

# Máquinas de Turing

Teoria da Computação – Ciência da Computação

Prof. Daniel Saad Nogueira Nunes



**INSTITUTO  
FEDERAL**

Brasília

---

Campus  
Taguatinga

# Sumário

Introdução

Definição

Linguagens Decidíveis e Reconhecíveis

# Máquinas de Turing

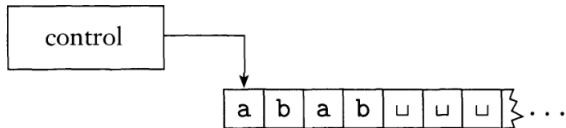
## Máquinas de Turing

- ▶ Modelo Proposto por Alan Turing em 1936.
- ▶ Máquina de estados + memória infinita.
- ▶ Considera-se que faz tudo o que é possível fazer com um computador.

# Máquinas de Turing

- ▶ Neste modelo possuímos uma fita infinita e um controle finito.
- ▶ A máquina possui uma cabeça de leitura e escrita que pode ler/escrever símbolos na fita e movimentar-se sobre ela.
- ▶ Inicialmente, a fita contém apenas a cadeia de entrada, e está em branco.

# Máquinas de Turing



# Máquinas de Turing

- ▶ Se a máquina precisa armazenar alguma informação, ela pode escrever algo na fita.
- ▶ Para acessar uma informação, basta posicionar a cabeça na posição correta e ler aquele símbolo.

# Máquinas de Turing

- ▶ A computação é feita ao alterar os estados do controle com base na informação contida na fita.
- ▶ A computação para quando chega em um estado de aceitação ou de rejeição. Neste caso a saída da máquina para uma entrada pode ser:
  - ▶ Aceita.
  - ▶ Rejeita.

# Máquinas de Turing

## Recapitulando

- ▶ Uma máquina de turing pode tanto escrever sobre a fita quanto ler.
- ▶ A cabeça de leitura/escrita pode mover tanto para a esquerda quanto para a direita um passo de cada vez.
- ▶ A fita é infinita.
- ▶ Os estados especiais para aceitar e rejeitar tem efeito imediato.



# Máquinas de Turing

- ▶ Tome uma máquina de Turing  $M_1$  para testar a pertinência na linguagem:

$$B = \{w\#w \mid w \in \{0, 1\}^*\}$$

- ▶ A linguagem das palavras que são seguidas por elas próprias e separadas por cerquilha.
- ▶ Como  $M_1$  funcionaria de acordo com a nossa descrição informal?

# Máquinas de Turing

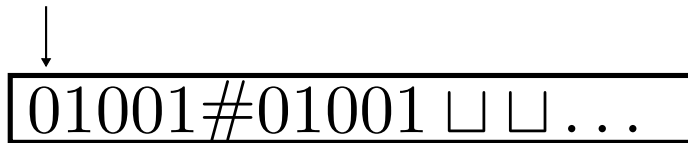
- ▶ Não podemos acessar cada símbolo através de acesso aleatório.
- ▶ Mas podemos mover sobre a fita e ler/escrever nela (acesso sequencial).

# Máquinas de Turing

## Estratégia Imediata

- ▶ Zigzaguear sobre os dois lados da cerquilha e comparar os caracteres.

# Máquinas de Turing



# Máquinas de Turing

## Estratégia Imediata

- ▶ Faça uma varredura na entrada para assegurar que ela contém uma única ocorrência do símbolo  $\#$ . Se não, **rejeite**.
- ▶ Faça um zigue-zague na fita para casar os símbolos nos dois lados da separação pelo  $\#$ . Se os símbolos casam, marque eles com um 'x', caso contrário, **rejeite**.
- ▶ Quando todos os símbolos da esquerda forem marcados com um 'x', verifique se existe algum símbolo diferente de 'x' e  $\square$  à direita. Se resta algum, **rejeite**, caso contrário, aceite.

# Máquinas de Turing

- Qual seria a estratégia de uma Máquina de Turing  $M_2$  para testar a pertinência na Linguagem

$$B = \{w\#w^R \mid w \in \{0, 1\}^*\}$$

# Sumário

Introdução

Definição

Linguagens Decidíveis e Reconhecíveis

# Sumário

## Definição

Definição Formal

Representação Gráfica

Configuração



# Máquinas de Turing

- ▶ Agora que ganhamos uma noção sobre máquinas de Turing, vamos defini-las formalmente.

# Máquinas de Turing

- ▶ O coração de uma máquina de turing é uma função de transição  $\delta$ .
- ▶ Ela que especifica como a máquina vai de um passo a próximo.

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

- ▶ Dado um estado e um símbolo na posição apontada pela cabeça,  $\delta$  mapeia em outro estado, escreve um símbolo na fita de acordo com a posição da cabeça e move para esquerda ou para direita.
- ▶ Ex:  $\delta(q, a) = (r, b, L)$ .

# Máquinas de Turing

- ▶ Ex:  $\delta(q, a) = (r, b, L)$ .
- ▶ A máquina escreve o símbolo  $b$  no lugar de  $a$ , muda do estado  $q$  para o estado  $r$  e move a cabeça uma posição para a esquerda.

# Máquinas de Turing

## Definição (Máquinas de Turing)

*Uma máquina de turing é uma 7-upla  $(Q, \Sigma, \Gamma, \delta, q_0, q_{aceita}, q_{rejeita})$ , em que  $Q$ ,  $\Sigma$  e  $\Gamma$  são conjuntos finitos.*

- ▶  $Q$ : o conjunto de estados.
- ▶  $\Sigma$ : o alfabeto de entrada, que não contém o símbolo  $\sqcup$ .
- ▶  $\Gamma$  o alfabeto da fita, onde  $\sqcup \in \Gamma$  e  $\Sigma \subseteq \Gamma$ .
- ▶  $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
- ▶  $q_0$  é o estado inicial.
- ▶  $q_{aceita}$  é o estado de aceitação.
- ▶  $q_{rejeita}$  é o estado de rejeição.

# Máquinas de Turing

- ▶ Uma vez definida a sintaxe sobre máquinas de Turing, podemos discutir a semântica de todos estes símbolos.

# Máquinas de Turing

Uma máquina de Turing  $M$  computa da seguinte forma:

- ▶ Inicialmente ela recebe a sua entrada  $w = w_1w_2 \dots w_n \in \Sigma^*$ .
- ▶ Esta entrada é colocada mais à esquerda na fita.
- ▶ O restante da fita possui símbolos  $\sqcup$  (branco).
- ▶ Ou seja, a fita só está preenchida com a entrada, e ela está mais a esquerda possível.
- ▶ Como  $\sqcup \notin \Sigma$  o símbolo em branco marca o final da entrada.

# Máquinas de Turing

- ▶ Uma vez que  $M$  começa, a computação prossegue de acordo com  $\delta$ .
- ▶ Se  $M$  em algum momento tenta mover a cabeça para a esquerda quando esta está sob a posição mais à esquerda da fita, a cabeça permanece naquele lugar.
- ▶ A computação continua até que ela entre nos estados de aceitação ou rejeição, nos quais ela para.
- ▶ Se ela nunca atinge estes estados, a computação persiste indefinidamente }:-)

# Máquinas de Turing

- ▶ À medida que uma máquina de Turing computa, ocorrem mudanças no estado atual, no conteúdo da fita e na localização da cabeça.
  - ▶ Tudo definido pela função  $\delta$ .



# Sumário

## Definição

Definição Formal

**Representação Gráfica**

Configuração

# Máquinas de Turing: Representação Gráfica

- ▶ Agora que definimos uma máquina de Turing, vamos construir uma que **aceite** as palavras que estão em

$$B = \{w\#w^R \mid w \in \{0, 1\}^*\}$$

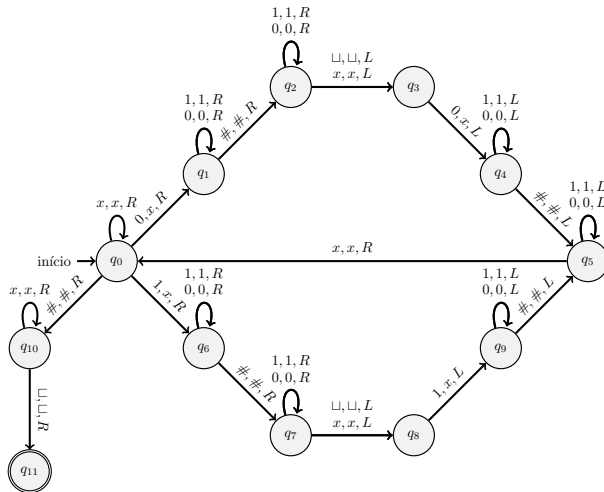
e **rejeite** as que não estão.

- ▶ Utilizaremos uma representação gráfica da nossa definição formal.

## Representação Gráfica

- ▶ Os círculos representarão os estados, sendo o estado inicial, e de aceitação explicitamente indicados. O estado inicial possui uma seta com o rótulo *início*, enquanto o estado de aceitação é formado por dois círculos concêntricos.
- ▶ Cada mapeamento da função de transição é representado por uma seta que liga um estado a outro. Esta seta possui um rótulo que indica que, ao ler um símbolo da fita, escreve-se outro no lugar, movimenta-se o cabeçote para a esquerda ou para a direita e, finalmente, vamos para outro estado.
- ▶ Para simplificar, podemos assumir que as transições que não constarem no gráfico, estão apontando para o estado de rejeição (implícito).

# Representação Gráfica



# Sumário

## Definição

Definição Formal

Representação Gráfica

Configuração

# Máquinas de Turing: Configuração

## Definição

*Configuração Uma configuração é uma combinação de três informações:*

- ▶ *O estado.*
- ▶ *O conteúdo da fita.*
- ▶ *A posição da cabeça.*

*Denotas uma configuração por  $uqv$ , em que  $q$  é o estado atual,  $uv$  é o conteúdo da fita, e a cabeça está apontando para o primeiro símbolo de  $v$ .*

# Configuração

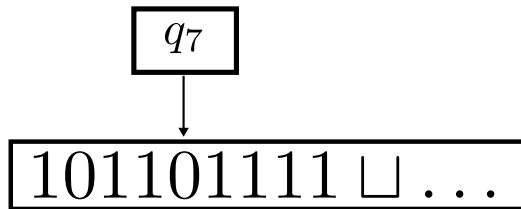


Figura: Configuração  $1011q_701111$ .

# Configuração

- ▶ Dizemos que uma configuração  $C_1$  produz uma  $C_2$  se a máquina pode legalmente sair de  $C_1$  e ir para  $C_2$ .
- ▶ Suponha  $a, b, c \in \Sigma$  e  $u, v \in \Sigma^*$ .
- ▶ Formalmente:  $u a q_i b v$  produz  $u q_j a c v$ , se  $\delta(q_i, b) = (q_j, c, L)$ .
- ▶ Simetricamente:  $u a q_i b v$  produz  $u a c q_j v$ , se  $\delta(q_i, b) = (q_j, c, R)$ .
- ▶ Exceção:  $q_i b v$  produz  $q_j c v$  se a transição for para a esquerda.



# Configuração

- ▶ A configuração inicial de  $M$  sobre a entrada é a configuração  $q_0w$ .
- ▶ Em uma configuração de aceitação, o estado da configuração é  $q_{aceita}$ .
- ▶ Em uma configuração de rejeição, o estado da configuração é  $q_{rejeita}$ .

# Configuração

## Definição (Aceitação)

*Uma máquina de Turing  $M$  aceita a entrada  $w$  se uma sequência de configurações  $C_1, C_2, \dots, C_k$  existe onde:*

- 1.  $C_1$  é a configuração inicial de  $M$  sobre  $w$ .*
- 2. Cada  $C_i$  produz  $C_{i+1}$ .*
- 3.  $C_k$  é uma configuração de aceitação.*

# Configuração

## Notação (Aceitação)

*A coleção de cadeias que  $M$  aceita é a linguagem de  $M$ , denotada por  $L(M)$ .*

# Sumário

Introdução

Definição

Linguagens Decidíveis e Reconhecíveis

# Linguagens Turing-reconhecíveis

## Definição (Linguagens Turing-Reconhecíveis)

*Uma linguagem  $L$  é chamada de Turing-reconhecível se existe alguma máquina de Turing que sempre chega a um estado de **aceitação** para toda entrada  $w \in L$ :*

*Se  $w \notin L$ , então  $M$  pode:*

- ▶ *Parar no estado de rejeição.*
- ▶ *Entrar em loop.*

# Linguagens Turing-decidíveis

## Definição (Linguagens Turing-decidíveis)

*Uma linguagem  $L$  é chamada de Turing-decidível se existe alguma máquina de Turing que sempre chega a um estado de aceitação ou rejeição para toda entrada  $w$ :*

- ▶ **Aceita** se  $w \in L$ .
- ▶ **Rejeita** se  $w \notin L$ .

# Propriedades de linguagens

## Observação

Reconhecibilidade e decidibilidade são propriedades de **linguagens**, não de máquinas de Turing.

**É incorreto** dizer que uma máquina é Turing-decidível ou Turing-reconhecível.

# Linguagens Turing-Decidíveis e Reconhecíveis

- ▶ Obviamente toda linguagem Turing-Decidível é Turing-Reconhecível, mas existem algumas linguagens que são Turing-Reconhecíveis, mas não Turing-Decidíveis.
- ▶ Qual a vantagem de Linguagens Turing-decidíveis?



# Linguagens Turing-Decidíveis e Reconhecíveis

- ▶ Obviamente toda linguagem Turing-Decidível é Turing-Reconhecível, mas existem algumas linguagens que são Turing-Reconhecíveis, mas não Turing-Decidíveis.
- ▶ Qual a vantagem de Linguagens Turing-decidíveis?
- ▶ Se uma Linguagem é Turing-reconhecível, mas não Turing-decidível, como podemos verificar se a máquina que está demorando para dar o resultado ou se ela entrou em loop?

# Linguagens Turing-Decidíveis e Reconhecíveis

- ▶ Por enquanto, vamos nos focar em Linguagens Turing-decidíveis.

# Linguagens Turing-Decidíveis

## Exemplo

Verifique que a linguagem

$$L = \{w | w \in \{0, 1\}^*\}$$

é Turing-Decidível.

# Linguagens Turing-Decidíveis

## Exemplo

Verifique que a linguagem

$$L = \{w \mid w \in \{0, 1\}^* \wedge w \text{ é um número ímpar}\}$$

é Turing-Decidível.

# Linguagens Turing-Decidíveis

## Exemplo

Verifique que a linguagem

$$L = \{w \mid w \in \{0, 1\}^* \wedge w = w^R\}$$

é Turing-Decidível.

# Linguagens Turing-Decidíveis

## Exemplo

Verifique que a linguagem

$$L = \{w|x \in \{0,1\}^* \wedge w = x\#x\}$$

é Turing-Decidível.

# Linguagens Turing-Decidíveis

## Exemplo

Verifique que a linguagem

$$L = \{w | w = 0^{2^n}\}$$

é Turing-Decidível.