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.

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

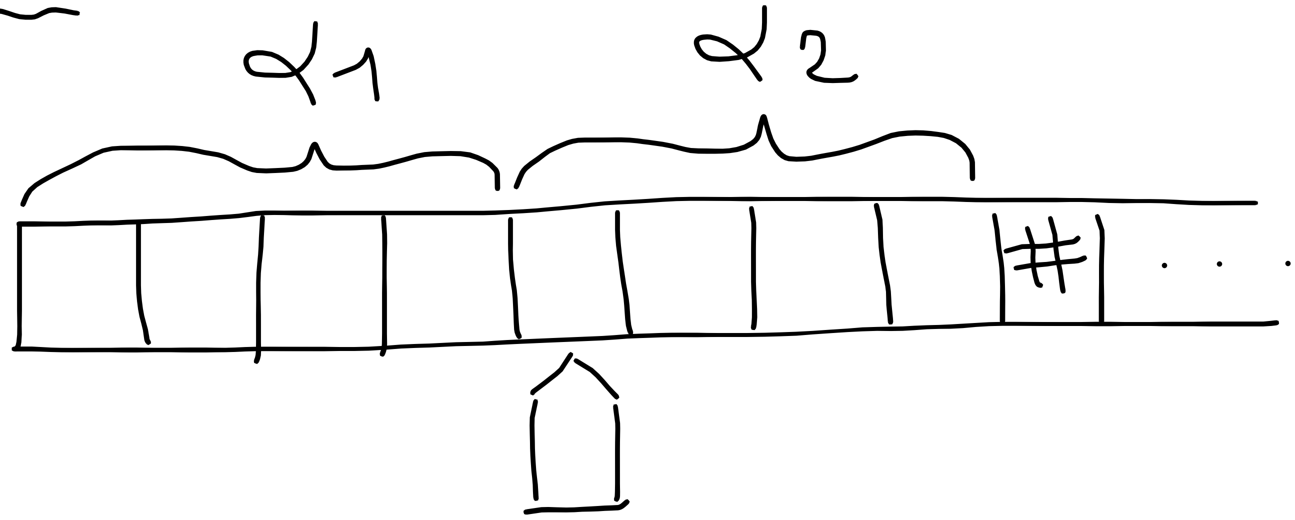
> Estando o autômato no estado q, com a cabeça apontando para o símbolo s, a transição pode ser disparada. Como resultado do disparo, o estado muda para q' ; o símbolo s da fita é substituído pelo símbolo b' ; e, finalmente, a cabeça se desloca para o sentido S (onde S é R - direita- , ou L - esquerda).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

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

Q x Γ^* x ($\varepsilon \cup \Gamma^*.(\Gamma-\{B\})$), ou seja, uma configuração é uma tripla: (q, α_1, α_2), onde :
$q \in Q$; $\alpha_1 \in \Gamma^*$; e

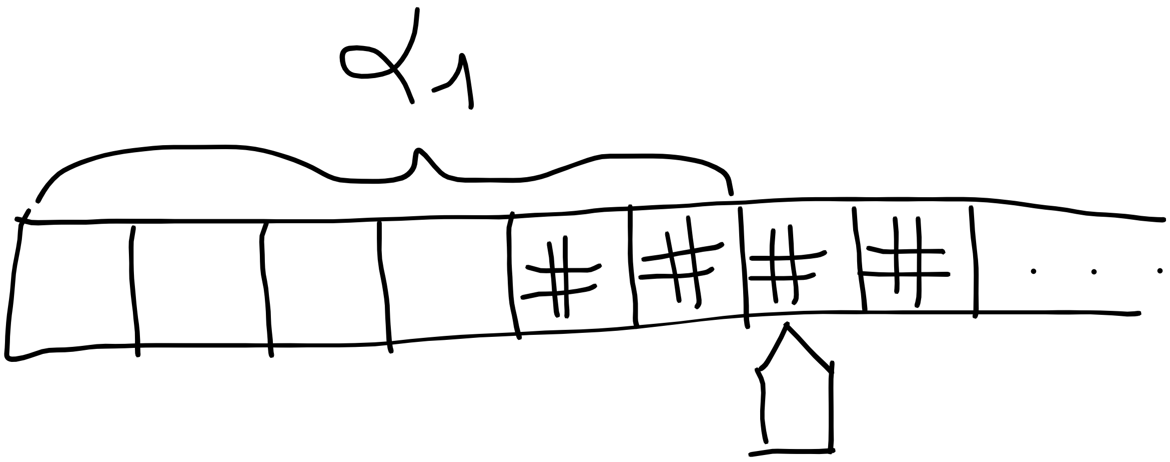
$\alpha_2 \in \varepsilon \cup \Gamma^*.(\Gamma-\{B\})$, ou seja, α_2 é a palavra que se encontra entre a cabeça (inclusive) e o último carácter não branco.
Informalmente, a configuração será dada pelo estado do autômato finito, pela palavra que aparece na fita e pela posição da cabeça de leitura.
Uma 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



$$\text{CONFIG} = (q, \alpha_1, \alpha_2)$$

Ex 2:



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

Seja a configuração :

(q, α_1, α_2),

escrita, como $(q, \alpha_1, b\alpha_2')$,

assumindo b = $\#$, no caso em que $\alpha_2 = \varepsilon$.

As configurações deriváveis em uma única etapa a partir de (q, α_1, α_2) são:

- Se $\delta(q, b) = (q', b', R)$:

  $(q, \alpha_1, b\alpha_2') \vdash_M (q', \alpha_1 b', \alpha_2')$.

- Se $\delta(q, b) = (q', b', L)$ e $\alpha_1 \neq \varepsilon$, i.e., $\alpha_1 = \alpha_1' a$:

  $(q, \alpha_1' a, b\alpha_2') \vdash_M (q', \alpha_1', a b' \alpha_2')$.

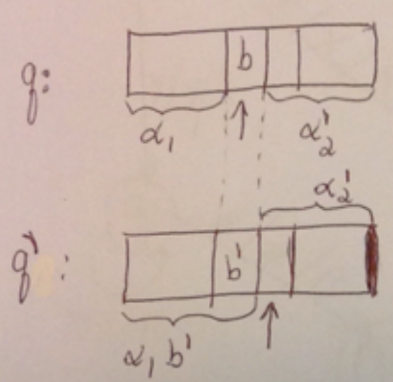
Obs: é impossível transição à esquerda se $\alpha_1 = \varepsilon$.

DERIVAÇÃO EM UM ÚNICO PASSO NAS MT:

<u>Ilustração da definição</u>

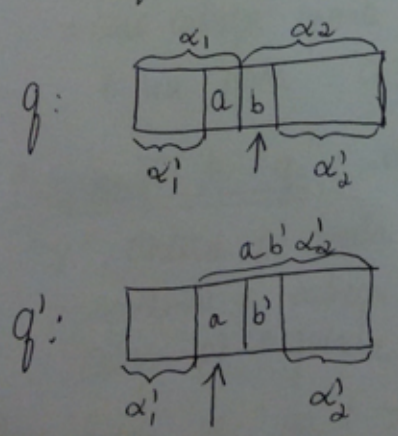
- Se $\delta(q,b) = (q', b', R)$:

$$(q, \alpha_1, b\alpha_2') \vdash_M (q', \alpha_1 b', \alpha_2')$$



- Se $\delta(q,b) = (q', b', L)$; $\alpha_1 = \alpha_1' a$ (i.e., $\alpha_1 \neq \epsilon$):

$$(q, \alpha_1' a, b\alpha_2') \vdash_M (q', \alpha_1', ab'\alpha_2')$$



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 C0, C1, ... , Ck tais que:

. C = C0
. C' = Ck
. Ci |-M Ci+1 para $0 \leq i < k$

Uma palavra é então aceita por uma máquina de Turing se a execução da máquina leva 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 :

$$(s, \varepsilon, w) \vdash_M^* (p, \alpha_1, \alpha_2)$$

com $\quad p \in F$

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 $\Sigma = \{a , b\}$

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 :

$$(s, \varepsilon, w) \vdash_M C_1 \vdash_M \cdots \vdash_M C_k \vdash_M \cdots$$

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).

5.4 Tese de Church-Turing

Tese (hipótese): as linguagem reconhecidas por um procedimento efetivo são as linguagens decididas por uma máquina de Turing.

Uma tese não é um teorema. Ela não pode ser provada!

De acordo com a TESE, qualquer programa que pode ser executada num computador pode ser representado então por uma máquina de Turing

O estudo da Tese de de Church-Turing será apresentado em detalho na disciplina de Teoria da Computação...