

Introdução à Teoria da Computação

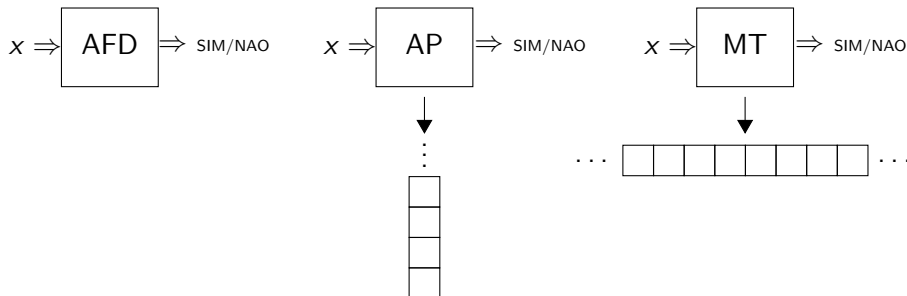
Máquinas de Turing

Professor Murilo V. G. da Silva

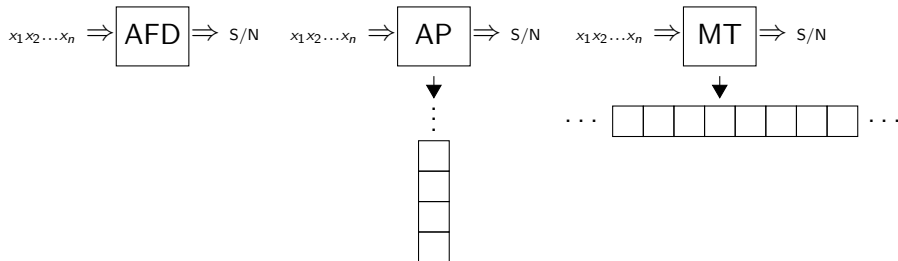
Departamento de Informática
Universidade Federal do Paraná

2026 / 2

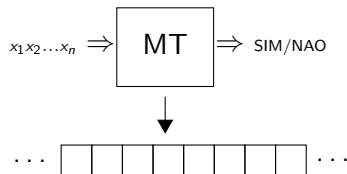
Máquinas de Turing



Máquinas de Turing



Máquinas de Turing: Detalhando o modelo

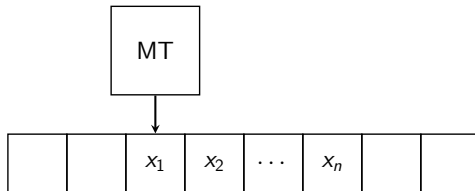


Máquinas de Turing: Detalhando o modelo

Mais precisamente, assumimos que a entrada $x_1x_2 \cdots x_n$ já está presente na fita:

Máquinas de Turing: Detalhando o modelo

Mais precisamente, assumimos que a entrada $x_1x_2 \cdots x_n$ já está presente na fita:



Máquinas de Turing - Definição

Máquinas de Turing - Definição

Seja $\mathcal{D} = \{L, R, S\}$.

Máquinas de Turing - Definição

Seja $\mathcal{D} = \{L, R, S\}$.

Uma **Máquina de Turing** é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

- Q é o conjunto finito de estados
- Σ é o alfabeto de entrada.
- Γ é o alfabeto da fita, tal que $\Sigma \subset \Gamma$.
-
- q_0 é o estado inicial
-
- $F \subseteq Q$ é o conjunto de estados finais.

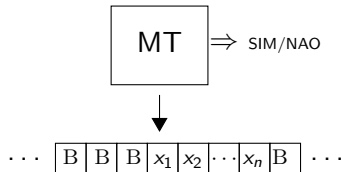
Máquinas de Turing - Definição

Seja $\mathcal{D} = \{L, R, S\}$.

Uma **Máquina de Turing** é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

- Q é o conjunto finito de estados
- Σ é o alfabeto de entrada.
- Γ é o alfabeto da fita, tal que $\Sigma \subset \Gamma$.
-
- q_0 é o estado inicial
- B é o símbolo especial chamado de *símbolo branco*, sendo $B \in \Gamma$
- $F \subseteq Q$ é o conjunto de estados finais.

Máquinas de Turing - Definição



Máquinas de Turing - Definição

Seja $\mathcal{D} = \{L, R, S\}$.

Uma **Máquina de Turing** é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

- Q é o conjunto finito de estados
- Σ é o alfabeto de entrada.
- Γ é o alfabeto da fita, tal que $\Sigma \subset \Gamma$.
-
- q_0 é o estado inicial
- B é o símbolo especial chamado de *símbolo branco*, sendo $B \in \Gamma$
- $F \subseteq Q$ é o conjunto de estados finais.

Máquinas de Turing - Definição

Seja $\mathcal{D} = \{L, R, S\}$.

Uma **Máquina de Turing** é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

- Q é o conjunto finito de estados
- Σ é o alfabeto de entrada.
- Γ é o alfabeto da fita, tal que $\Sigma \subset \Gamma$.
-
- q_0 é o estado inicial
- B é o símbolo especial chamado de *símbolo branco*, sendo $B \in \Gamma$
- $F \subseteq Q$ é o conjunto de estados finais.

Máquinas de Turing - Definição

Seja $\mathcal{D} = \{L, R, S\}$.

Uma **Máquina de Turing** é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

- Q é o conjunto finito de estados
- Σ é o alfabeto de entrada.
- Γ é o alfabeto da fita, tal que $\Sigma \subset \Gamma$.
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$
- q_0 é o estado inicial
- B é o símbolo especial chamado de *símbolo branco*.
- $F \subseteq Q$ é o conjunto de estados finais.

Máquinas de Turing - Definição

Seja $\mathcal{D} = \{L, R, S\}$.

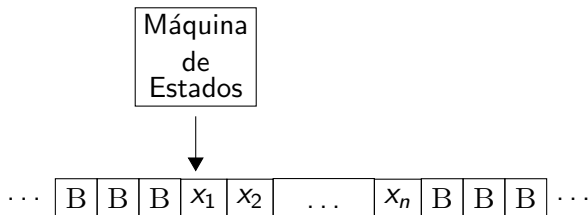
Uma **Máquina de Turing** é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

- Q é o conjunto finito de estados
- Σ é o alfabeto de entrada.
- Γ é o alfabeto da fita, tal que $\Sigma \subset \Gamma$.
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$ é uma *função parcial*
- q_0 é o estado inicial
- B é o símbolo especial chamado de *símbolo branco*.
- $F \subseteq Q$ é o conjunto de estados finais.

Máquinas de Turing - Semântica

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

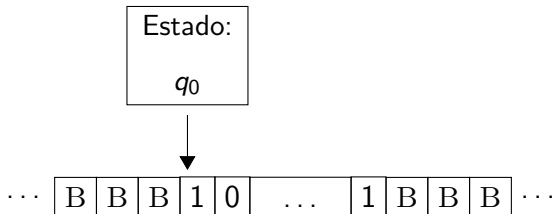
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



Máquinas de Turing - Semântica

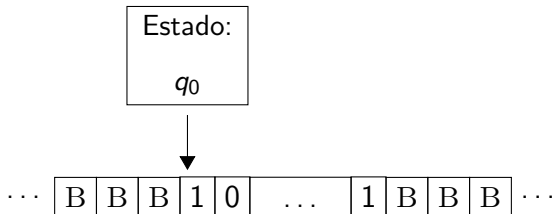
Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

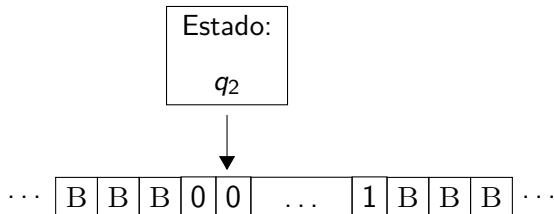
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



Por exemplo, digamos que : $\delta(q_0, 1) = (q_2, 0, R)$

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.

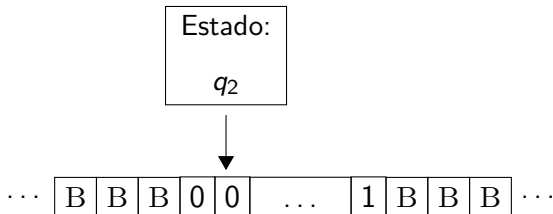


Por exemplo, digamos que : $\delta(q_0, 1) = (q_2, 0, R)$

Máquinas de Turing - Semântica

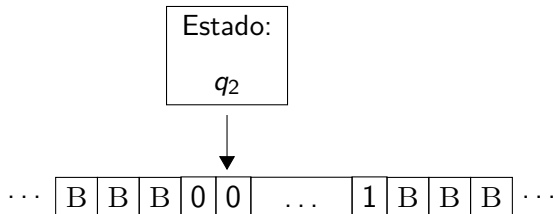
Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

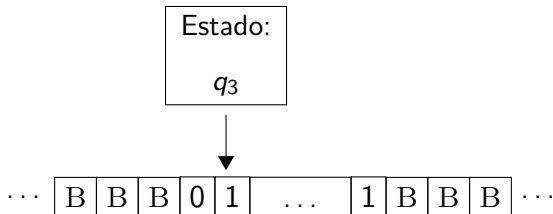
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



Digamos agora que $\delta(q_2, 0) = (q_3, 1, S)$

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.

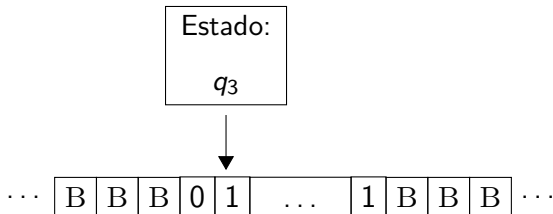


Digamos agora que $\delta(q_2, 0) = (q_3, 1, S)$

Máquinas de Turing - Semântica

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

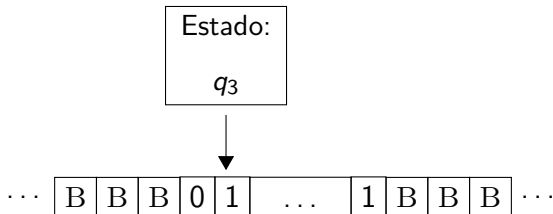
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



Máquinas de Turing - Semântica

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

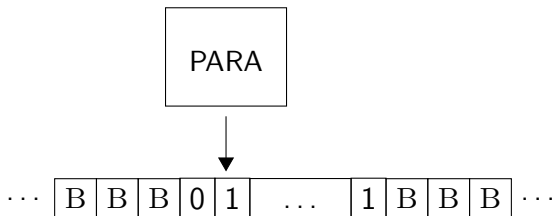
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



Agora, digamos que $\delta(q_3, 1)$ não definido

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.

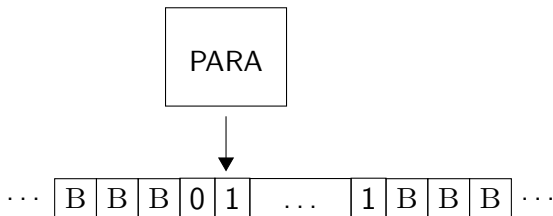


Agora, digamos que $\delta(q_3, 1)$ não definido

Máquinas de Turing - Semântica

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



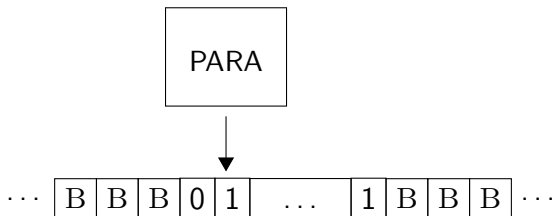
Agora, digamos que $\delta(q_3, 1)$ não definido

Parar é um passo determinístico

Máquinas de Turing - Semântica

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

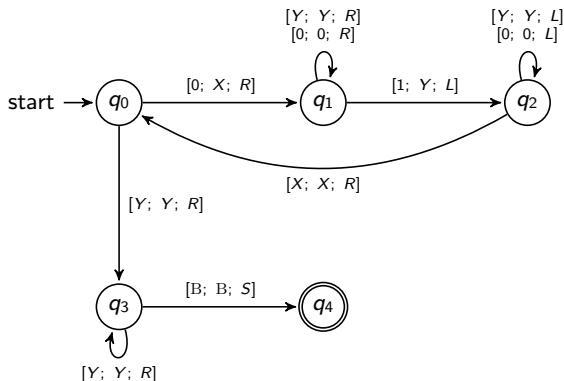
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times \mathcal{D}$.



Agora, digamos que $\delta(q_3, 1)$ não definido

Parar é um passo determinístico $\Rightarrow$ Aceita ou rejeita dependendo do estado que parou

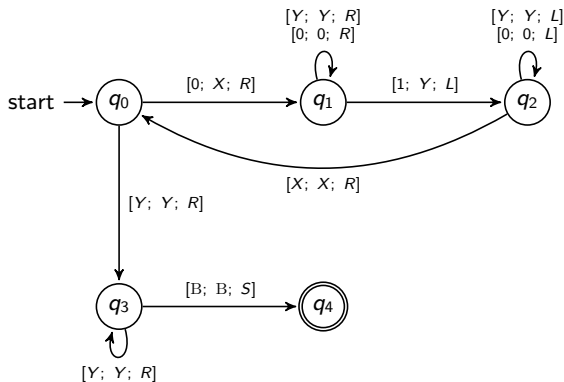
Máquinas de Turing - Um exemplo



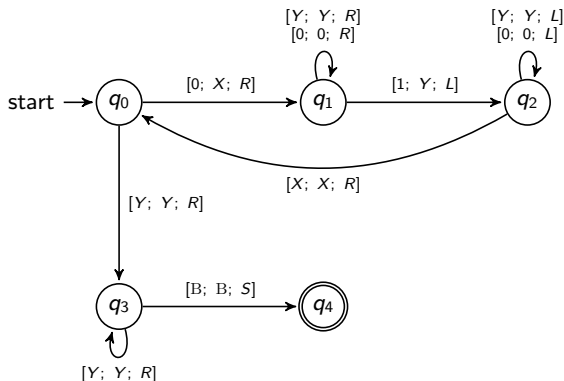
$M_{01} = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ tal que

- $Q = \{q_0, \dots, q_4\}$ $\Sigma = \{0, 1\}$ $\Gamma = \{0, 1, X, Y, B\}$ $F = \{q_4\}$
- $\delta(q_0, 0) = (q_1, X, R)$ $\delta(q_0, Y) = (q_3, Y, R)$ $\delta(q_1, 0) = (q_1, 0, R)$ $\delta(q_1, Y) = (q_1, Y, R)$
 $\delta(q_1, 1) = (q_2, Y, 0)$ $\delta(q_2, 0) = (q_2, 0, L)$ $\delta(q_2, Y) = (q_2, Y, L)$ $\delta(q_3, Y) = (q_3, Y, R)$ $\delta(q_3, B) = (q_4, B, S)$

Máquinas de Turing - Configurações

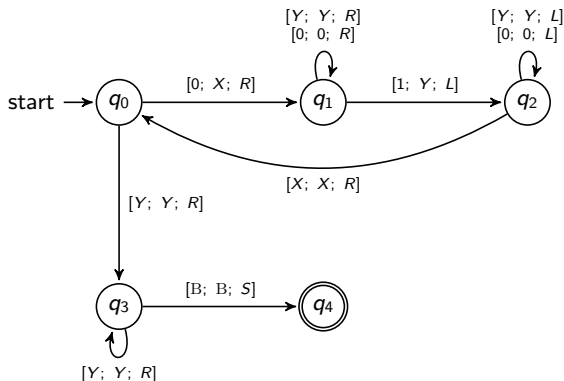


Máquinas de Turing - Configurações



A cada passo: A MT se encontra em uma **configuração**:

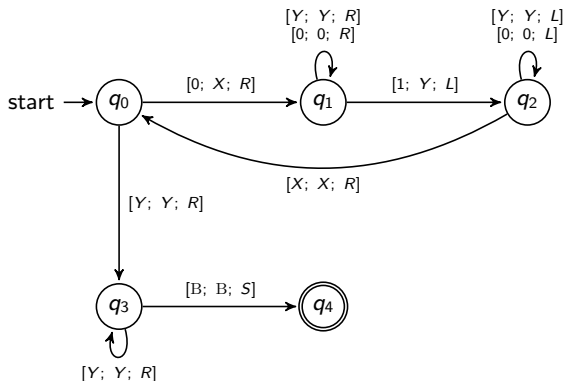
Máquinas de Turing - Configurações



A cada passo: A MT se encontra em uma **configuração**:

- O que define a configuração?

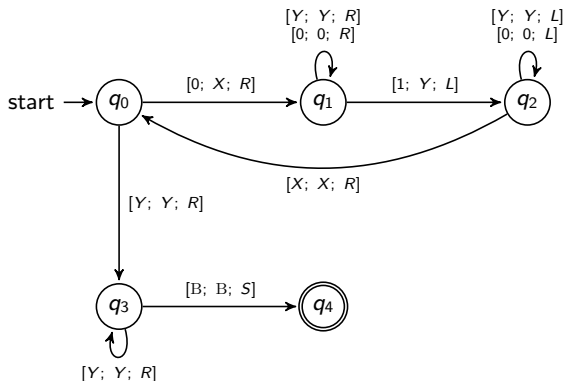
Máquinas de Turing - Configurações



A cada passo: A MT se encontra em uma **configuração**:

- O que define a configuração? O estado q ,

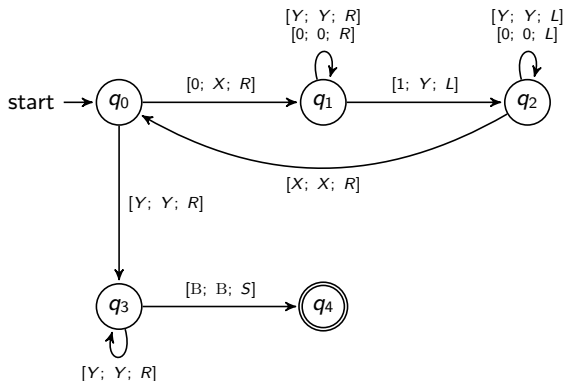
Máquinas de Turing - Configurações



A cada passo: A MT se encontra em uma **configuração**:

- O que define a configuração? O estado q , a string w na fita e

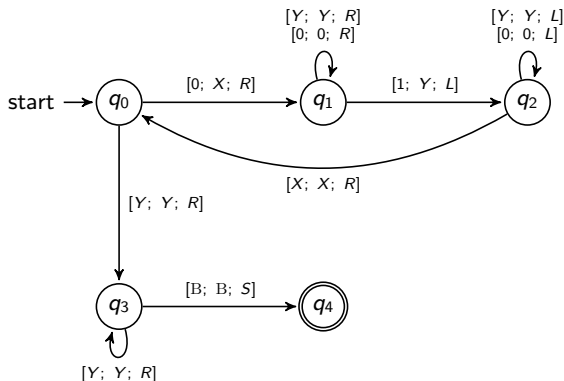
Máquinas de Turing - Configurações



A cada passo: A MT se encontra em uma **configuração**:

- O que define a configuração? O **estado** q , a **string** w na fita e **posição** do cabeçote

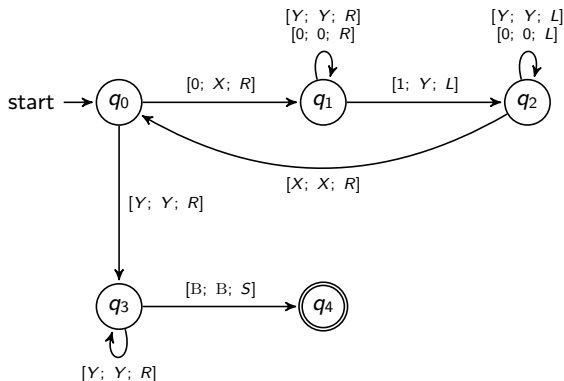
Máquinas de Turing - Configurações



A cada passo: A MT se encontra em uma **configuração**:

- O que define a configuração? O estado q , a string w na fita e posição do cabeçote
- Alternativamente, a tripla (estado q , string α à esquerda do cabeçote, string restante β)

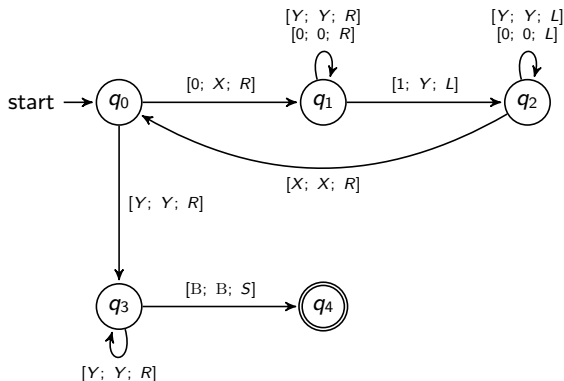
Máquinas de Turing - Configurações



A cada passo: A MT se encontra em uma **configuração**:

- O que define a configuração? O estado q , a string w na fita e posição do cabeçote
- Alternativamente, a tripla (estado q , string α à esquerda do cabeçote, string restante β) ou seja, $w = \alpha\beta$

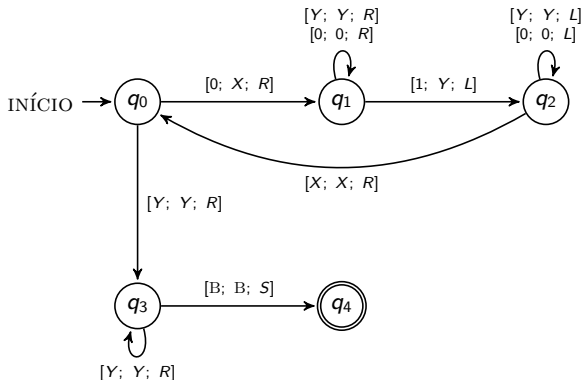
Máquinas de Turing - Configurações



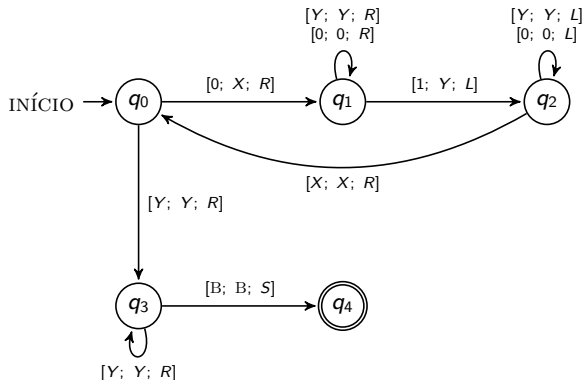
A cada passo: A MT se encontra em uma **configuração**:

- O que define a configuração? O **estado** q , a **string** w na fita e **posição do cabeçote**
- Alternativamente, a tripla (**estado** q , **string** α à esquerda do cabeçote, **string restante** β) ou seja, $w = \alpha\beta$ e o cabeçote está sobre o primeiro símbolo de β

Máquinas de Turing - Configurações

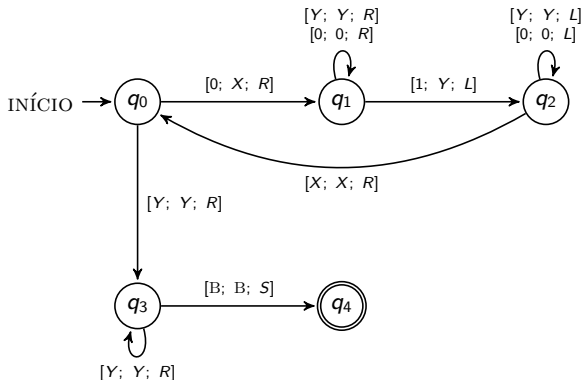


Máquinas de Turing - Configurações



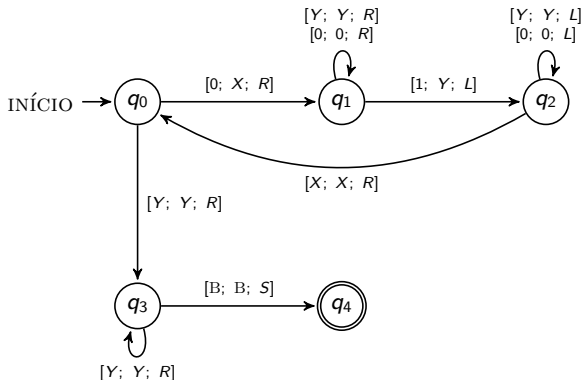
Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) ,

Máquinas de Turing - Configurações



Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

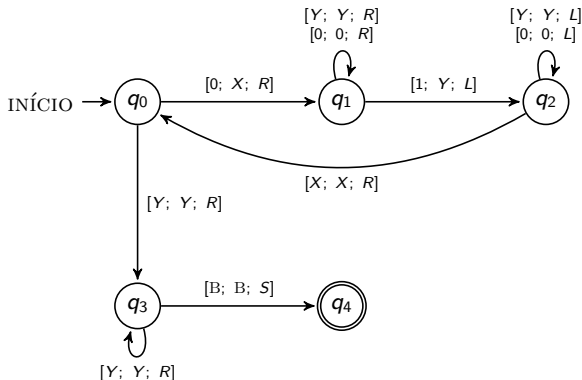
Máquinas de Turing - Configurações



Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

- Suponha que a MT esteja na configuração ($X00, q_1, 111$)

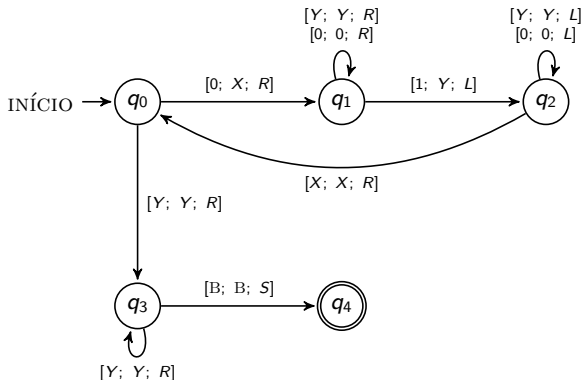
Máquinas de Turing - Configurações



Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

- Suponha que a MT esteja na configuração (~~X~~00, q_1 , 111)
- No próximo passo, qual será a configuração?

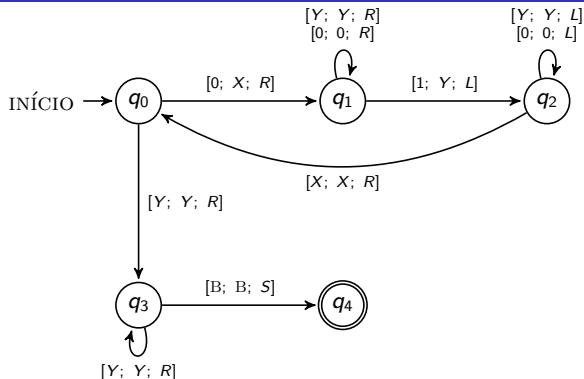
Máquinas de Turing - Configurações



Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

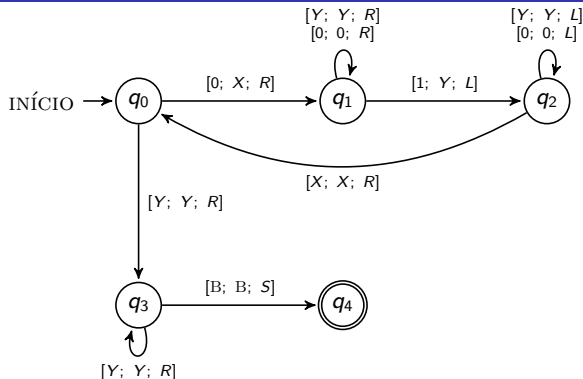
- Suponha que a MT esteja na configuração $(X00, q_1, 111)$
- No próximo passo, qual será a configuração? Resposta: $(X0, q_2, 0Y11)$

Máquinas de Turing - Configurações



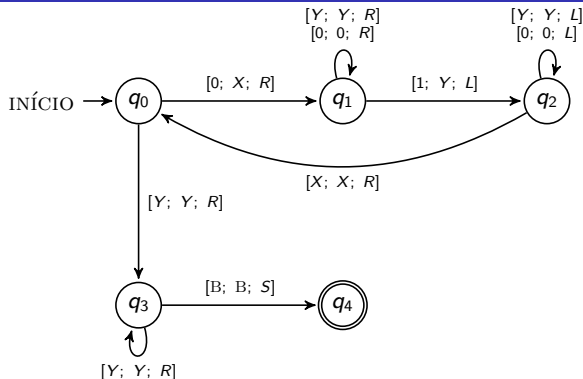
Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) ,

Máquinas de Turing - Configurações



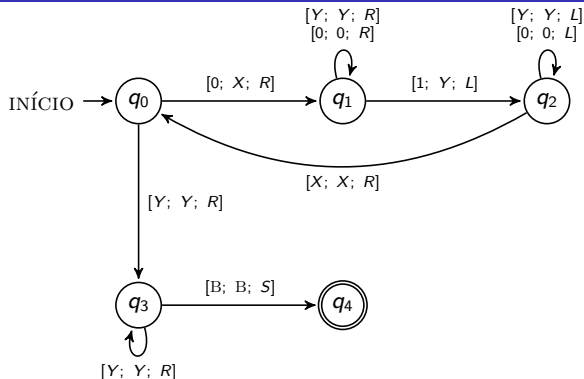
Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Máquinas de Turing - Configurações



Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

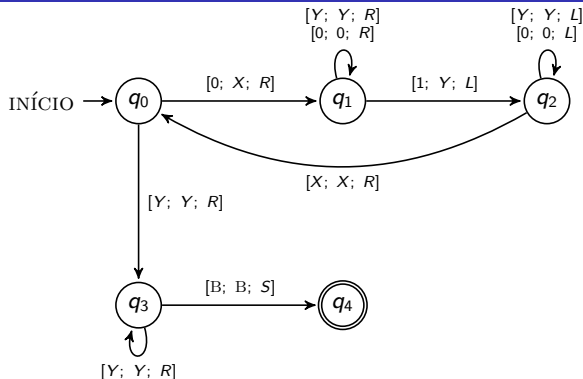
Máquinas de Turing - Configurações



Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

- Sequência de passos computacionais:

Máquinas de Turing - Configurações

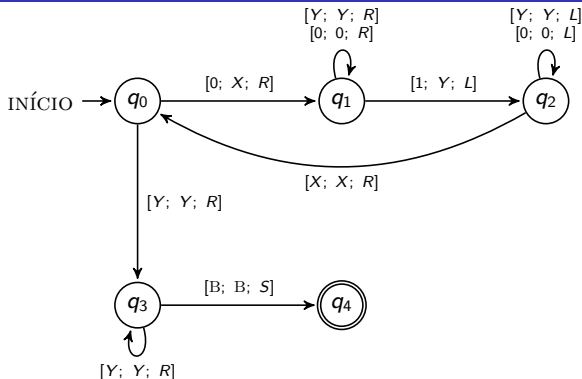


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

- Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash$

Máquinas de Turing - Configurações

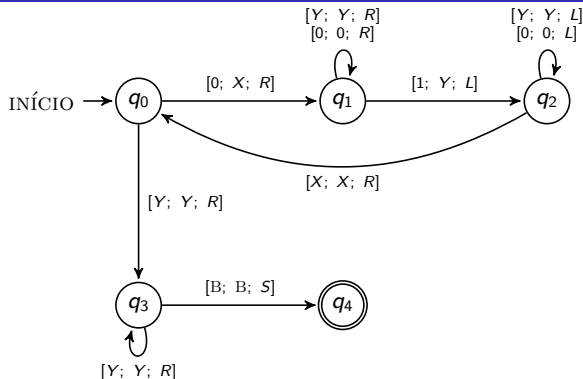


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash$

Máquinas de Turing - Configurações

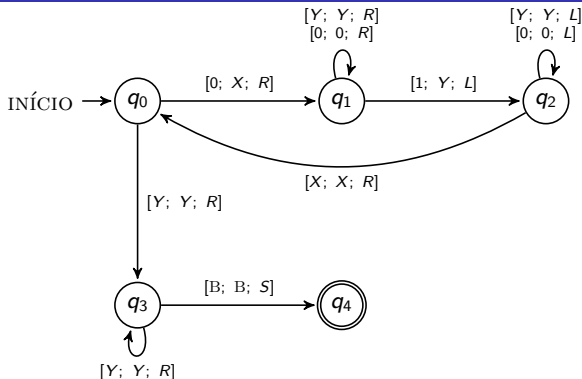


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash$

Máquinas de Turing - Configurações

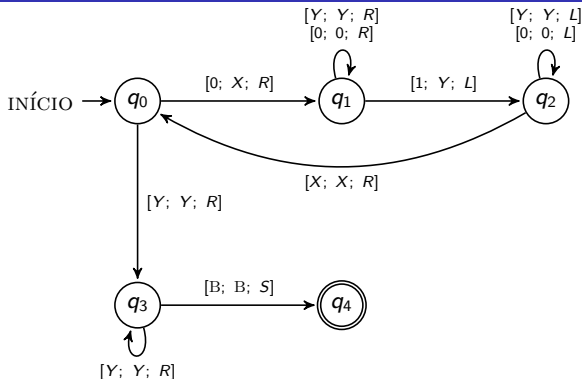


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash$

Máquinas de Turing - Configurações

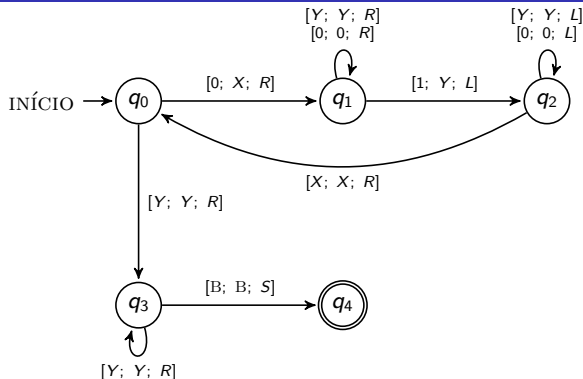


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$

Máquinas de Turing - Configurações

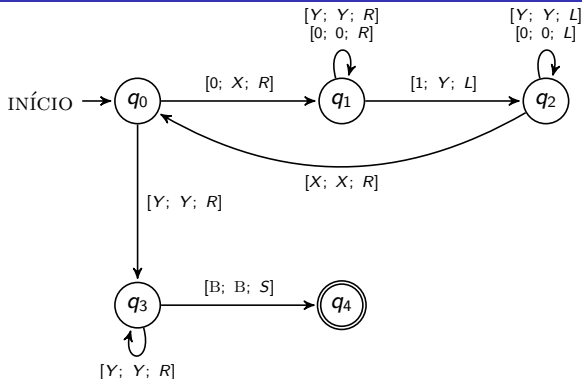


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash$

Máquinas de Turing - Configurações

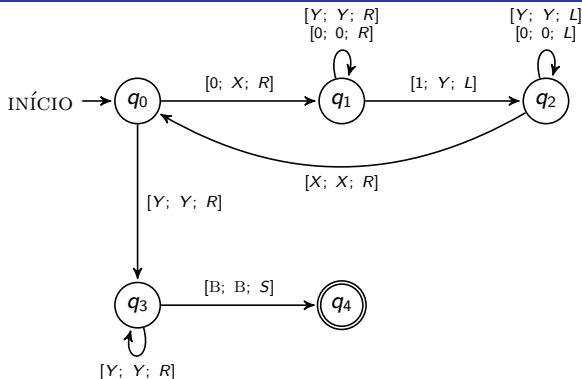


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash$

Máquinas de Turing - Configurações

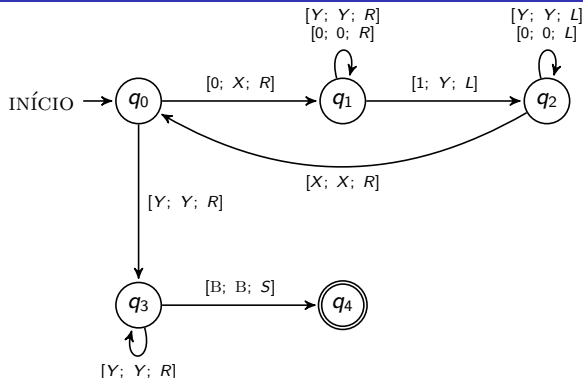


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash$

Máquinas de Turing - Configurações

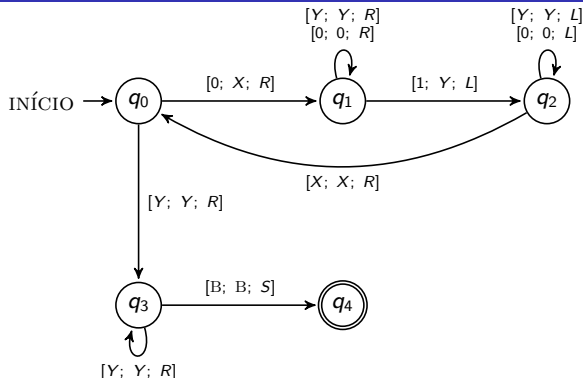


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash$

Máquinas de Turing - Configurações

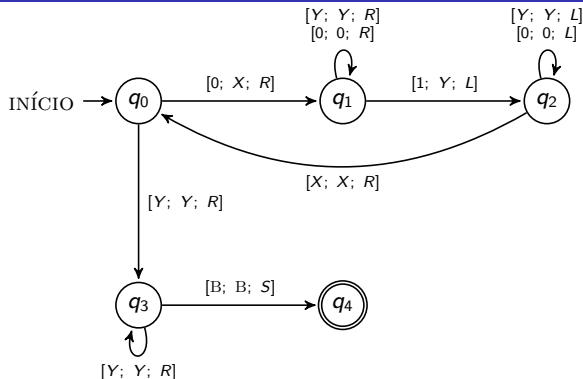


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$

Máquinas de Turing - Configurações

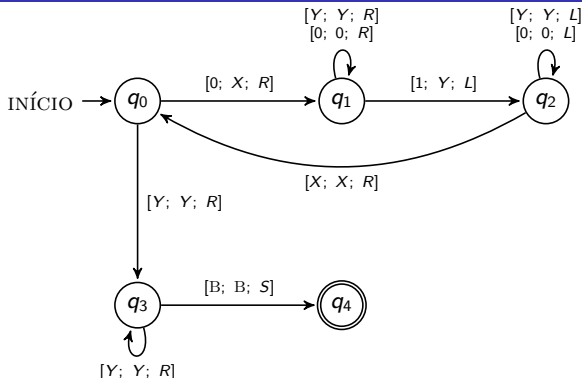


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash$

Máquinas de Turing - Configurações

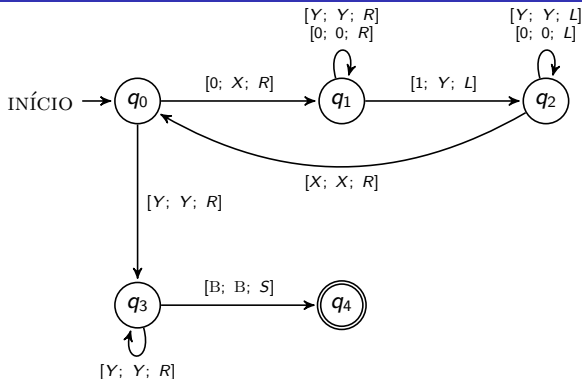


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash$

Máquinas de Turing - Configurações

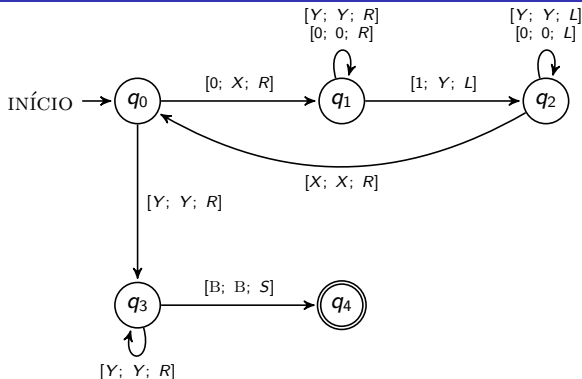


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash$

Máquinas de Turing - Configurações

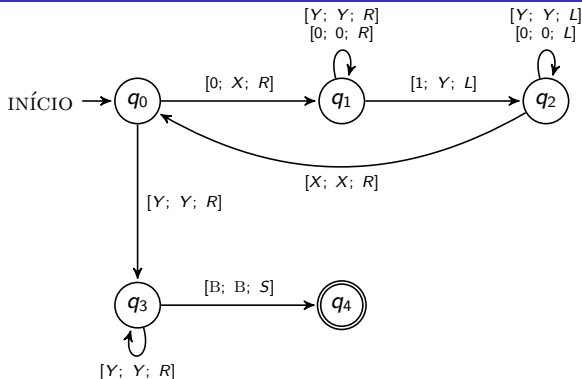


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash$

Máquinas de Turing - Configurações

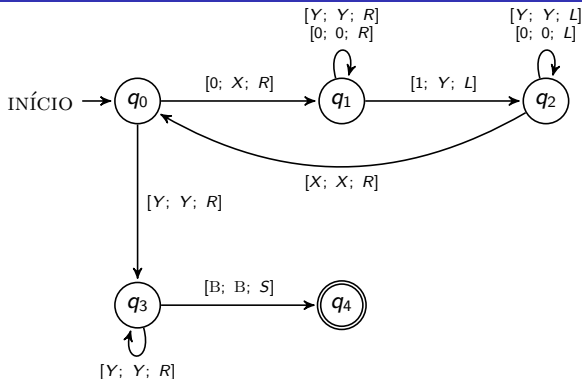


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$

Máquinas de Turing - Configurações

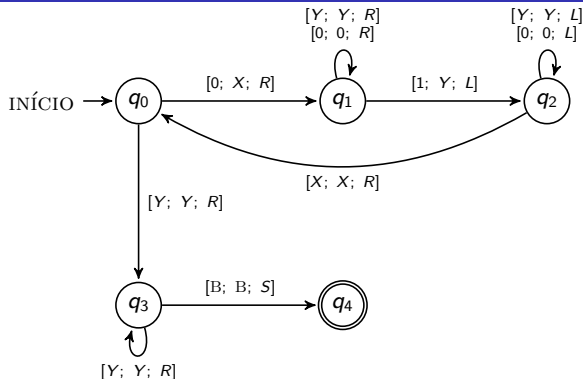


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash$

Máquinas de Turing - Configurações

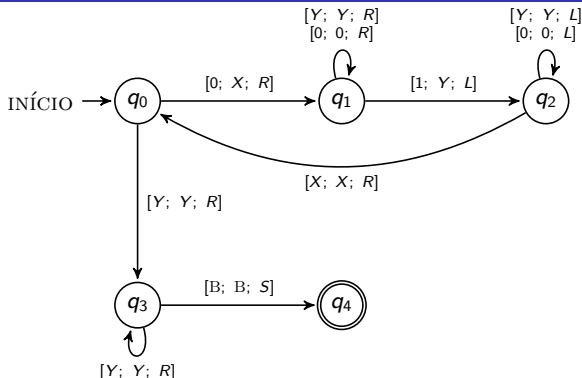


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

• Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXXY, q_1, Y1) \vdash$

Máquinas de Turing - Configurações

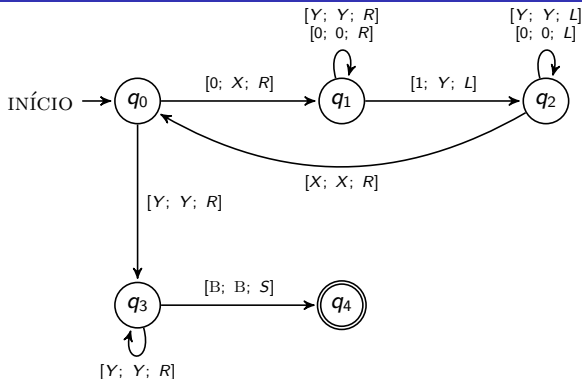


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXX, q_1, Y1) \vdash (XXXYY, q_1, 1) \vdash$

Máquinas de Turing - Configurações

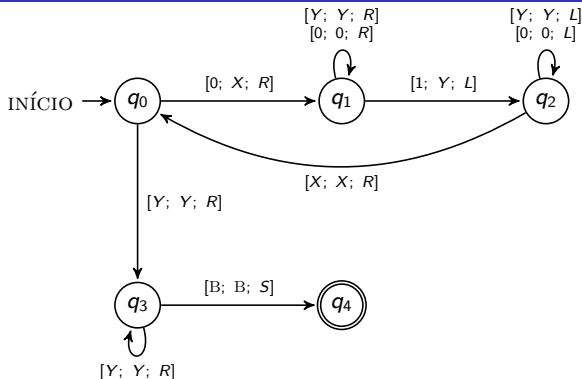


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_1, Y1) \vdash$

Máquinas de Turing - Configurações

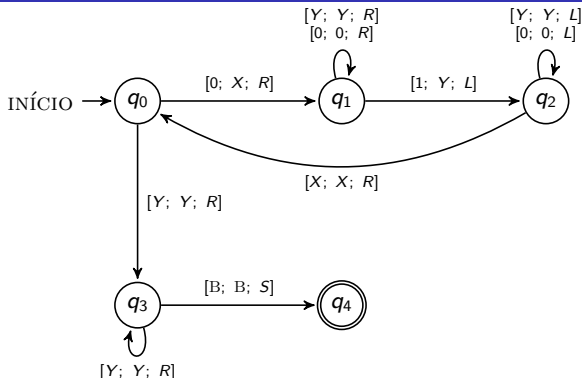


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_1, 1) \vdash (XXX, q_2, YY) \vdash (XXX, q_2, YYY) \vdash$

Máquinas de Turing - Configurações

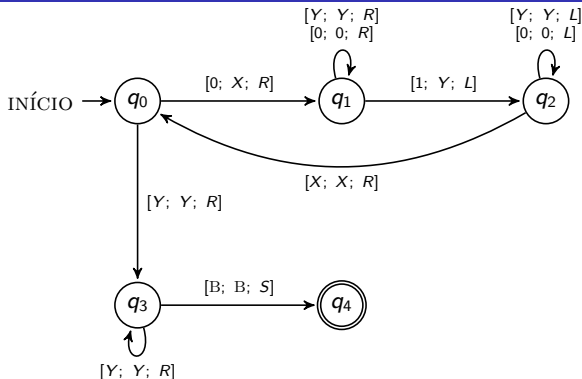


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_2, YY) \vdash (XXX, q_2, YYY) \vdash$
 $(XX, q_2, XYYY) \vdash$

Máquinas de Turing - Configurações

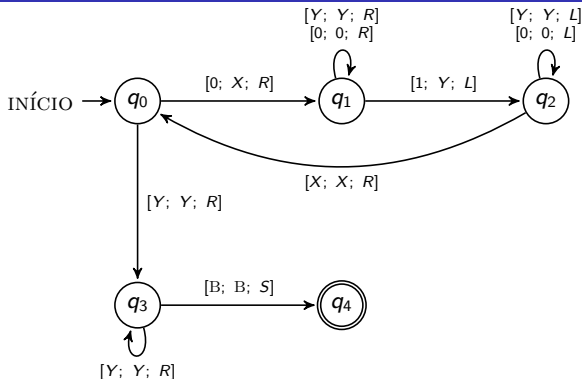


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_1, 1) \vdash (XXX, q_2, YY) \vdash (XXX, q_2, YYY) \vdash$
 $(XX, q_2, XYYY) \vdash (XXX, q_0, YYY)$

Máquinas de Turing - Configurações

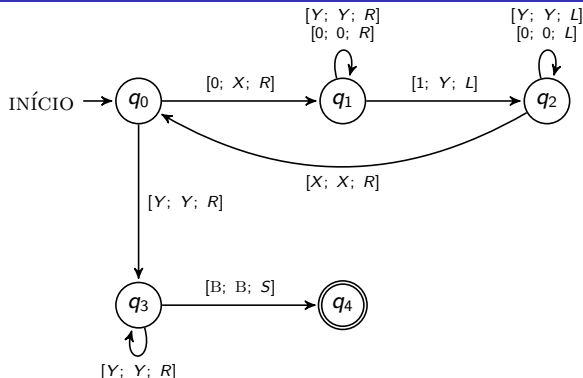


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXX, q_1, YY1) \vdash (XXX, q_1, YY1) \vdash (XXX, q_1, YY1) \vdash (XXX, q_1, YY1) \vdash$
 $(XX, q_2, XYYY) \vdash (XXX, q_0, YYY) \vdash (XXX, q_2, YYY) \vdash (XXX, q_2, YYY) \vdash (XXX, q_2, YYY) \vdash$

Máquinas de Turing - Configurações

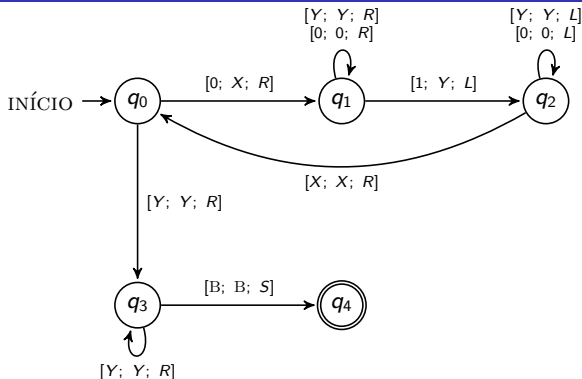


Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXX, q_1, YY1) \vdash (XXX, q_1, YY1) \vdash (XXX, q_1, YY1) \vdash (XXX, q_1, YY1) \vdash$
 $(XX, q_2, XYYY) \vdash (XXX, q_0, YYY) \vdash (XXX, q_3, YY) \vdash (XXX, q_3, Y) \vdash$

Máquinas de Turing - Configurações

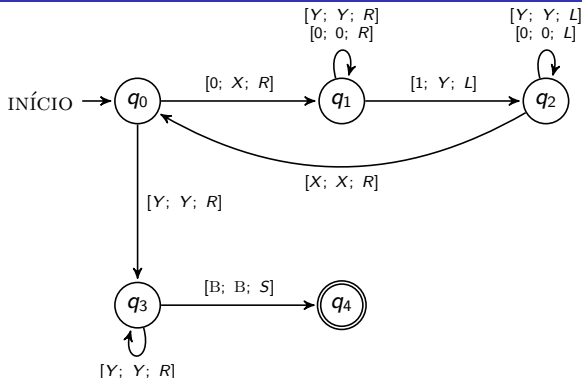


Def.: Uma **configuração** de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

- Sequência de passos computacionais:

$$\begin{aligned} &(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash \\ &(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash \\ &(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash \\ &(XXX, q_1, YY1) \vdash (XXXY, q_1, Y1) \vdash (XXXY, q_1, 1) \vdash (XXXY, q_2, YY) \vdash (XXX, q_2, YYY) \vdash \\ &(XX, q_2, XYYY) \vdash (XXX, q_0, YYY) \vdash (XXXY, q_3, YY) \vdash (XXXY, q_3, Y) \vdash \\ &(XXXXYY, q_3, \epsilon) \vdash \end{aligned}$$

Máquinas de Turing - Configurações



Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

$(\epsilon, q_0, 000111) \vdash (X, q_1, 00111) \vdash (X0, q_1, 0111) \vdash (X00, q_1, 111) \vdash (X0, q_2, 0Y11) \vdash$
 $(X, q_2, 00Y11) \vdash (\epsilon, q_2, X00Y11) \vdash (X, q_0, 00Y11) \vdash (XX, q_1, 0Y11) \vdash (XX0, q_1, Y11) \vdash$
 $(XX0Y, q_1, 11) \vdash (XX0, q_2, YY1) \vdash (XX, q_2, 0YY1) \vdash (X, q_2, X0YY1) \vdash (XX, q_0, 0YY1) \vdash$
 $(XXX, q_1, YY1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_1, Y1) \vdash (XXX, q_2, YY) \vdash (XXX, q_2, YYY) \vdash$
 $(XX, q_2, XYYY) \vdash (XXX, q_0, YYY) \vdash (XXX, q_3, YY) \vdash (XXX, q_3, Y) \vdash$
 $(XXXXYY, q_3, \epsilon) \vdash (XXXXYY, q_4, \epsilon).$

Exercícios: MTs, aceitação de strings, loop infinito

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercícios: MTs, aceitação de strings, loop infinito

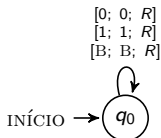
Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

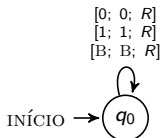
Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?

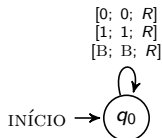


Exercício 3: Apresente a definição formal da MT do Exercício 2.

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



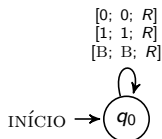
Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.:

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



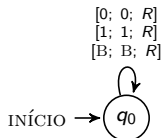
Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.: $M_{\emptyset} = (\{q_0\},$

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



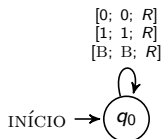
Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.: $M_{\emptyset} = (\{q_0\}, \{0, 1\},$

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



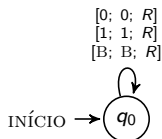
Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.: $M_{\emptyset} = (\{q_0\}, \{0, 1\}, \{0, 1, B\},$

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



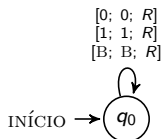
Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.: $M_{\emptyset} = (\{q_0\}, \{0, 1\}, \{0, 1, B\}, \delta,$

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



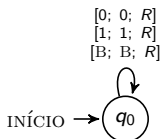
Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.: $M_{\emptyset} = (\{q_0\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0,$

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



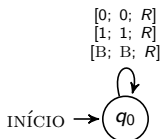
Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.: $M_{\emptyset} = (\{q_0\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B,$

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



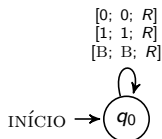
Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.: $M_{\emptyset} = (\{q_0\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \emptyset)$

Exercícios: MTs, aceitação de strings, loop infinito

Exercício 1: (para casa, exercício resolvido no livro) Forneça uma MT que aceite uma string de entrada w se e somente se w pertence a linguagem $\{a^n b^n c^n : n \geq 1\}$

Exercício 2: Qual é o conjunto de strings aceitas pela MT abaixo?



Exercício 3: Apresente a definição formal da MT do Exercício 2.

Resp.: $M_{\emptyset} = (\{q_0\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \emptyset)$, tal que $\forall S \in \{0, 1, B\}, \delta(q_0, S) = (q_0, S, R)$.

Exercícios: MTs, aceitação de strings, loop infinito

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_\emptyset$ (slide anterior) fica em *loop infinito* para qualquer entrada possível

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares.

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop**

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop** (e.g., MT M_{01}).

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop** (e.g., MT M_{01}).
Mostre que neste caso sempre existe uma outra MT M' tal que:

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop** (e.g., MT M_{01}). Mostre que neste caso sempre existe uma outra MT M' tal que:

→ Se M **para a execução** em estado de aceitação para uma string x , então M' **deve ficar em loop** para a mesma string x

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop** (e.g., MT M_{01}). Mostre que neste caso sempre existe uma outra MT M' tal que:

- Se M para a execução em estado de aceitação para uma string x , então M' deve ficar em loop para a mesma string x
- Se M para a execução em estado de não aceitação para uma string x , então M' também deve parar a execução em estado de não aceitação para x .

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop** (e.g., MT M_{01}). Mostre que neste caso sempre existe uma outra MT M' tal que:

- Se M para a execução em estado de aceitação para uma string x , então M' deve ficar em loop para a mesma string x
- Se M para a execução em estado de não aceitação para uma string x , então M' também deve parar a execução em estado de não aceitação para x .

Exercício 5: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop**

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_\emptyset$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop** (e.g., MT M_{01}). Mostre que neste caso sempre existe uma outra MT M' tal que:

- Se M para a execução em estado de aceitação para uma string x , então M' deve ficar em loop para a mesma string x
- Se M para a execução em estado de não aceitação para uma string x , então M' também deve parar a execução em estado de não aceitação para x .

Exercício 5: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop**. Mostre que neste caso sempre existe uma outra MT M' tal que:

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop** (e.g., MT M_{01}). Mostre que neste caso sempre existe uma outra MT M' tal que:

- Se M para a execução em estado de aceitação para uma string x , então M' deve ficar em loop para a mesma string x
- Se M para a execução em estado de não aceitação para uma string x , então M' também deve parar a execução em estado de não aceitação para x .

Exercício 5: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop**. Mostre que neste caso sempre existe uma outra MT M' tal que:

- Se M para a execução em estado de aceitação para uma string x , então M' também deve parar em estado de aceitação para x

Exercícios: MTs, aceitação de strings, loop infinito

Observe o seguinte:

- A MT $M_{\emptyset}$ (slide anterior) fica em *loop infinito* para qualquer entrada possível
- A MT M_{01} (vista anteriormente) não fica em *loop infinito* para nenhuma entrada
- Estas MTs são casos particulares. Em geral, MTs podem parar para algumas strings e ficar em loop para outras.

Exercício 4: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop** (e.g., MT M_{01}). Mostre que neste caso sempre existe uma outra MT M' tal que:

- Se M para a execução em estado de aceitação para uma string x , então M' deve ficar em loop para a mesma string x
- Se M para a execução em estado de não aceitação para uma string x , então M' também deve parar a execução em estado de não aceitação para x .

Exercício 5: Seja uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ que **nunca nunca fica loop**. Mostre que neste caso sempre existe uma outra MT M' tal que:

- Se M para a execução em estado de aceitação para uma string x , então M' também deve parar em estado de aceitação para x
- Se M para a execução em estado de não aceitação para uma string x , então M' deve ficar em loop para a mesma string x

Passos computacionais: Definição formal

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1$$

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e
- $i = n$ e $Y = B$.

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e
- $i = n$ e $Y = B$.

Nestes casos temos o seguinte:

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e
- $i = n$ e $Y = B$.

Nestes casos temos o seguinte:

(1) Se $i = 1$, então $(\epsilon, q, X_1 \dots X_n) \vdash_M^1 (\epsilon, p, B Y X_2 \dots X_n)$

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e
- $i = n$ e $Y = B$.

Nestes casos temos o seguinte:

- (1) Se $i = 1$, então $(\epsilon, q, X_1 \dots X_n) \vdash_M^1 (\epsilon, p, B Y X_2 \dots X_n)$
- (2) Se $i = n$ e $Y = B$, então $(X_1 X_2 \dots X_{n-1}, q, X_n) \vdash_M^1 (X_1 X_2 \dots X_{n-2}, p, X_{n-1})$

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e
- $i = n$ e $Y = B$.

Nestes casos temos o seguinte:

- (1) Se $i = 1$, então $(\epsilon, q, X_1 \dots X_n) \vdash_M^1 (\epsilon, p, B Y X_2 \dots X_n)$
- (2) Se $i = n$ e $Y = B$, então $(X_1 X_2 \dots X_{n-1}, q, X_n) \vdash_M^1 (X_1 X_2 \dots X_{n-2}, p, X_{n-1})$

Definições semelhantes: **passo computacional à direita**

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e
- $i = n$ e $Y = B$.

Nestes casos temos o seguinte:

- (1) Se $i = 1$, então $(\epsilon, q, X_1 \dots X_n) \vdash_M^1 (\epsilon, p, B Y X_2 \dots X_n)$
- (2) Se $i = n$ e $Y = B$, então $(X_1 X_2 \dots X_{n-1}, q, X_n) \vdash_M^1 (X_1 X_2 \dots X_{n-2}, p, X_{n-1})$

Definições semelhantes: **passo computacional à direita** e **passo computacional estático**

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e
- $i = n$ e $Y = B$.

Nestes casos temos o seguinte:

- (1) Se $i = 1$, então $(\epsilon, q, X_1 \dots X_n) \vdash_M^1 (\epsilon, p, B Y X_2 \dots X_n)$
- (2) Se $i = n$ e $Y = B$, então $(X_1 X_2 \dots X_{n-1}, q, X_n) \vdash_M^1 (X_1 X_2 \dots X_{n-2}, p, X_{n-1})$

Definições semelhantes: **passo computacional à direita** e **passo computacional estático**

Notação: $\vdash_M^r$ e $\vdash_M^s$

Passos computacionais: Definição formal

- Precisamos definir separadamente passos à esquerda, à direita e estático

Def.: Passo computacional à esquerda

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ uma MT e $(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n)$ uma configuração da máquina M para uma string $x = X_1 \dots X_n$.

Se $\delta(q, X_i) = (p, Y, L)$, então escrevemos

$$(X_1 X_2 \dots X_{i-1}, q, X_i X_{i+1} \dots X_n) \vdash_M^1 (X_1 X_2 \dots X_{i-2}, p, X_{i-1} Y X_{i+1} \dots X_n)$$

Exceto nos casos

- $i = 1$ e
- $i = n$ e $Y = B$.

Nestes casos temos o seguinte:

- (1) Se $i = 1$, então $(\epsilon, q, X_1 \dots X_n) \vdash_M^1 (\epsilon, p, B Y X_2 \dots X_n)$
- (2) Se $i = n$ e $Y = B$, então $(X_1 X_2 \dots X_{n-1}, q, X_n) \vdash_M^1 (X_1 X_2 \dots X_{n-2}, p, X_{n-1})$

Definições semelhantes: **passo computacional à direita** e **passo computacional estático**

Notação: $\vdash_M^r$ e $\vdash_M^s$ **(exercício do livro)**

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Base: $C_i \vdash_M^* C_j$, caso $i = j$

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Base: $C_i \vdash_M^* C_j$, caso $i = j$

Indução: $C_i \vdash_M^* C_j$ se $\exists C_k$ tal que $C_i \vdash_M C_k$ e $C_k \vdash_M^* C_j$.

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Base: $C_i \vdash_M^* C_j$, caso $i = j$

Indução: $C_i \vdash_M^* C_j$ se $\exists C_k$ tal que $C_i \vdash_M C_k$ e $C_k \vdash_M^* C_j$.

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Base: $C_i \vdash_M^* C_j$, caso $i = j$

Indução: $C_i \vdash_M^* C_j$ se $\exists C_k$ tal que $C_i \vdash_M C_k$ e $C_k \vdash_M^* C_j$.

Configurações finais de uma MT:

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Base: $C_i \vdash_M^* C_j$, caso $i = j$

Indução: $C_i \vdash_M^* C_j$ se $\exists C_k$ tal que $C_i \vdash_M C_k$ e $C_k \vdash_M^* C_j$.

Configurações finais de uma MT: Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Base: $C_i \vdash_M^* C_j$, caso $i = j$

Indução: $C_i \vdash_M^* C_j$ se $\exists C_k$ tal que $C_i \vdash_M C_k$ e $C_k \vdash_M^* C_j$.

Configurações finais de uma MT: Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

- Se $q_F \in F$,

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Base: $C_i \vdash_M^* C_j$, caso $i = j$

Indução: $C_i \vdash_M^* C_j$ se $\exists C_k$ tal que $C_i \vdash_M C_k$ e $C_k \vdash_M^* C_j$.

Configurações finais de uma MT: Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

- Se $q_F \in F$, então (α, q_F, β) é chamada de configuração final de M

Passos computacionais: Definição formal

Def. Passo computacional ($\vdash_M$)

Dada uma MT M , um *passo computacional* de M , denotado $\vdash_M$, se refere a qualquer um dos três casos de passos computacionais $\vdash_M^l$, $\vdash_M^r$ ou $\vdash_M^s$.

Múltiplos passos computacionais ($\vdash_M^*$)

Dada uma MT M , definimos $C_i \vdash_M^* C_j$ recursivamente:

Base: $C_i \vdash_M^* C_j$, caso $i = j$

Indução: $C_i \vdash_M^* C_j$ se $\exists C_k$ tal que $C_i \vdash_M C_k$ e $C_k \vdash_M^* C_j$.

Configurações finais de uma MT: Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

- Se $q_F \in F$, então (α, q_F, β) é chamada de configuração final de M
 $\alpha, \beta \in \Gamma^*$ quaisquer.

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$,

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma *configuração final* de M .

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma *configuração final* de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma *configuração final* de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é *aceita* por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$,

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma *configuração final* de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma *configuração final* de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é *aceita* por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma *configuração final* de M ;

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma *configuração final* de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma *configuração final* de M ;
- a máquina para ao atingir a configuração C_N .

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma **configuração final** de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma configuração final de M ;
- a máquina para ao atingir a configuração C_N .

Importante: Uma string w que não é aceita por M , não necessariamente é rejeitada!

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma **configuração final** de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma configuração final de M ;
- a máquina para ao atingir a configuração C_N .

Importante: Uma string w que não é aceita por M , não necessariamente é rejeitada!

M pode ficar em *loop infinito* com w

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma **configuração final** de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma configuração final de M ;
- a máquina para ao atingir a configuração C_N .

Importante: Uma string w que não é aceita por M , não necessariamente é rejeitada!
 M pode ficar em *loop infinito* com w (i.e., não respeita nenhuma das condições acima)

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma **configuração final** de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma configuração final de M ;
- a máquina para ao atingir a configuração C_N .

Importante: Uma string w que não é aceita por M , não necessariamente é rejeitada!
 M pode ficar em *loop infinito* com w (i.e., não respeita nenhuma das condições acima)

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma **configuração final** de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma configuração final de M ;
- a máquina para ao atingir a configuração C_N .

Importante: Uma string w que não é aceita por M , não necessariamente é rejeitada!

M pode ficar em *loop infinito* com w (i.e., não respeita nenhuma das condições acima)

- **Relembrando que:** Uma certa MT pode:

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma **configuração final** de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma configuração final de M ;
- a máquina para ao atingir a configuração C_N .

Importante: Uma string w que não é aceita por M , não necessariamente é rejeitada!

M pode ficar em *loop infinito* com w (i.e., não respeita nenhuma das condições acima)

- **Relembrando que:** Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes”

Máquinas de Turing: Aceitação e Rejeição de strings

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ e $w \in \Sigma^*$

Strings aceitas: Dizemos que w é *aceita* por M se $(\epsilon, q_0, w) \vdash_M^* C_F$, tal que,

- C_F é uma **configuração final** de M .

Dado w , caso não se aplique a def. acima, dizemos que w não é aceita por M .

Strings rejeitadas Dizemos que w é *rejeitada* por M se $(\epsilon, q_0, w) \vdash_M^* C_N$, tal que,

- C_N não é uma configuração final de M ;
- a máquina para ao atingir a configuração C_N .

Importante: Uma string w que não é aceita por M , não necessariamente é rejeitada!

M pode ficar em *loop infinito* com w (i.e., não respeita nenhuma das condições acima)

- **Relembrando que:** Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes” i.e.,
$$C = \Sigma^* \setminus (A \cup B)$$

Máquinas de Turing e Linguagens

- Relembrando que: Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes” i.e.,
 $C = \Sigma^* \setminus (A \cup B)$

Máquinas de Turing e Linguagens

- **Relembrando que:** Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes” i.e.,
 $C = \Sigma^* \setminus (A \cup B)$

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Máquinas de Turing e Linguagens

- **Relembrando que:** Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes” i.e.,
 $C = \Sigma^* \setminus (A \cup B)$

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Dito de outra forma, $L(M)$ é o conjunto de strings aceitas por M

Máquinas de Turing e Linguagens

- **Relembrando que:** Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes” i.e.,
 $C = \Sigma^* \setminus (A \cup B)$

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Dito de outra forma, $L(M)$ é o conjunto de strings aceitas por M

- **Resolvendo um problema associado à linguagem L :**

Máquinas de Turing e Linguagens

- **Relembrando que:** Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes” i.e.,
 $C = \Sigma^* \setminus (A \cup B)$

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Dito de outra forma, $L(M)$ é o conjunto de strings aceitas por M

- **Resolvendo um problema associado à linguagem L :** não queremos MT que fique em *loop*

Máquinas de Turing e Linguagens

- **Relembrando que:** Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes” i.e.,
 $C = \Sigma^* \setminus (A \cup B)$

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Dito de outra forma, $L(M)$ é o conjunto de strings aceitas por M

- **Resolvendo um problema associado à linguagem L :** não queremos MT que fique em *loop*
 $\Rightarrow$ ou seja, não basta apresentar M tal que $L(M) = L$

Máquinas de Turing e Linguagens

- **Relembrando que:** Uma certa MT pode:
 - Aceitar strings de um certo conjunto $A \subseteq \Sigma^*$
 - Rejeitar strings de um certo conjunto $B \subseteq \Sigma^*$
 - Ficar em *loop infinito* para o conjunto C das strings “restantes” i.e.,
 $C = \Sigma^* \setminus (A \cup B)$

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Dito de outra forma, $L(M)$ é o conjunto de strings aceitas por M

- **Resolvendo um problema associado à linguagem L :** não queremos MT que fique em *loop*
 $\Rightarrow$ ou seja, não basta apresentar M tal que $L(M) = L$
 $\Rightarrow$ **queremos uma Máquina de Turing que sempre para tal que $L = L(M)$.**

Máquinas de Turing e Linguagens

Do slide anterior

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M* .

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Linguagens decididas por MTs: Seja L uma linguagem.

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Linguagens decididas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing que sempre para* M tal que $L(M) = L$, dizemos que L é decidida por *decidida* M .

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Linguagens decididas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing que sempre para* M tal que $L(M) = L$, dizemos que L é decidida por *decidida* M .

Linguagens aceitas por MTs: Seja L uma linguagem.

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Linguagens decididas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* que sempre para M tal que $L(M) = L$, dizemos que L é decidida por *decidida* M .

Linguagens aceitas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* M tal que $L(M) = L$, dizemos que L é *aceita* por M .

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Linguagens decididas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* que sempre para M tal que $L(M) = L$, dizemos que L é decidida por *decidida* M .

Linguagens aceitas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* M tal que $L(M) = L$, dizemos que L é *aceita* por M .

Linguagens decididas por MTs: Também chamadas de **linguagens recursivas**

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Linguagens decididas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* que sempre para M tal que $L(M) = L$, dizemos que L é decidida por *decidida* M .

Linguagens aceitas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* M tal que $L(M) = L$, dizemos que L é *aceita* por M .

Linguagens decididas por MTs: Também chamadas de **linguagens recursivas**
Conjunto de todas linguagens recursivas: $\mathcal{R}$

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Linguagens decididas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* que sempre para M tal que $L(M) = L$, dizemos que L é decidida por *decidida* M .

Linguagens aceitas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* M tal que $L(M) = L$, dizemos que L é *aceita* por M .

Linguagens decididas por MTs: Também chamadas de **linguagens recursivas**
Conjunto de todas linguagens recursivas: $\mathcal{R}$

Linguagens aceitas por MTs: Também chamadas de **linguagens recursivamente enumeráveis**

Máquinas de Turing e Linguagens

Do slide anterior

Linguagem de uma MT: Dada uma MT $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, a linguagem

$$L(M) = \{w \in \Sigma^* : (\epsilon, q_0, w) \vdash_M^* C_F, \text{ tal que } C_F \text{ é uma configuração final de } M\}$$

é chamada de *linguagem de M*.

Linguagens decididas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* que sempre para M tal que $L(M) = L$, dizemos que L é decidida por *decidida* M .

Linguagens aceitas por MTs: Seja L uma linguagem. Se existe uma *Máquina de Turing* M tal que $L(M) = L$, dizemos que L é *aceita* por M .

Linguagens decididas por MTs: Também chamadas de **linguagens recursivas**

Conjunto de todas linguagens recursivas: $\mathcal{R}$

Linguagens aceitas por MTs: Também chamadas de **linguagens recursivamente enumeráveis**

Conjunto de todas linguagens recursivamente enumeráveis: $\mathcal{RE}$

Máquinas de Turing e Algoritmos

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L ,

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

- O algoritmo M *resolve* o problema L .

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

- O algoritmo M *resolve* o problema L .
- A MT M *decide* a linguagem L .

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

- O algoritmo M *resolve* o problema L .
- A MT M *decide* a linguagem L .

Obs: Nem todo problema de interesse em Teoria da Computação é um problema de decisão

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

- O algoritmo M *resolve* o problema L .
- A MT M *decide* a linguagem L .

Obs: Nem todo problema de interesse em Teoria da Computação é um problema de decisão

- Problemas de decisão: a saída é apenas um bit

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

- O algoritmo M *resolve* o problema L .
- A MT M *decide* a linguagem L .

Obs: Nem todo problema de interesse em Teoria da Computação é um problema de decisão

- Problemas de decisão: a saída é apenas um bit
- Problema de busca: a saída é um string de bits

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

- O algoritmo M *resolve* o problema L .
- A MT M *decide* a linguagem L .

Obs: Nem todo problema de interesse em Teoria da Computação é um problema de decisão

- Problemas de decisão: a saída é apenas um bit
- Problema de busca: a saída é um string de bits

Exercício 6 (para casa, veja no livro): Apresente uma MT que faça uma cópia da string

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

- O algoritmo M *resolve* o problema L .
- A MT M *decide* a linguagem L .

Obs: Nem todo problema de interesse em Teoria da Computação é um problema de decisão

- Problemas de decisão: a saída é apenas um bit
- Problema de busca: a saída é um string de bits

Exercício 6 (para casa, veja no livro): Apresente uma MT que faça uma cópia da string de entrada e pare (i.e., dada x , a MT para em estado de aceitação com xx na fita)

Máquinas de Turing e Algoritmos

Definição: Um *Algoritmo* é uma Máquina de Turing que sempre para.

Terminologia: Uma linguagem também é chamada de *Problema de Decisão*

Terminologia: Dado um problema de decisão L , se existe um *Algoritmo* M tal que $L(M) = L$, usaremos as duas terminologias:

- O algoritmo M *resolve* o problema L .
- A MT M *decide* a linguagem L .

Obs: Nem todo problema de interesse em Teoria da Computação é um problema de decisão

- Problemas de decisão: a saída é apenas um bit
- Problema de busca: a saída é um string de bits

Exercício 6 (para casa, veja no livro): Apresente uma MT que faça uma cópia da string de entrada e pare (i.e., dada x , a MT para em estado de aceitação com xx na fita)

Exercício 7: Mostre que $\mathcal{R} \subseteq \mathcal{RE}$.

Variações de Máquinas de Turing

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um.

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita.

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Definição: Uma *MT com k fitas* é uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, com os componentes da tupla iguais às MTs com uma fita,

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Definição: Uma *MT com k fitas* é uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, com os componentes da tupla iguais às MTs com uma fita, com a exceção de que δ é uma função do tipo $\delta : (Q \setminus F) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \mathcal{D}^k$.

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Definição: Uma *MT com k fitas* é uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, com os componentes da tupla iguais às MTs com uma fita, com a exceção de que δ é uma função do tipo $\delta : (Q \setminus F) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \mathcal{D}^k$.

Obs: Neste caso, a computação começa com a string de entrada na fita 1

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Definição: Uma *MT com k fitas* é uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, com os componentes da tupla iguais às MTs com uma fita, com a exceção de que δ é uma função do tipo $\delta : (Q \setminus F) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \mathcal{D}^k$.

Obs: Neste caso, a computação começa com a string de entrada na fita 1

Teorema: Se uma MT M com alfabeto da fita Γ decide uma linguagem L ,

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Definição: Uma MT com k fitas é uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, com os componentes da tupla iguais às MTs com uma fita, com a exceção de que δ é uma função do tipo $\delta : (Q \setminus F) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \mathcal{D}^k$.

Obs: Neste caso, a computação começa com a string de entrada na fita 1

Teorema: Se uma MT M com alfabeto da fita Γ decide uma linguagem L , então existe uma MT M' com alfabeto da fita $\Gamma' = \{0, 1, B\}$ que decide L .

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Definição: Uma MT com k fitas é uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, com os componentes da tupla iguais às MTs com uma fita, com a exceção de que δ é uma função do tipo $\delta : (Q \setminus F) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \mathcal{D}^k$.

Obs: Neste caso, a computação começa com a string de entrada na fita 1

Teorema: Se uma MT M com alfabeto da fita Γ decide uma linguagem L , então existe uma MT M' com alfabeto da fita $\Gamma' = \{0, 1, B\}$ que decide L .

Teorema: Se uma MT M com k fitas decide uma linguagem L ,

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Definição: Uma MT com k fitas é uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, com os componentes da tupla iguais às MTs com uma fita, com a exceção de que δ é uma função do tipo $\delta : (Q \setminus F) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \mathcal{D}^k$.

Obs: Neste caso, a computação começa com a string de entrada na fita 1

Teorema: Se uma MT M com alfabeto da fita Γ decide uma linguagem L , então existe uma MT M' com alfabeto da fita $\Gamma' = \{0, 1, B\}$ que decide L .

Teorema: Se uma MT M com k fitas decide uma linguagem L , então existe uma MT M' com uma fita que decide L .

Variações de Máquinas de Turing

- O alfabeto Γ de nossas MTs pode ser qualquer um. E se fosse apenas binário?
- Nossa máquina tem apenas uma fita. E se tivesse duas?

Definição: Uma MT com k fitas é uma 7-tupla $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$, com os componentes da tupla iguais às MTs com uma fita, com a exceção de que δ é uma função do tipo $\delta : (Q \setminus F) \times \Gamma^k \rightarrow Q \times \Gamma^k \times \mathcal{D}^k$.

Obs: Neste caso, a computação começa com a string de entrada na fita 1

Teorema: Se uma MT M com alfabeto da fita Γ decide uma linguagem L , então existe uma MT M' com alfabeto da fita $\Gamma' = \{0, 1, B\}$ que decide L .

Teorema: Se uma MT M com k fitas decide uma linguagem L , então existe uma MT M' com uma fita que decide L .

- Ideias centrais das demonstrações nos próximos dois slides

Simulando alfabetos arbitrários com fita binária

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X, Y , não podem ser lidos/escritos por M'

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X, Y , não podem ser lidos/escritos por M' (exceto B)

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X , Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X , Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário
por exemplo, X torna-se 00 , Y torna-se 11 , e assim por diante

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X, Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário
por exemplo, X torna-se 00 , Y torna-se 11 , e assim por diante

Suponha que a máquina M tenha uma transição $\delta(p, X) = (q, Y, R)$,

Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X, Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário
por exemplo, X torna-se 00 , Y torna-se 11 , e assim por diante

Suponha que a máquina M tenha uma transição $\delta(p, X) = (q, Y, R)$, $X, Y \notin \{0, 1, B\}$

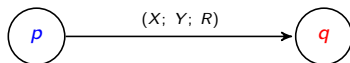
Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X, Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário
por exemplo, X torna-se 00 , Y torna-se 11 , e assim por diante

Suponha que a máquina M tenha uma transição $\delta(p, X) = (q, Y, R)$, $X, Y \notin \{0, 1, B\}$



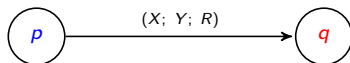
Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X , Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário
por exemplo, X torna-se 00 , Y torna-se 11 , e assim por diante

Suponha que a máquina M tenha uma transição $\delta(p, X) = (q, Y, R)$, $X, Y \notin \{0, 1, B\}$



M' terá um estado adicional $r_{p,q}$ e a transição original é substituída por duas transições:

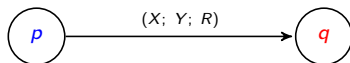
Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

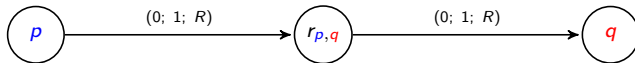
Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X, Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário
por exemplo, X torna-se 00 , Y torna-se 11 , e assim por diante

Suponha que a máquina M tenha uma transição $\delta(p, X) = (q, Y, R)$, $X, Y \notin \{0, 1, B\}$



M' terá um estado adicional $r_{p,q}$ e a transição original é substituída por duas transições:



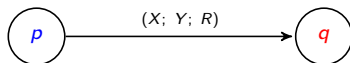
Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

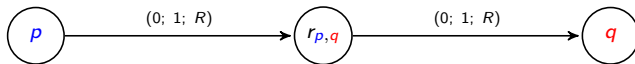
Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X, Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário
por exemplo, X torna-se 00 , Y torna-se 11 , e assim por diante

Suponha que a máquina M tenha uma transição $\delta(p, X) = (q, Y, R)$, $X, Y \notin \{0, 1, B\}$



M' terá um estado adicional $r_{p,q}$ e a transição original é substituída por duas transições:



Note que M' tem mais estados que M e executa mais passos

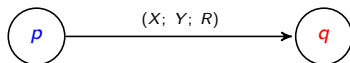
Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ que decide uma certa linguagem.

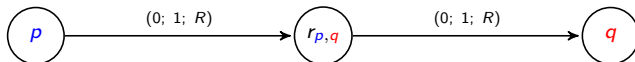
Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ que decide a mesma linguagem:

- Símbolos não binários de M , e.g., X, Y , não podem ser lidos/escritos por M' (exceto B)
- Mas M' pode ler/escrever os símbolos codificados em binário
por exemplo, X torna-se 00 , Y torna-se 11 , e assim por diante

Suponha que a máquina M tenha uma transição $\delta(p, X) = (q, Y, R)$, $X, Y \notin \{0, 1, B\}$



M' terá um estado adicional $r_{p,q}$ e a transição original é substituída por duas transições:



Note que M' tem mais estados que M e executa mais passos (isso não é um problema)

Simulando duas fitas em uma

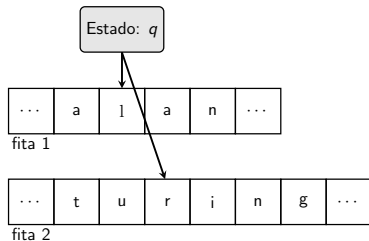
Seja M com **duas fitas** que resolve certo problema.

Simulando duas fitas em uma

Seja M com **duas fitas** que resolve certo problema. Digamos que $\Gamma = \{a, b, c, \dots, z, B\}$

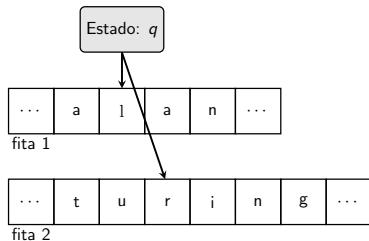
Simulando duas fitas em uma

Seja M com **duas fitas** que resolve certo problema. Digamos que $\Gamma = \{a, b, c, \dots, z, B\}$



Simulando duas fitas em uma

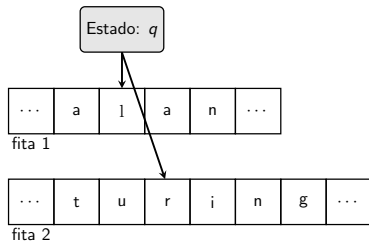
Seja M com **duas fitas** que resolve certo problema. Digamos que $\Gamma = \{a, b, c, \dots, z, B\}$



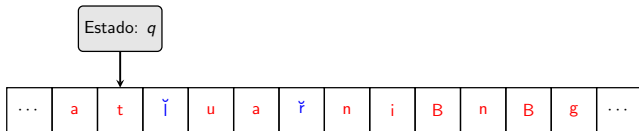
A MT M' com **uma fita** para o mesmo problema terá $\Gamma' = \{a, b, \dots, z, B, \check{a}, \check{b}, \dots, \check{z}, \check{B}\}$,

Simulando duas fitas em uma

Seja M com **duas fitas** que resolve certo problema. Digamos que $\Gamma = \{a, b, c, \dots, z, B\}$

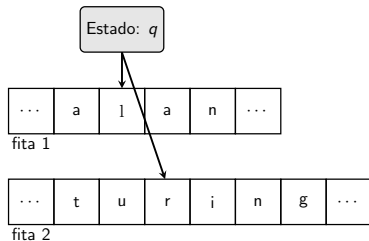


A MT M' com **uma fita** para o mesmo problema terá $\Gamma' = \{a, b, \dots, z, B, \check{a}, \check{b}, \dots, \check{z}, \check{B}\}$,

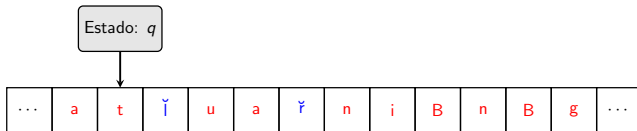


Simulando duas fitas em uma

Seja M com **duas fitas** que resolve certo problema. Digamos que $\Gamma = \{a, b, c, \dots, z, B\}$



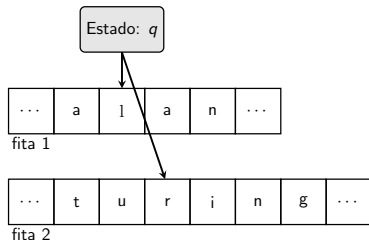
A MT M' com **uma fita** para o mesmo problema terá $\Gamma' = \{a, b, \dots, z, B, \check{a}, \check{b}, \dots, \check{z}, \check{B}\}$,



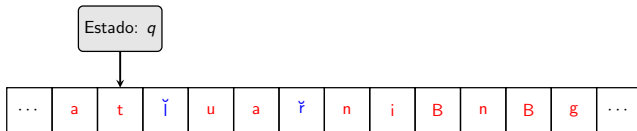
Conteúdo das fitas 1 e 2
intercalados na fita única

Simulando duas fitas em uma

Seja M com **duas fitas** que resolve certo problema. Digamos que $\Gamma = \{a, b, c, \dots, z, B\}$



A MT M' com **uma fita** para o mesmo problema terá $\Gamma' = \{a, b, \dots, z, B, \check{a}, \check{b}, \dots, \check{z}, \check{B}\}$,

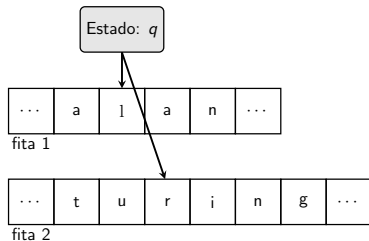


Conteúdo das fitas 1 e 2
intercalados na fita única

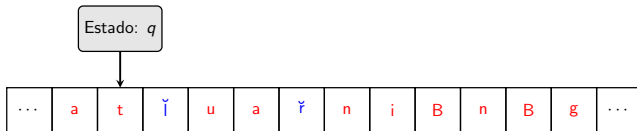
Símbolos novos usados
para marcar cabeçotes

Simulando duas fitas em uma

Seja M com **duas fitas** que resolve certo problema. Digamos que $\Gamma = \{a, b, c, \dots, z, B\}$



A MT M' com **uma fita** para o mesmo problema terá $\Gamma' = \{a, b, \dots, z, B, \check{a}, \check{b}, \dots, \check{z}, \check{B}\}$,



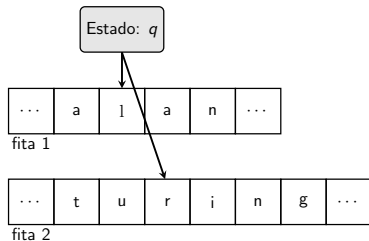
Conteúdo das fitas 1 e 2
intercalados na fita única

Símbolos novos usados
para marcar cabeçotes

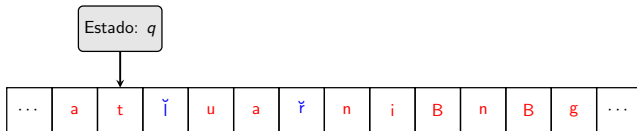
Além de alfabeto maior, M' tem mais estados e executa mais passos

Simulando duas fitas em uma

Seja M com **duas fitas** que resolve certo problema. Digamos que $\Gamma = \{a, b, c, \dots, z, B\}$



A MT M' com **uma fita** para o mesmo problema terá $\Gamma' = \{a, b, \dots, z, B, \check{a}, \check{b}, \dots, \check{z}, \check{B}\}$,



Conteúdo das fitas 1 e 2
intercalados na fita única

Símbolos novos usados
para marcar cabeçotes

Além de alfabeto maior, M' tem mais estados e executa mais passos

Note: Esta demonstração pode ser generalizada para k fitas

Exemplo de MT para um problema de busca

Exemplo de MT para um problema de busca

Exercício 8: Projete uma MT que some dois números a e b .

Exemplo de MT para um problema de busca

Exercício 8: Projete uma MT que some dois números a e b .

- Seja $a = a_1 a_2 \cdots a_n$ e $b = b_1 b_2 \cdots b_n$.

Exemplo de MT para um problema de busca

Exercício 8: Projete uma MT que some dois números a e b .

- Seja $a = a_1 a_2 \cdots a_n$ e $b = b_1 b_2 \cdots b_n$.
- MT com duas fitas, $\Sigma = \{0, 1, \#\}$, $\Gamma = \{0, 1, \#, B\}$,

Exemplo de MT para um problema de busca

Exercício 8: Projete uma MT que some dois números a e b .

- Seja $a = a_1 a_2 \cdots a_n$ e $b = b_1 b_2 \cdots b_n$.
- MT com duas fitas, $\Sigma = \{0, 1, \#\}$, $\Gamma = \{0, 1, \#, B\}$,
Recebe na primeira fita $a_1 a_2 \cdots a_n \# b_1 b_2 \cdots b_n$,

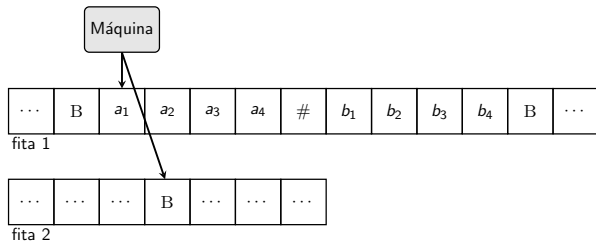
Exemplo de MT para um problema de busca

Exercício 8: Projete uma MT que some dois números a e b .

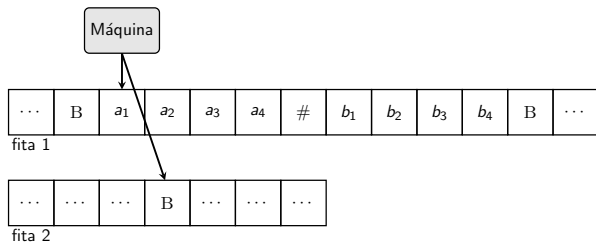
- Seja $a = a_1 a_2 \dots a_n$ e $b = b_1 b_2 \dots b_n$.
- MT com duas fitas, $\Sigma = \{0, 1, \#\}$, $\Gamma = \{0, 1, \#, B\}$,

Recebe na primeira fita $a_1 a_2 \dots a_n \# b_1 b_2 \dots b_n$,
calcula a soma e termina a execução com o resultado na segunda fita.

Somando com MTs

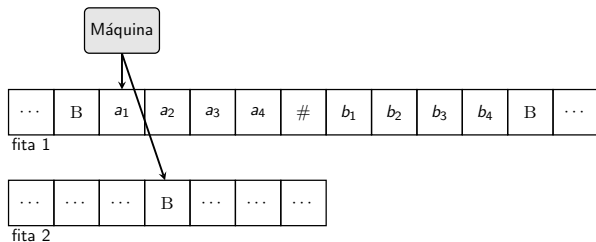


Somando com MTs



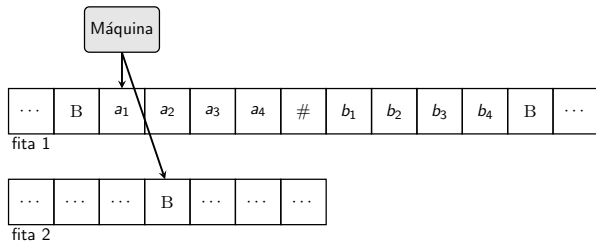
Passos iniciais: A MT copia b para a segunda fita.

Somando com MTs

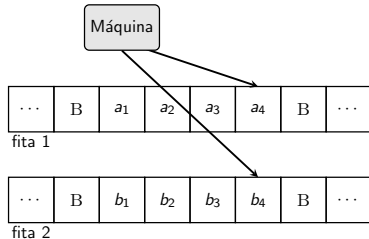


Passos iniciais: A MT copia b para a segunda fita. Depois disso, a configuração será:

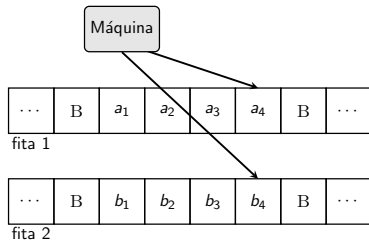
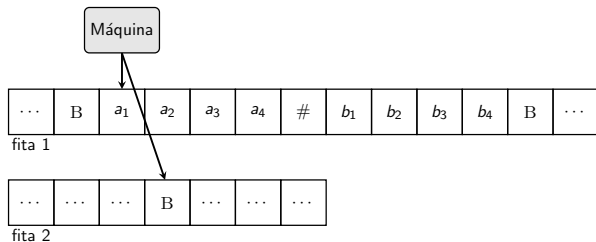
Somando com MTs



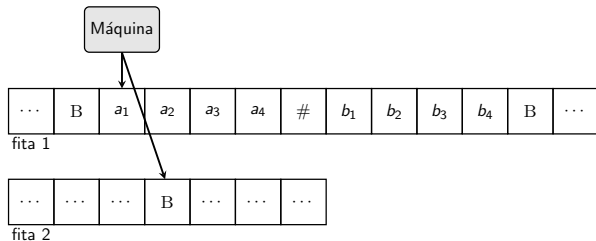
Passos iniciais: A MT copia b para a segunda fita. Depois disso, a configuração será:



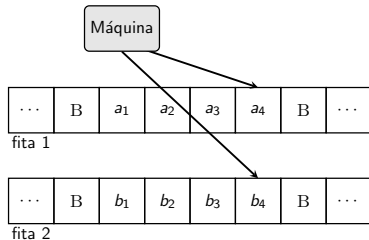
Somando com MTs



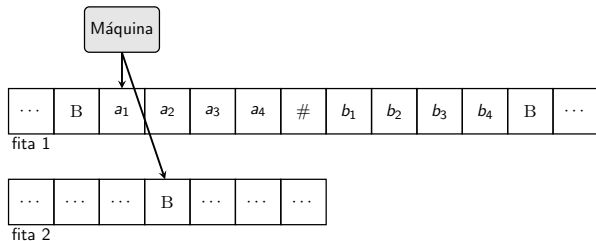
Somando com MTs



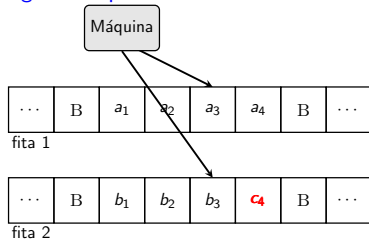
Agora computa a soma:



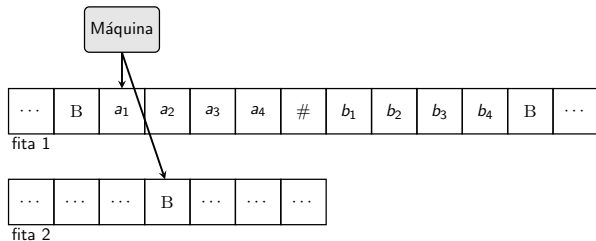
Somando com MTs



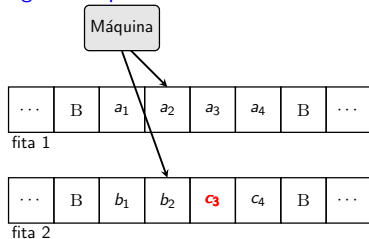
Agora computa a soma:



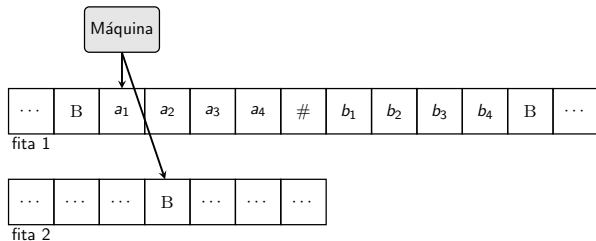
Somando com MTs



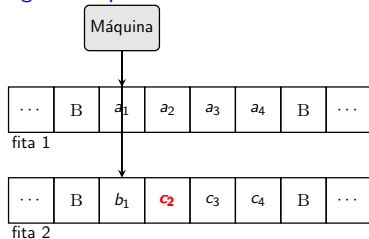
Agora computa a soma:



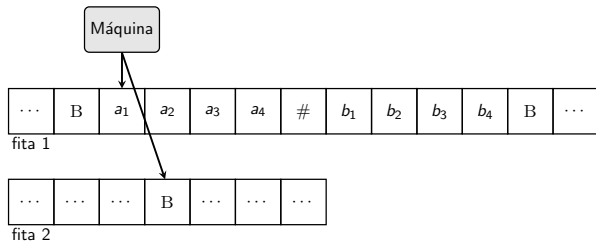
Somando com MTs



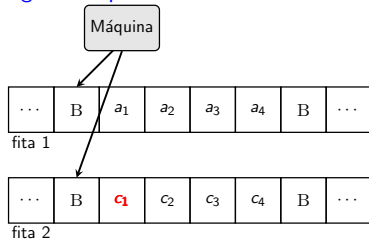
Agora computa a soma:



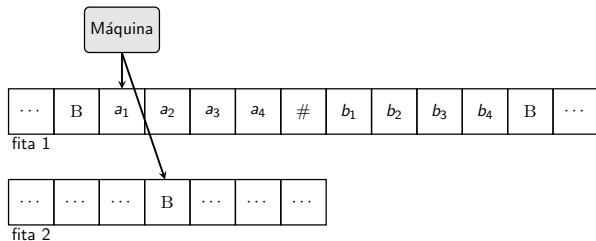
Somando com MTs



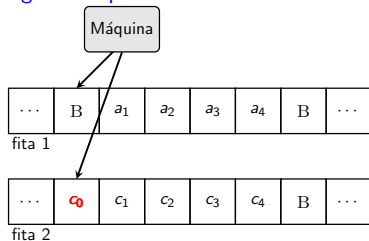
Agora computa a soma:



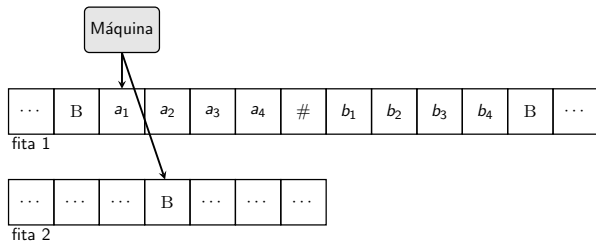
Somando com MTs



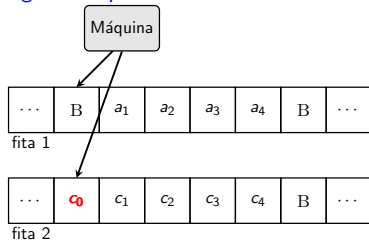
Agora computa a soma:



Somando com MTs



Agora computa a soma:



Obs: c_0 é B, caso não haja *vai um* no último bit.