

5. Máquinas de Turing

5.1 Introdução

As máquinas de Turing são uma proposta de formalização da noção de procedimento efetivo (programa que sempre fornece uma resposta quando executado em um computador)

A máquina de Turing não é nada mais do que um autômato finito determinístico com uma fita, onde será escrita a palavra a processar, e uma cabeça de leitura que pode movimentar-se, tanto à direita, quanto à esquerda. Com a cabeça de leitura podemos então ler e escrever símbolos do alfabeto de fita.

5.2 Definição de MT

Uma máquina de Turing é formalmente descrita pela tupla de 7 valores:

$M = (Q, \Gamma, \Sigma, \delta, s, B, F)$ onde:

- Q é um conjunto finito de estados
- Γ é o alfabeto de fita
- $\Sigma \subseteq \Gamma$ é o alfabeto de entrada
- $s \in Q$ é o estado inicial
- $F \subseteq Q$ é o conjunto dos estados de aceitação
- $B \in \Gamma - \Sigma$ é o símbolo branco (geralmente anotado #)
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ é a função de transição (L=Left e R=Right)

Uma transição $\delta(q, b) = (q', b', S)$ significa:

> Estando o autômato no estado q , com a cabeça apontando para o símbolo b , a transição pode ser disparada. Como resultado do disparo, o estado muda para q' ; o símbolo b da fita é substituído pelo símbolo b' ; e, finalmente, a cabeça se desloca para o sentido S (onde S é R - direita-, ou L - esquerda). Neste caso, o arco no grafo que representa tal transição terá como rótulo: $b/b'/S$.

- A configuração de uma máquina de Turing é um elemento que pertence a:

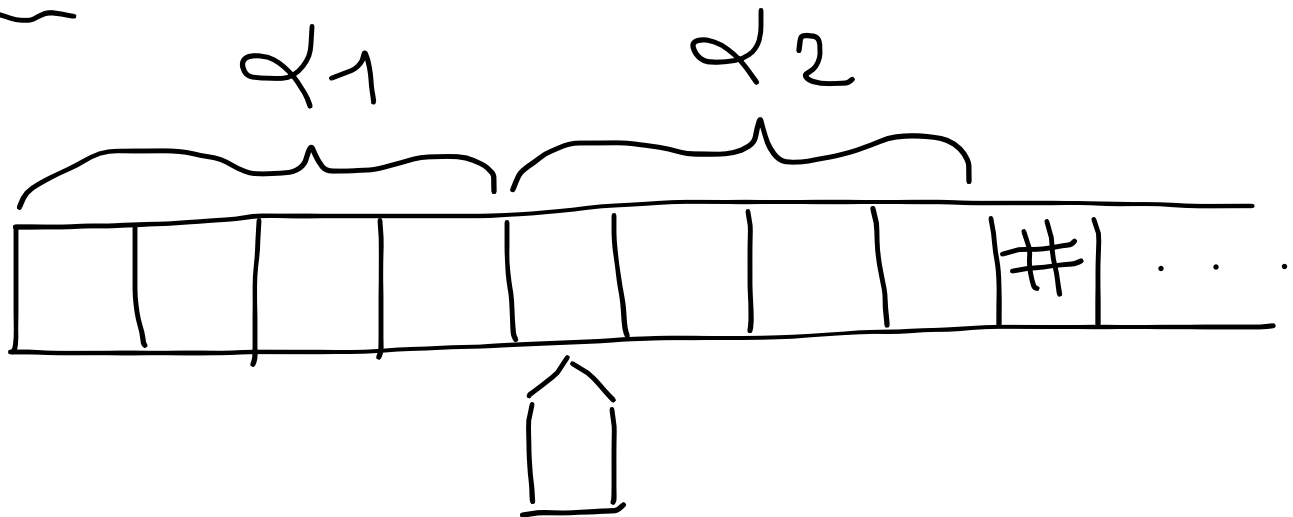
$Q \times \Gamma^* \times (\epsilon \cup \Gamma^* \cdot (\Gamma - \{B\}))$, ou seja, uma configuração é uma tripla: (q, α_1, α_2) , onde:

> $q \in Q$; $\alpha_1 \in \Gamma^*$

> $\alpha_2 \in \epsilon \cup \Gamma^* \cdot (\Gamma - \{B\})$, ou seja, α_2 é a palavra que se encontra entre a cabeça (inclusive) e o último caracter não branco.

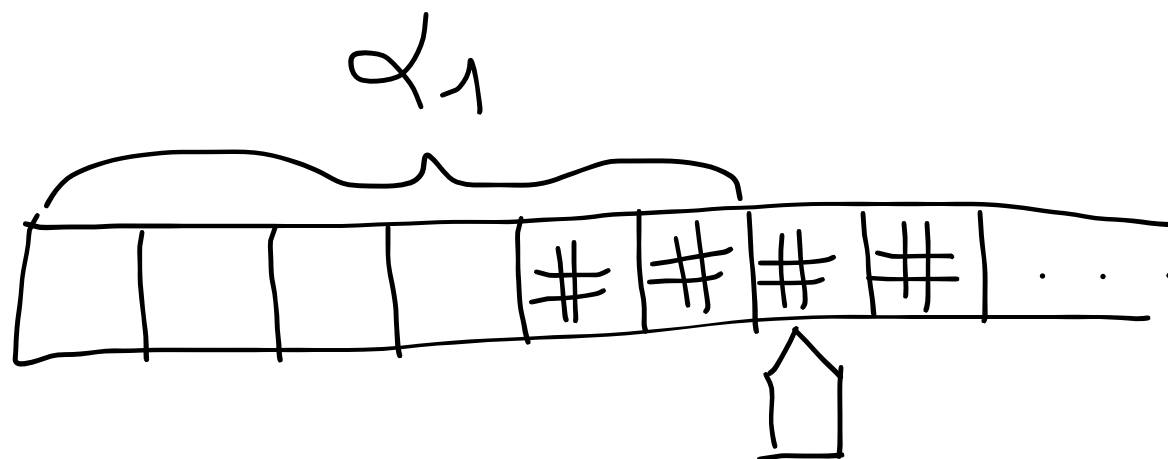
- Informalmente, a configuração será dada pelo estado do autômato finito, pela palavra que aparece na fita e pela posição da cabeça de leitura. Uma forma de definir a posição da cabeça de leitura é considerar a palavra que aparece antes da cabeça de leitura e a palavra que se encontra entre a posição da cabeça e o último símbolo não branco na fita.

Ex 1



$$\text{CONF}ig = (q, q_1, q_2)$$

Ex 2:



$$(ONFIG = (q, \alpha_1, \varepsilon))$$

Ex 3:

α_1		α_2				
		b	#	#	...	

↑

config: (q, α_1, b)

Ex 4:

α_1			α_2				
		#	#	a	#	b	#

↑

config: $(q, \alpha_1, \#a\#b)$

Ex 5:

α_1			α_2				
		a	#	#	a	#	#

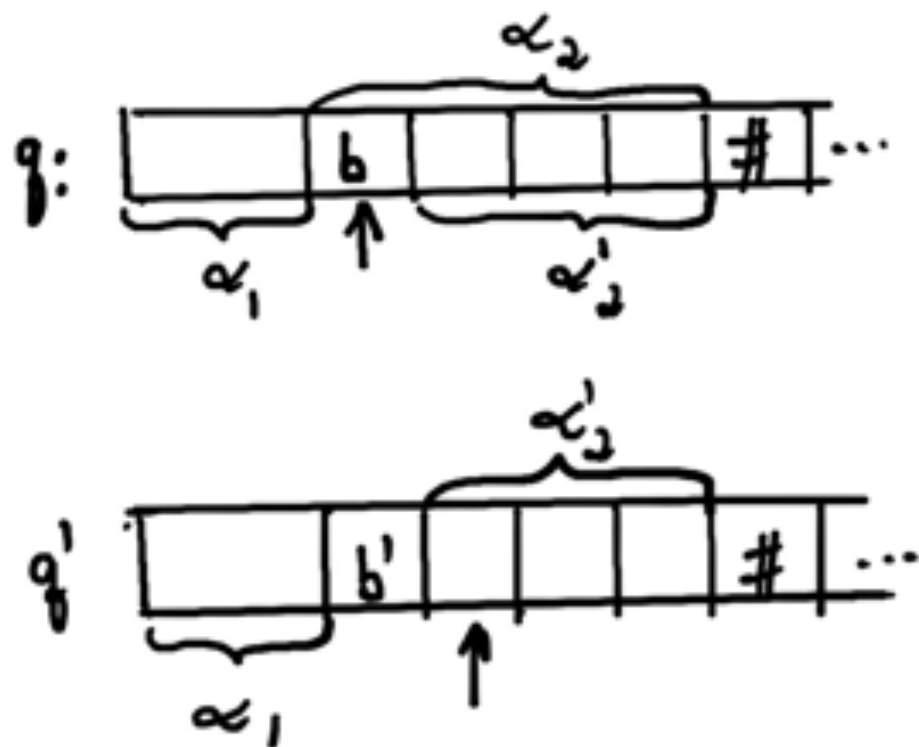
↑

config: $(q, \alpha_1, \#\#a)$

- São as seguintes as configurações deriváveis em um único passo a partir da configuração (q, α_1, α_2) em uma Máquina de Turing M:

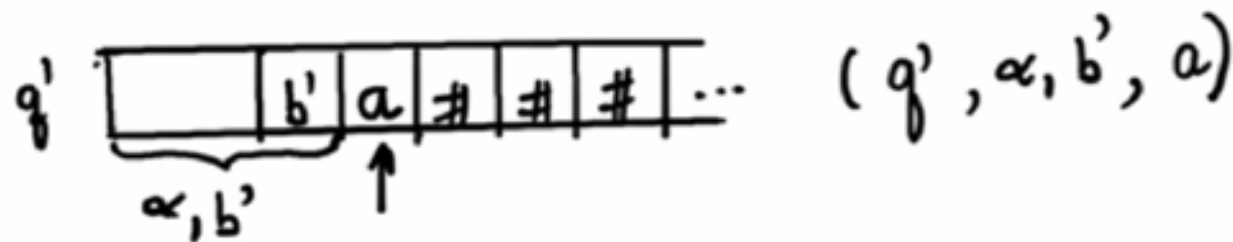
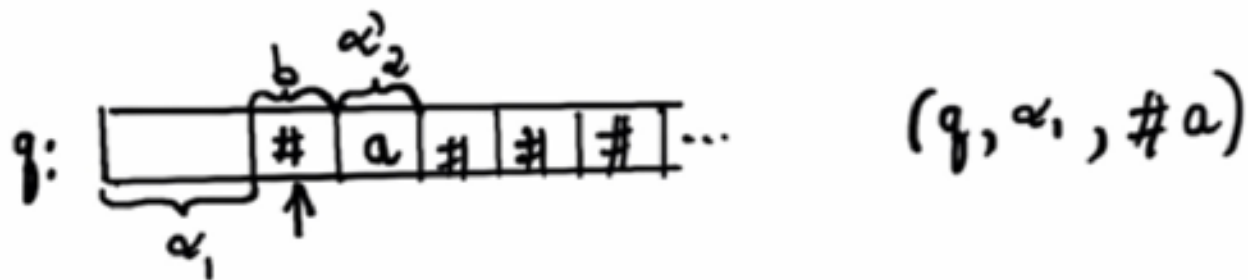
- 1) Para regras de transição do tipo $\delta(q, b) = (q', b', R)$, re-escrevendo (q, α_1, α_2) como $(q, \alpha_1, b\alpha'_2)$, onde, obviamente, $b = \#$ quando $\alpha_2 = \varepsilon$, tem-se:

$$(q, \alpha_1, b\alpha'_2) \vdash (q', \alpha_1 b', \alpha'_2)$$

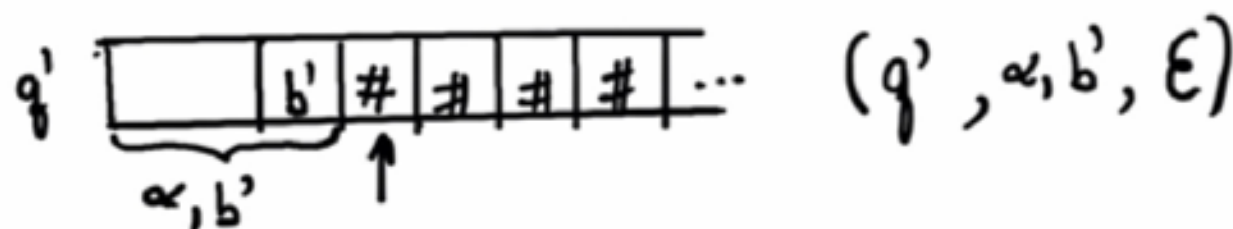
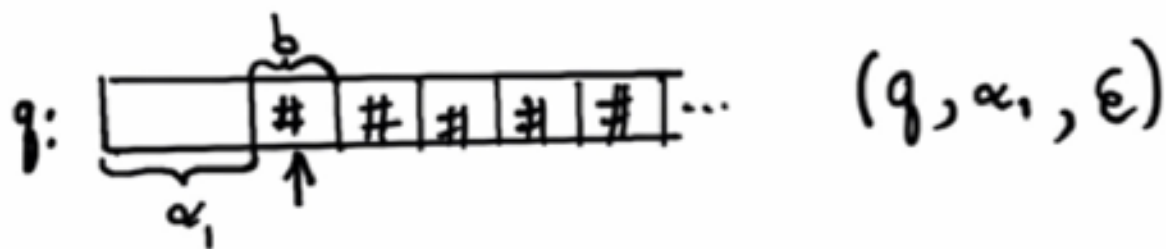


- Alguns exemplos dessa transição para a direita em um único passo pela regra $\delta(q, b) = (q', b', R)$:

Quando $\alpha_2 = \#a$; e $b = \#$:

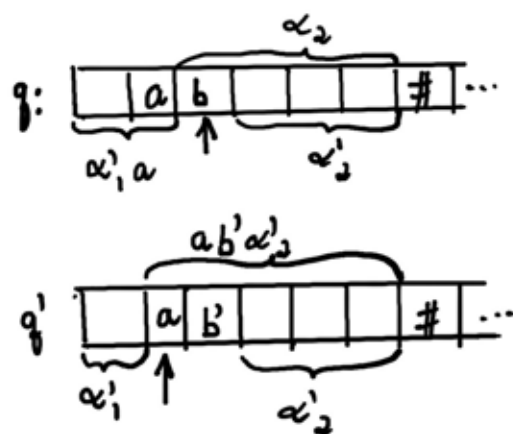


Quando $\alpha_2 = \epsilon$; e $b = \#$



- 2) Para regras de transição do tipo. $\delta(q, b) = (q', b', l)$, re-escrevendo (q, α_1, α_2) como $(q, \alpha'_1 a, b\alpha'_2)$, onde $\alpha_1 \neq \varepsilon$ (pois, no nosso modelo de MT a fita é finita à esquerda e infinita à direita), isto é, $\alpha_1 = \alpha'_1 a$, e $b = \#$ quando $\alpha_2 = \varepsilon$, tem-se:

$$(q, \alpha'_1 a, b\alpha'_2) \vdash (q', \alpha'_1, ab'\alpha'_2)$$



Definição : uma configuração C' é derivável em várias etapas a partir da configuração C pela máquina M se existe $k \geq 0$ e as configurações intermediárias C_0, C_1, \dots, C_k tais que:

- . $C = C_0$
- . $C' = C_k$
- . $C_i \vdash_M C_{i+1}$ para $0 \leq i < k$

> Uma palavra w é aceita por uma máquina de Turing M se, e somente se, a execução de w por M leva a uma configuração cujo estado é de aceitação.

Definição : A linguagem $L(M)$ aceita por uma máquina de Turing é o conjunto de palavras w tais que :

$$(S, \varepsilon, w) \vdash_m^* (P, \alpha_1, \alpha_2)$$

$$\text{com} \quad p \in F$$

OBSERVAÇÃO IMPORTANTE:

> Teoricamente, o símbolo # (branco) faz parte do alfabeto da fita e pode ocupar qualquer posição dela. Contudo, como esse símbolo não faz parte do alfabeto de entrada, ele não pode compor as palavras processadas por uma Máquina de Turing (MT).

Assim sendo, a configuração inicial dessas máquinas para processar uma palavra w deve ser: (s, ϵ, w) , onde s é o estado inicial. Assim sendo, nesse início de execução, a palavra w deve ser alocada a partir da primeira posição à esquerda da fita e, imediatamente após ela, deve ser colocado o símbolo #. Dessa forma, caso a palavra w seja, por exemplo um programa, não se deve confundir os símbolos de "espaço" de w com esse símbolo de branco.

Exercício: Construir a máquina que aceita a linguagem $a(n)b(n)$ para $n > 0$. Testar a máquina com a palavra de entrada aaabbb

Dica: $\Gamma = \{a, b, X, Y, \#\}$ e $\Sigma = \{a, b\}$

- > VER PASTA: "Exemplos-Maq-Turing" (LFA) ou "TC-2-TM1-Exemplos" (TC)
- > FAZER LISTAS DE EXERCÍCIOS: "Lista-3" (TC) ou "Lista-9" (LFA).

5.3 Linguagens aceitas e decididas

Consideremos as configurações derivadas a partir de uma configuração qualquer (s, ε, w) . Várias situações podem ocorrer:

- 1) na seqüência de configurações aparece um estado de aceitação e a palavra w é então aceita;
- 2) a seqüência de configurações atinge um estado para o qual não existe função de transição definida: neste caso, a palavra w não é aceita;
- 3) a seqüência de configurações não passa por um estado de aceitação e é infinita, situação em que a MT não consegue definir se w é aceita ou não.

Diz-se que uma MT ACEITA uma linguagem L quando há a garantia de que ela aceita todas as suas instâncias positivas, contudo não há a garantia de que ela consiga concluir o processamento de todas as instâncias negativas (pode haver instâncias negativas cujos processamentos representam uma seqüência de configurações infinita, que não passa por um estado de aceitação). Logo, as MT não são um procedimento efetivo para as linguagens aceitas por ele.

Definição: a execução de uma máquina de Turing para a palavra w é a seqüência de configurações :

$$(S, \varepsilon, w) \vdash_M C_1 \vdash_M \dots \vdash_M C_K \vdash_M \dots$$

máxima tal que:

- ela seja infinita;
- ou ela atinge um estado de aceitação;
- ou ela atinge um estado, diferente de um estado de aceitação, para o qual não há transições.

Definição: uma linguagem L é decidida por uma máquina de Turing M se:

- M aceita L
- M não tem nenhuma execução infinita.

Logo, uma MT DECIDE uma linguagem L quando ela consegue aceitar todas as suas instâncias positivas e rejeitar todas as suas instâncias negativas. Logo tal MT é um procedimento efetivo para L .

Definição: uma linguagem é recursiva se ela é DECIDIDA por uma máquina de Turing.

Definição: uma linguagem é recursivamente enumerável se ela é ACEITA por uma máquina de Turing.

Logo, o conjunto das linguagens Recursivas está contido no conjunto das linguagens Recursivamente Enumeráveis (pois toda linguagem decidida por uma MT, também é aceita por ela).