

Introdução à Teoria da Computação

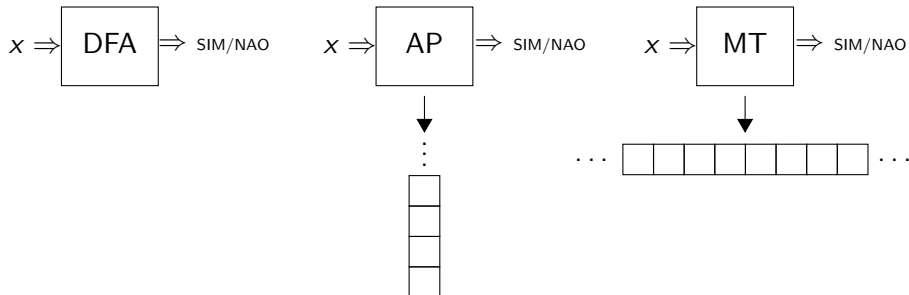
Máquinas de Turing

Professor Murilo V. G. da Silva

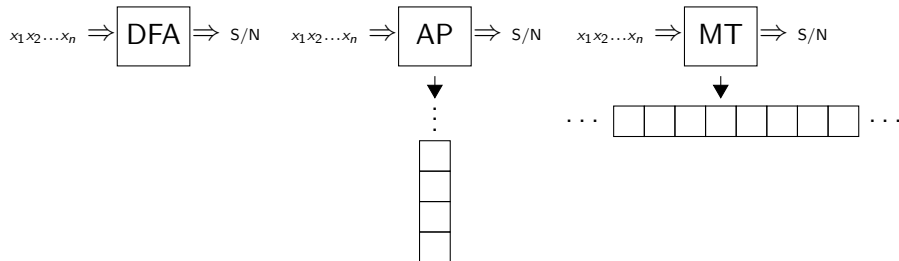
Departamento de Informática
Universidade Federal do Paraná

2025 / 2

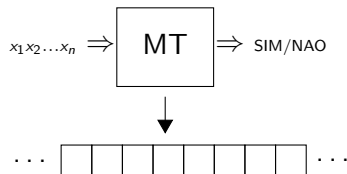
Máquinas de Turing



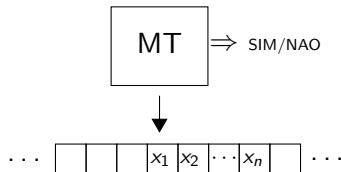
Máquinas de Turing



Máquinas de Turing



Máquinas de Turing



Máquinas de Turing

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

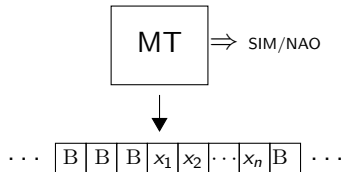
- Q é o conjunto finito de estados
- Σ é o alfabeto de entrada.
- Γ é o alfabeto da fita, tal que $\Sigma \subseteq \Gamma$.
-
- q_0 é o estado inicial
-
- $F \subseteq Q$ é o conjunto de estados finais.

Máquinas de Turing

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 $\Sigma \subseteq \Gamma$.
-
- 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



Máquinas de Turing

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 $\Sigma \subseteq \Gamma$.
-
- 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

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 $\Sigma \subseteq \Gamma$.
-
- 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.

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 $\Sigma \subseteq \Gamma$.
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times D$
- 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

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 $\Sigma \subseteq \Gamma$.
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$
- 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

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 $\Sigma \subseteq \Gamma$.
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$ 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.

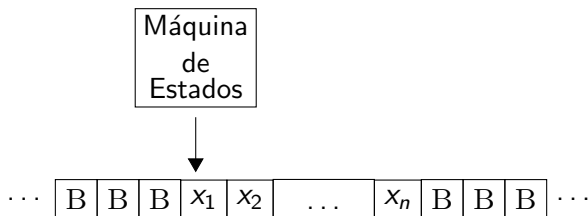
Uma **Máquina de Turing** é uma 7-tupla $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$,

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

Máquinas de Turing

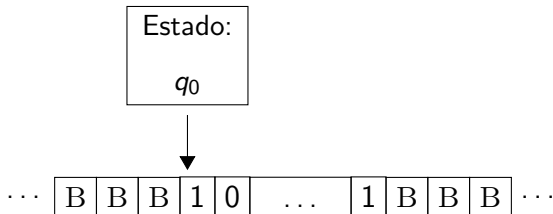
Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$.

- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$ e que $D = \{L, R, S\}$.



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

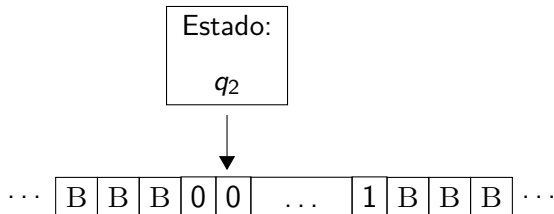
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$ e que $D = \{L, R, S\}$.



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

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

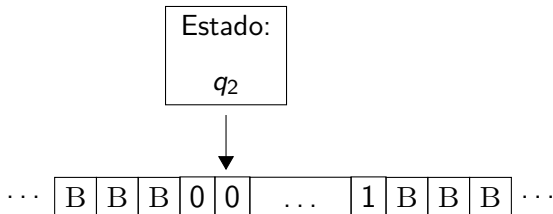
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$ e que $D = \{L, R, S\}$.



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

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

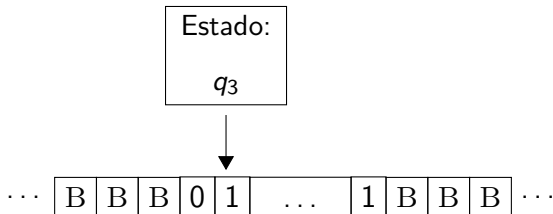
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$ e que $D = \{L, R, S\}$.



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

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

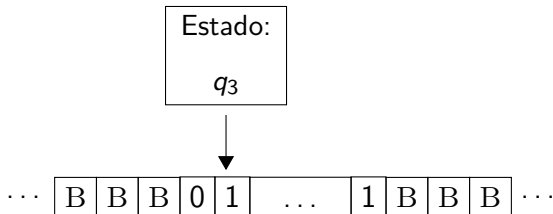
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$ e que $D = \{L, R, S\}$.



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

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

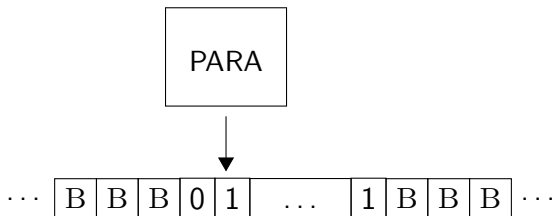
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$ e que $D = \{L, R, S\}$.



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

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

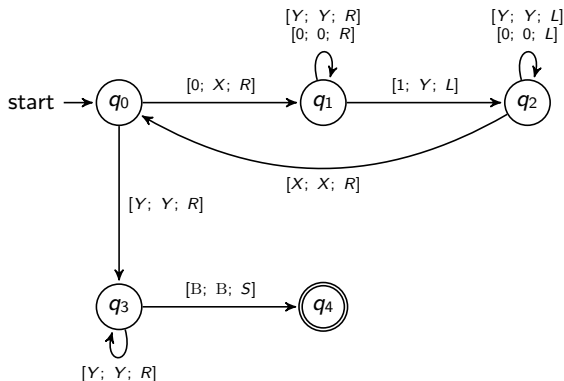
- $\delta : (Q \setminus F) \times \Gamma \rightarrow Q \times \Gamma \times D$ e que $D = \{L, R, S\}$.



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

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

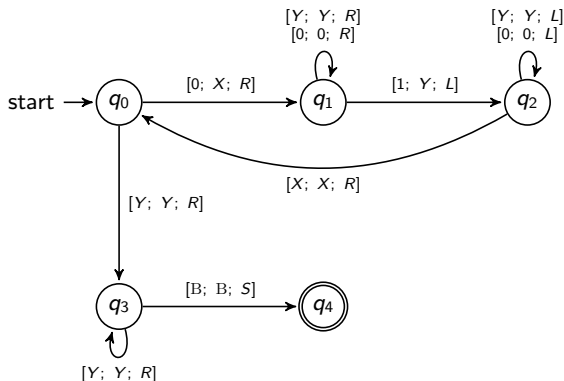
Máquinas de Turing



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

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

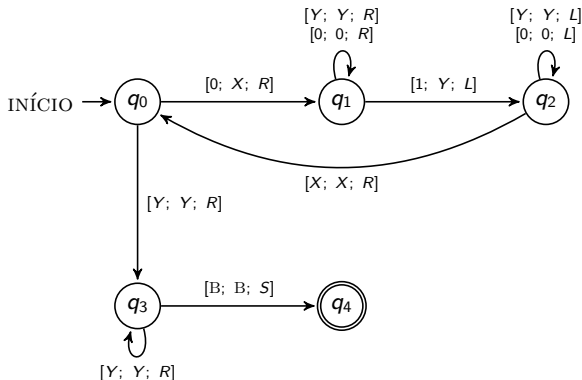
Máquinas de Turing



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 primeiro símbolo de β

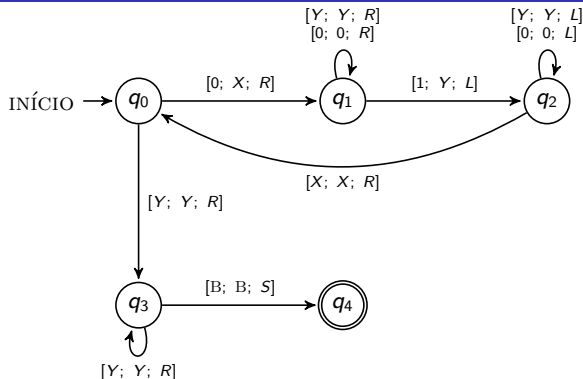
Máquinas de Turing



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



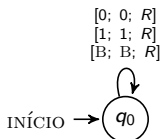
Def.: Uma configuração de $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$ é uma tripla (α, q, β) , $\alpha, \beta \in \Gamma^*$ e $q \in Q$.

Sequência de passos computacionais:

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

Máquinas de Turing podem ficar em *loop infinito*

Considere a abaixo:



$M = (\{q_0\}, \{0, 1\}, \{0, 1, B\}, \delta, q_0, B, \{q_0\})$, sendo $\delta(q_0, S) = (q_0, S, R)$ para todo $S \in \Gamma$

- $\forall x \in \Sigma^*$, M fica em *loop infinito* com a entrada x
- Note que a MT M_{01} (slide anterior) não fica em *loop infinito* para nenhuma entrada
- MTs (em geral) podem parar para um conjunto de strings e ficar em loop para as demais.

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

Mostre que neste caso sempre existe uma MT M' tal que:

- (1) Se M para em estado final para uma string x , então M' deve ficar em loop para x
- (2) Se M para em estado que não é final para x , então M' também para em estado que não é final para x .

Máquinas de Turing: Configurações

- 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)**

Máquinas de Turing: Configurações

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 Final 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 $w \in \Sigma^*$

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

- sendo 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 $(\epsilon, q_0, w) \vdash_M^* C_N$,

- 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.,
 $C = \Sigma^* \setminus (A \cup B)$

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 de decisão L : a MT que nunca deve ficar 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

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

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 .

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

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 $(Q, \Sigma, \Gamma, \delta, q_0, F)$, semelhante às MTs com uma fita, com a diferença sendo que δ é uma função $\delta : (Q \setminus F) \times \Gamma^k \rightarrow Q \times \Gamma^k \times D^k$.

Obs: A string de entrada fica armazenada 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 .

- Rascunho das provas nos próximos dois slides

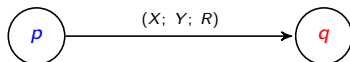
Simulando alfabetos arbitrários com fita binária

Seja M com alfabeto da fita Γ para um certo problema.

Vamos mostrar $\exists M'$ com alfabeto $\{0, 1, B\}$ para o mesmo problema:

- Note que M' não pode ler/escrever X, Y de M que não sejam binários
- M' escreve símbolos codificados em binário: e.g., X torna-se 00 e Y torna-se 11

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



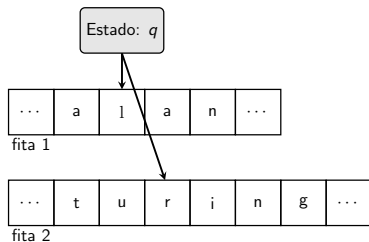
Transições equivalentes para M' :



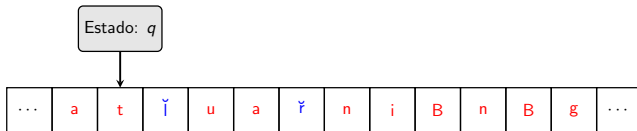
Note que M' tem mais estados que M e executa mais passos (isso não é um problema)

Simulando duas fitas em uma

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

Note: Esta demonstração pode ser generalizada para k fitas

Mais exemplos de Máquinas de Turing

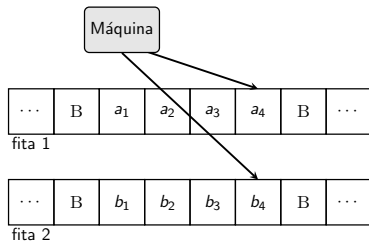
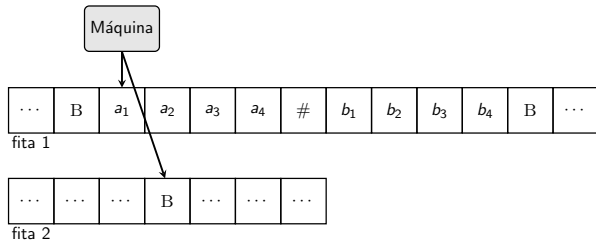
Exercício: Projete uma MT que some dois números a e b . $a = a_1a_2 \cdots a_n$, $b = b_1b_2 \cdots b_n$

- MT com duas fitas, $\Sigma = \{0, 1, \#\}$, $\Gamma = \{0, 1, \#, B\}$,

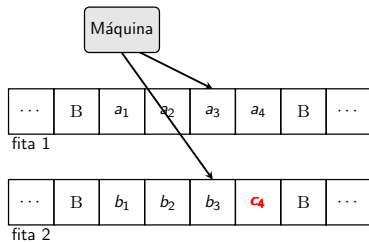
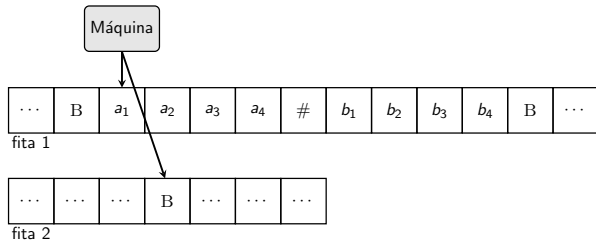
Recebe na primeira fita $a_1a_2 \cdots a_n\#b_1b_2 \cdots b_n$,

calcula a soma e termina a execução com o resultado na segunda fita.

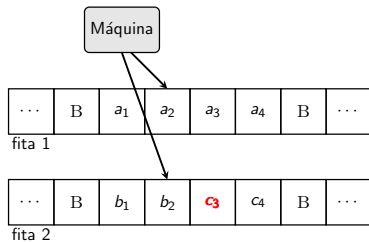
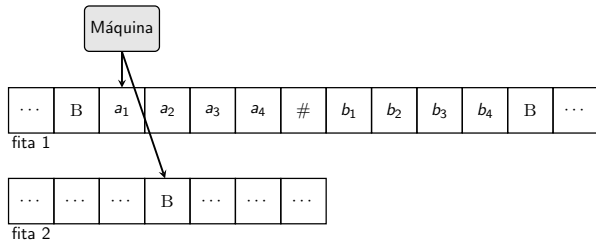
Somando com MTs



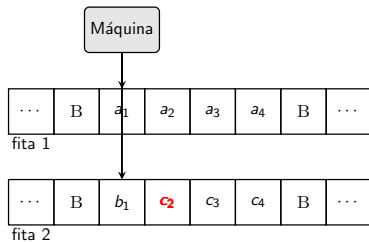
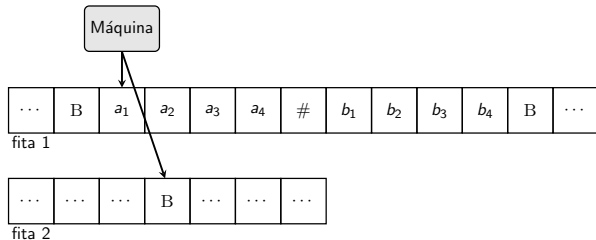
Somando com MTs



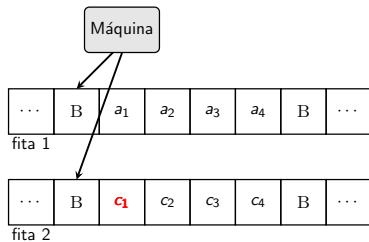
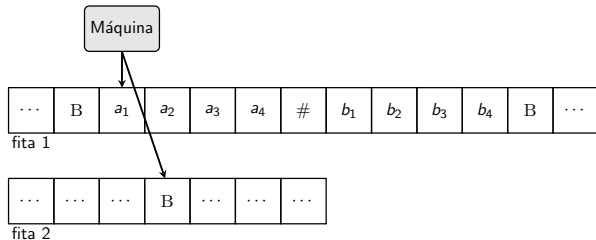
Somando com MTs



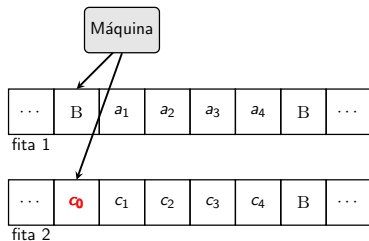
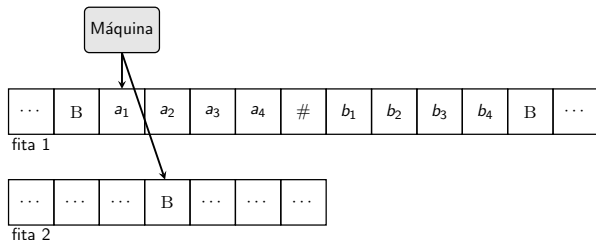
Somando com MTs



Somando com MTs



Somando com MTs



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