Enumerável***

- $\mathcal{R}$ = conjunto das linguagens recursivas
- $\mathcal{RE}$ = conjunto das linguagens recursivamente enumeráveis

Exercício 5.4 Mostre que $\mathscr{R} \subseteq \mathscr{RE}$. ■

Algoritmos



Algoritmos

- **Definição.** Um algoritmo é uma máquina de Turing que sempre para
  - Independente da string de entrada, a máquina sempre para, aceitando ou rejeitando a string
  - Segue que $\mathcal{R}$ é o conjunto dos problemas de decisão decidíveis por algoritmos

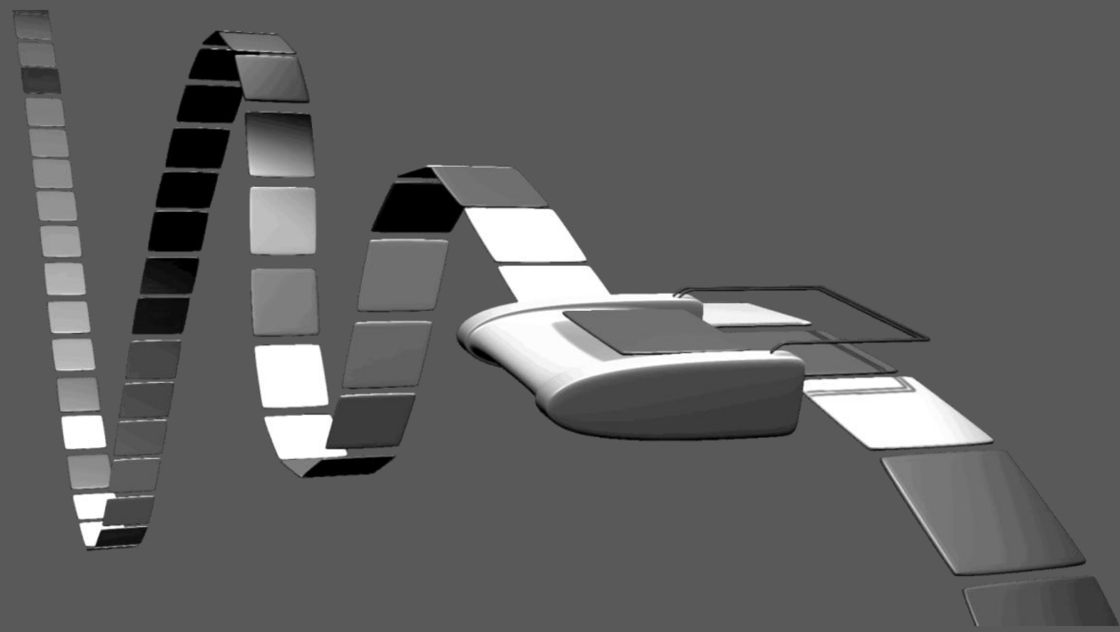
Exercícios

Exercício 5.6 (Resolvido) Projete uma MT que decida se a string de entrada pertence a linguagem $L_{abc} = \{a^n b^n c^n : n \geq 1\}$, sobre o alfabeto $\Sigma = \{a, b, c\}$.

Exercício 5.7 Forneça uma MT $M_{\text{COPY}} = (C, \Sigma, \Gamma, \delta_{\text{COPY}}, c_0, B, F_{\text{COPY}})$ que tenha o seguinte comportamento quando uma string x é fornecida como entrada. A máquina deve adicionar ao

Exercício. Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT. Mostre que existe uma MT M' com o seguinte comportamento: se M aceita x, então M' entra em loop infinito na entrada x.

Para fechar



Para fechar

- Hoje
  - Máquinas de Turing
    - Nosso último modelo de computação (essencialmente)
    - Aceitação, não aceitação e rejeição de strings
    - Linguagens recursivas e recursivamente enumeráveis
- Próxima aula **(só no dia 23/04)**
  - Variações de máquinas de Turing
  - Computabilidade

Referências

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