Capítulo 9

Máquinas de Turing

9.1. A máquina de Turing (TM) padrão

9.2. Combinações de máquinas de Turing

9.3. A Tese de Turing

- Linguagens regulares $\Rightarrow$ Autómatos finitos

- Linguagens livres de contexto $\Rightarrow$ Autómatos finitos com uma pilha

- Linguagens não livres de contexto $\Rightarrow$???

- Qual o autómato mais poderoso ?

- Quais os limites da computação ?

1940 – Alan Turing, procura formalizar a noção de algoritmo, identificando as operações fundamentais e primitivas.

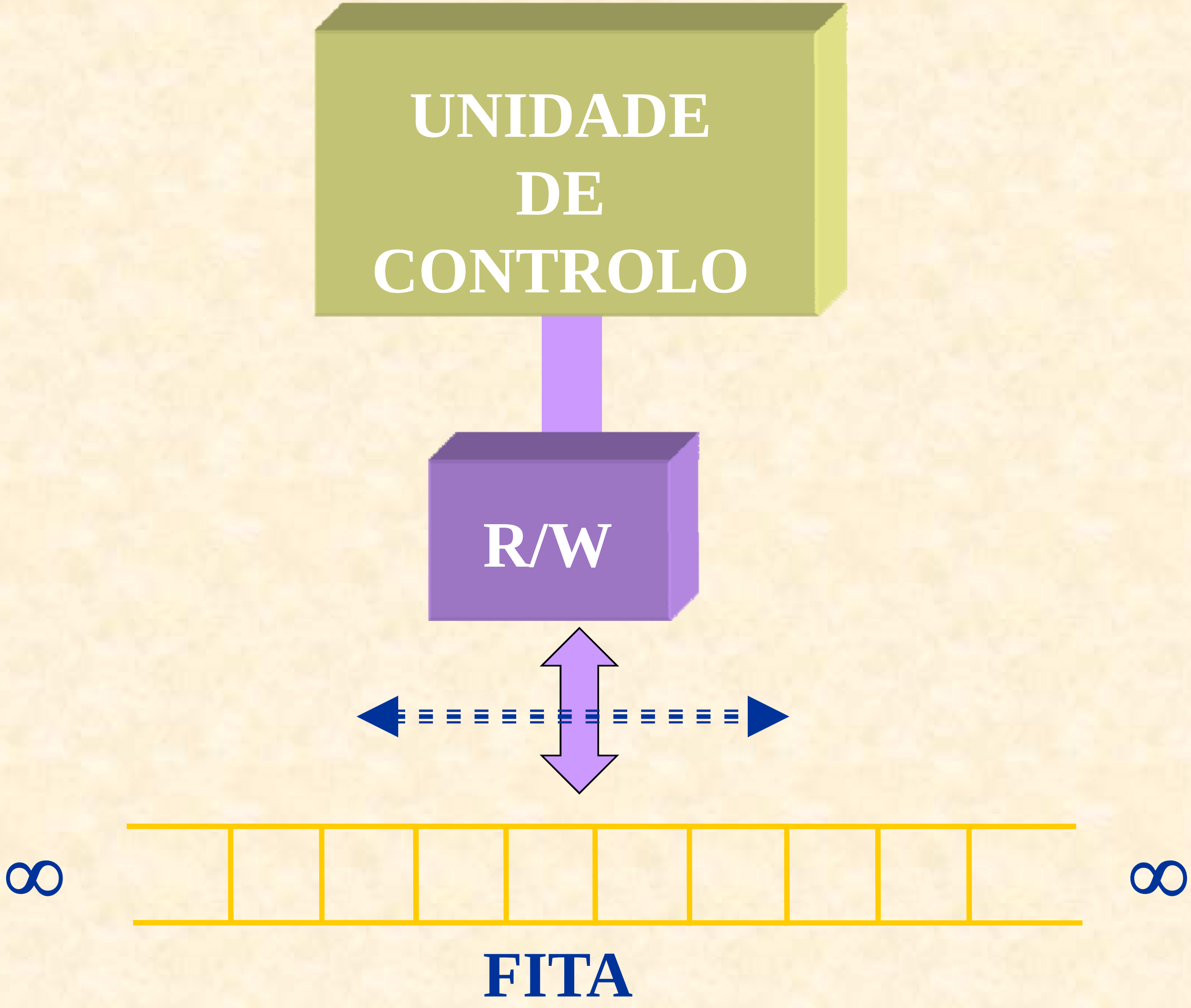
Depois define uma máquina abstracta capaz de executar essas operações segundo regras bem definidas.

Máquina de Turing

Um modelo abstracto de computação

Problemas resolúveis e irresolúveis

MT



Definição 9.1. MT

$$M = (Q , \Sigma, \Gamma, \delta , q_0 , \Box , F)$$

Q é o conjunto de estados internos

Σ é o **alfabeto de entrada**

Γ é o conjunto finito **alfabeto da fita**

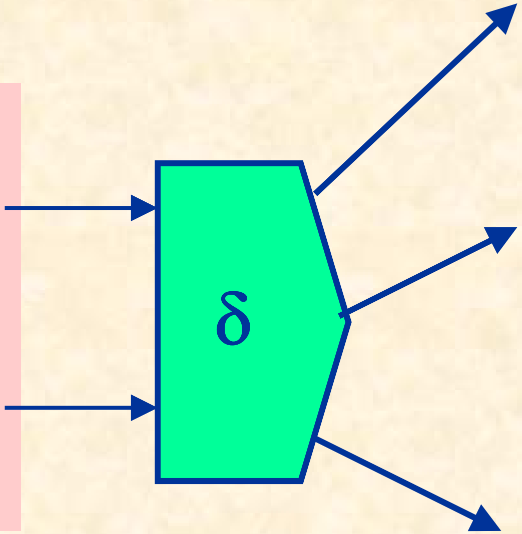
δ é a função de transição

$\Box \in \Gamma$ é o carácter **branco**, símbolo especial

$F \subseteq Q$ é o conjunto de estados finais

Função de transição

- estado actual UC
- carácter da fita a ser lido no momento



- 1º - Q , próximo estado da UC
- 2º - escrever na fita um novo símbolo em substituição do presente
- 3º - fazer uma movida para a direita ou para a esquerda

R/W

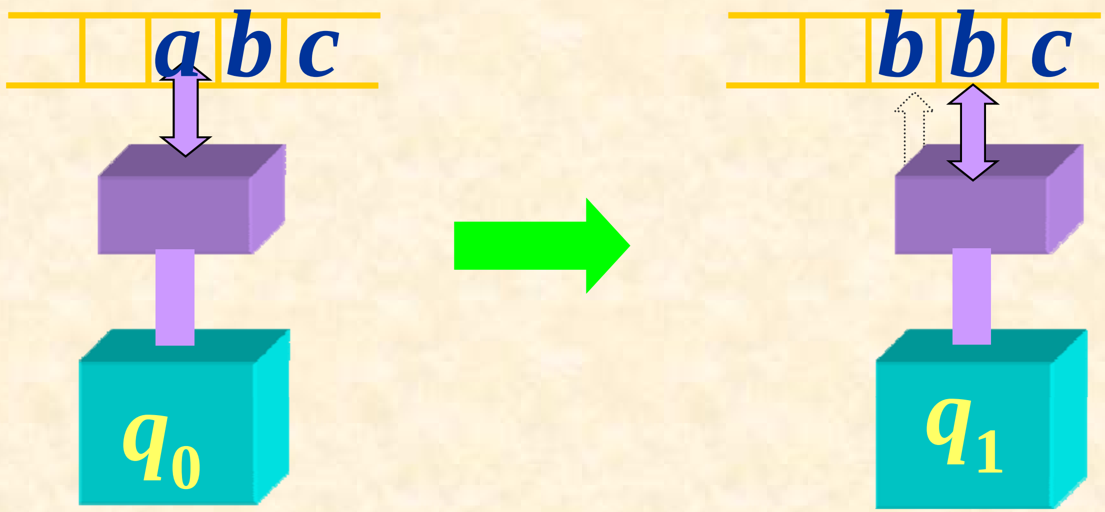
$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$$

Exemplo 1 $\Sigma = \{ a, b, c \}$

$\Gamma = \{ a, b,$ □ □$\}$

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

$$\delta (q_0, a) = (q_1, b, R)$$



Uma M Turing pode ser encarada como um computador muito simples:

- tem uma unidade de processamento com memória finita (número finito de estados)

- tem uma segunda unidade de armazenamento de capacidade infinita.

- a função de transição é o "programa" do computador ; é uma função parcial.

Exemplo 2. Seja a MT

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

$\Gamma = \{ a, b, \square \}$

$Q = \{q_0, q_1\}$

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

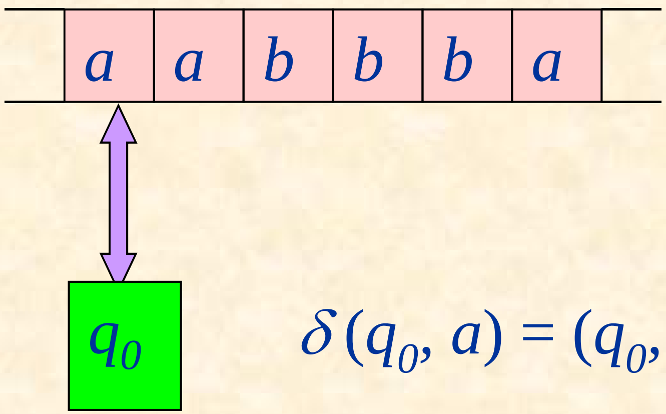
$\Gamma = \{a, b, \square\}$

$F = \{q_1\}$

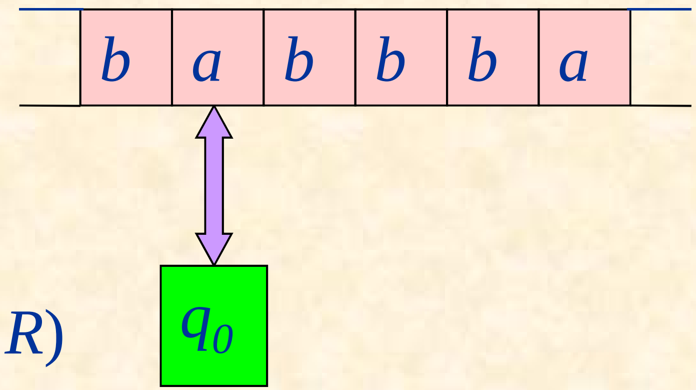
$\delta(q_0, a) = (q_0, b, R)$

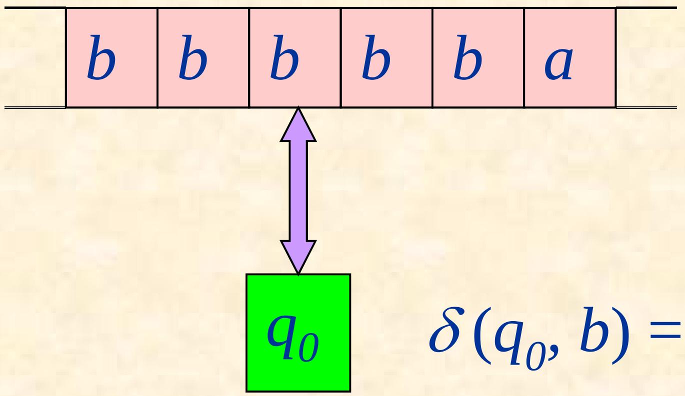
$\delta(q_0, b) = (q_0, a, R)$

$\delta(q_0, \square) = (q_0, \square, R)$

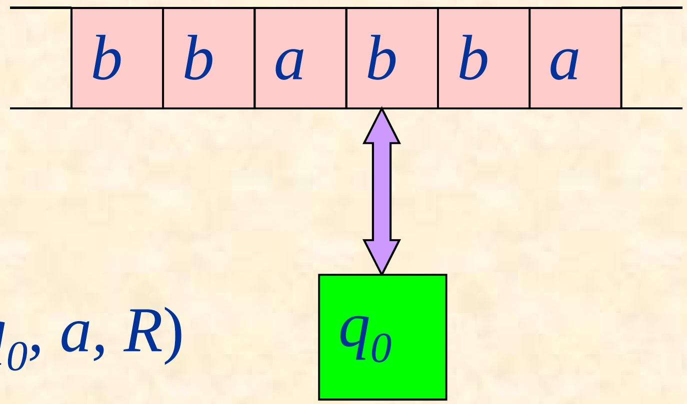


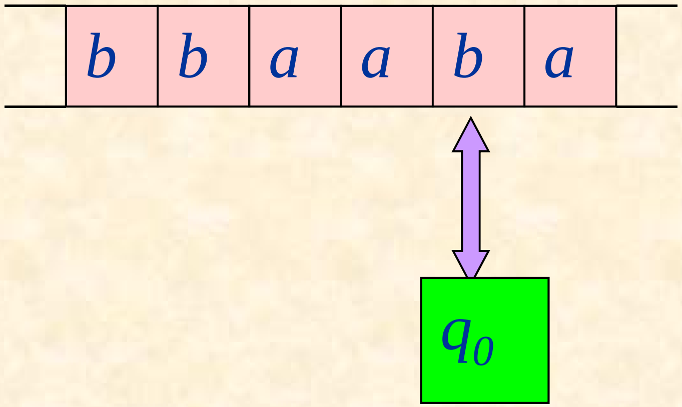
$$\delta(q_0, a) = (q_0, b, R)$$



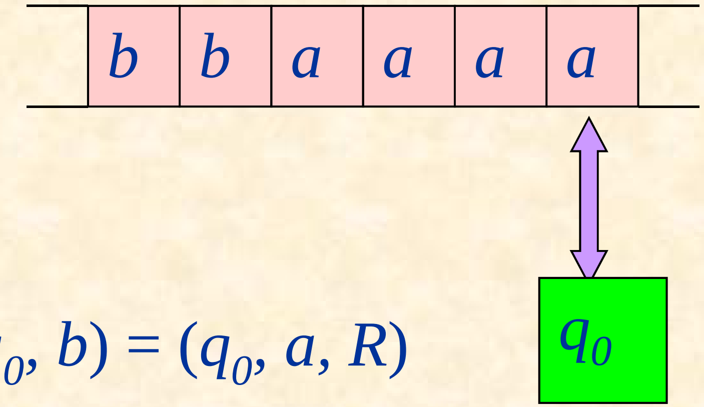


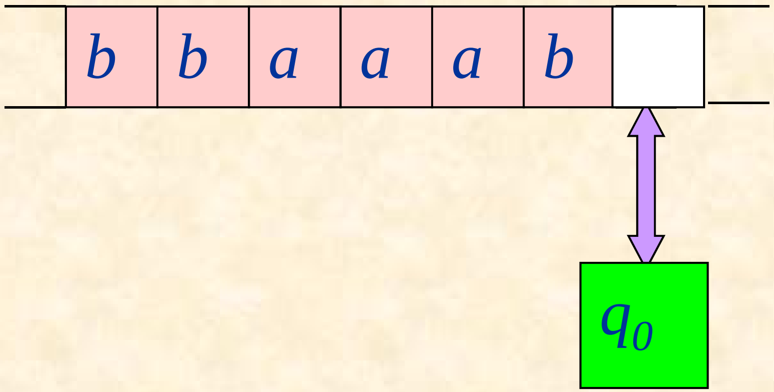
$$\delta(q_0, b) = (q_0, a, R)$$



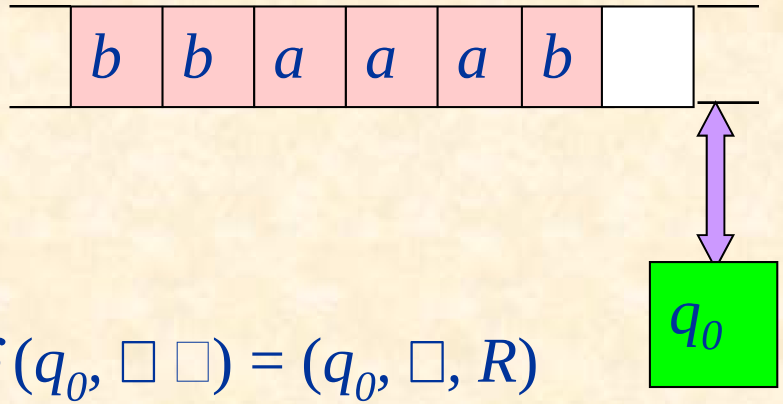


$\delta(q_0, b) = (q_0, a, R)$





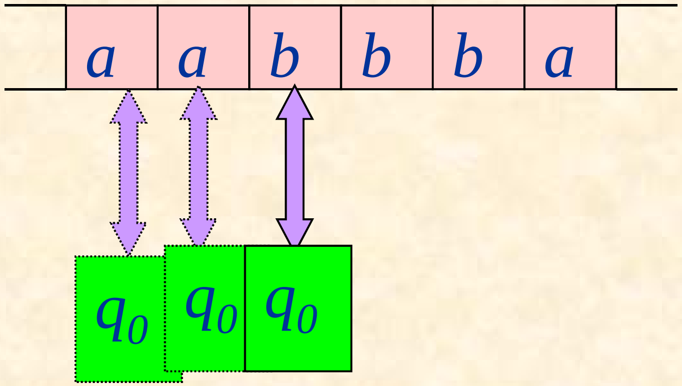
$\delta(q_0, \square\ \square) = (q_0, \square, R)$



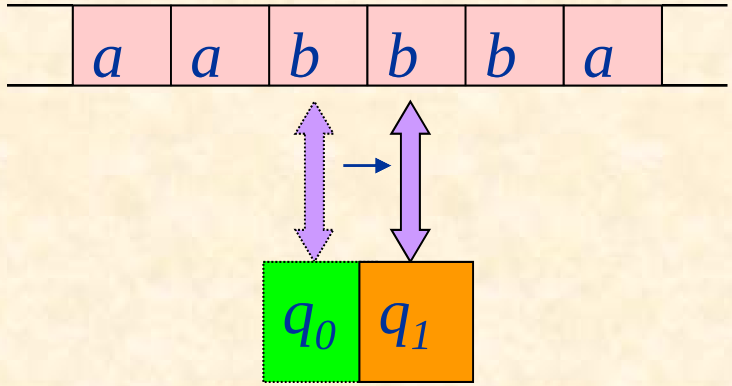
Se a função de transição for

$$\delta(q_0, a) = (q_0, a, R)$$
$$\delta(q_0, b) = (q_1, b, R)$$
$$\delta(q_0, \square) = (q_0, \square, R)$$
$$\delta(q_1, a) = (q_0, b, L)$$
$$\delta(q_1, b) = (q_0, b, L)$$
$$\delta(q_1, \square) = (q_0, \square, L)$$

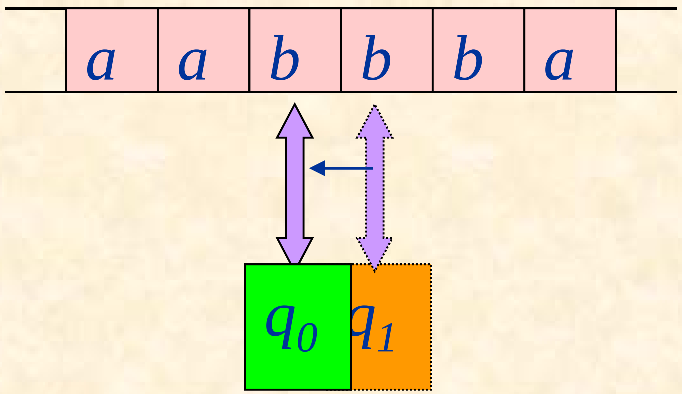
$\delta(q_0, a) = (q_0, a, R)$



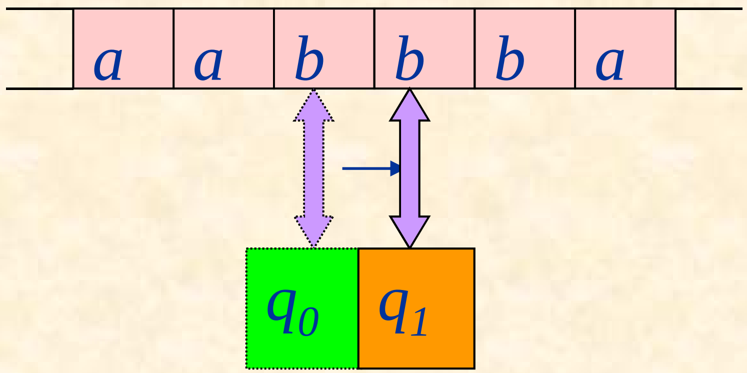
$\delta(q_0, b) = (q_1, b, R)$



$\delta(q_1, b) = (q_0, b, L)$



$\delta(q_0, b) = (q_1, b, R)$



Características da MT padrão

- Uma fita ilimitada em ambas as direcções permitindo um número arbitrário de movidas à esquerda e à direita.

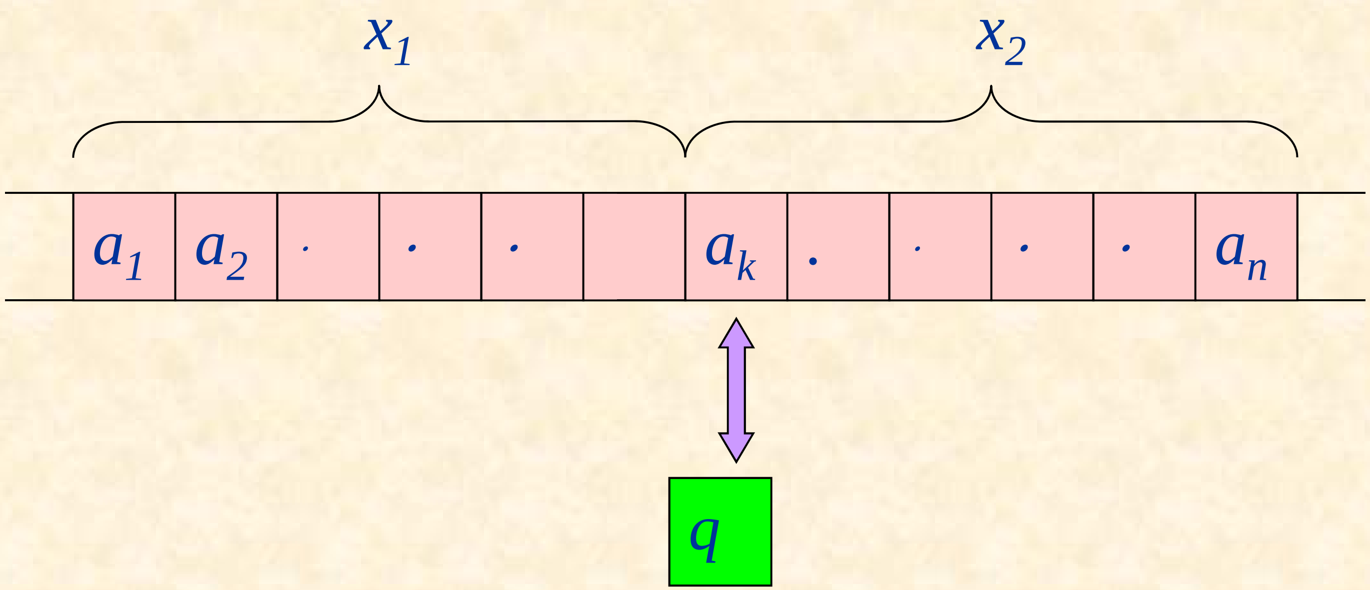
- É determinística na medida em que δ define no máximo uma movida para cada configuração.

- No instante inicial a fita tem algum conteúdo especificado. Parte deste pode ser considerado a entrada. Sempre que a máquina pare (***halt***), algum do conteúdo da fita pode ser considerado como saída.

descrição instantânea da MT

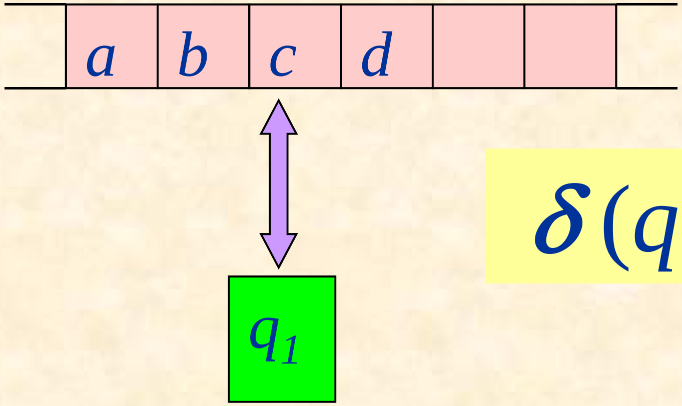
- o estado actual da unidade de controlo
- conteúdo da fita
- a posição da cabeça de leitura/escrita

$$x_1\, q\, x_2$$

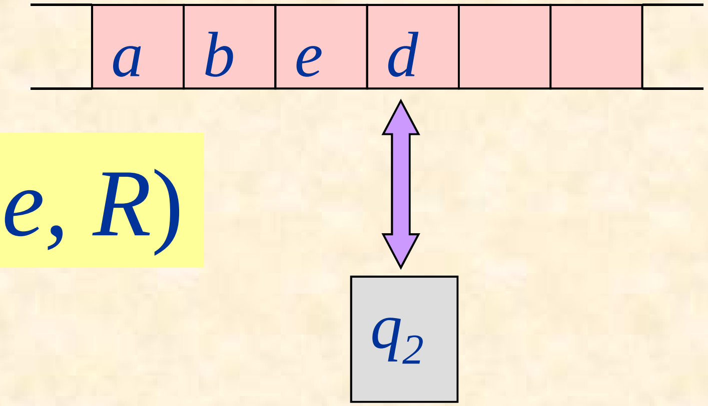


movidas de uma MT

configuração (k) $\vdash$ configuração (k+1)



$$\delta(q_1, c) = (q_2, e, R)$$



$$abq_1cd \vdash abeq_2d$$

movida

$$a_1 a_2 \ldots\ldots a_{k-1} q_1 a_k a_{k+1} \ldots a_n \vdash a_1 a_2 \ldots\ldots a_{k-1} b q_2 \, a_{k+1} \ldots a_n$$

$$\Updownarrow$$

$$\delta(q_1, a_k) = (q_2, b, R)$$

movida

$$a_1 a_2 \ldots\ldots a_{k-1} q_1 a_k a_{k+1} \ldots a_n \vdash a_1 a_2 \ldots\ldots q_2 a_{k-1} b a_{k+1} \ldots a_n$$

$$\Updownarrow$$

$$\delta(q_1, a_k) = (q_2, b, L)$$

... um número arbitrário de movidas

$$\vdash^{*}$$

... de uma certa máquina *M*

$$\vdash^{*}_{M}$$

Computação

M parou partindo de alguma configuração inicial $x_1 q_i x_2$

se $$x_1 q_i x_2 \overset{*}{\vdash} y_1 q_j a y_2$$

para algum q_j e algum a
para os quais $\delta(q_j, a)$ seja **não definida.**

Chama-se **computação** à sequência das configurações que levam a MT do estado inicial ao estado de paragem (**halt**)

Ciclo infinito

A situação em que a MT entra num ciclo infinito, nunca parando, nem de lá podendo sair, denota-se por

$$x_1 q\, x_2 \overset{*}{\vdash} \infty$$

Esta situação é muito importante na teoria das MT

Função de transição segundo :

Linz e JFLAP

$$\delta : (Q \times \Gamma) \rightarrow (Q \times \Gamma \times \{ L, R \}) \qquad \delta(q_0, b) = (q_1, b, R)$$

Taylor e *DEM*

$$\delta : (Q \times \Gamma) \rightarrow (\Gamma \cup \{L, R\}) \times Q \qquad \delta(q_0, b) = (b, q_1)$$

$$\delta(q_1, b) = (R, q_1)$$

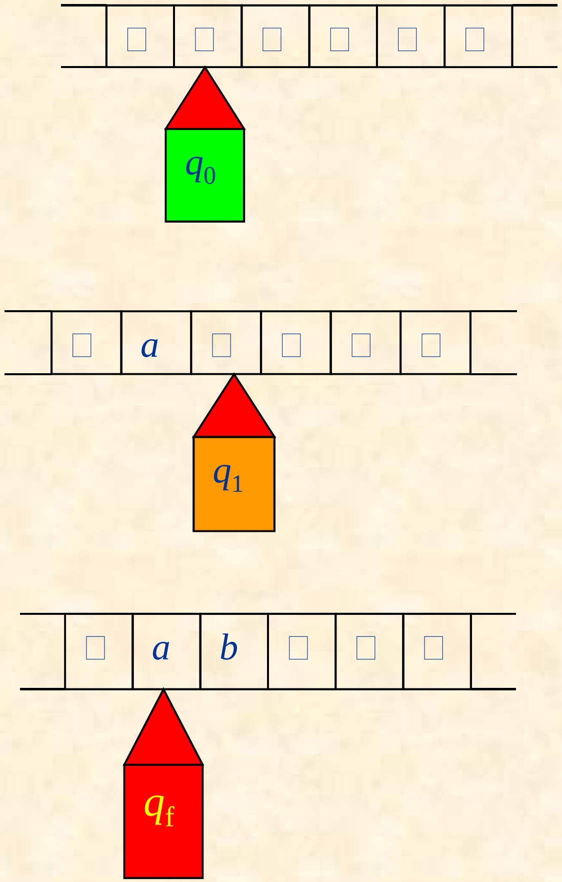
São necessárias mais instruções para a mesma funcionalidade.

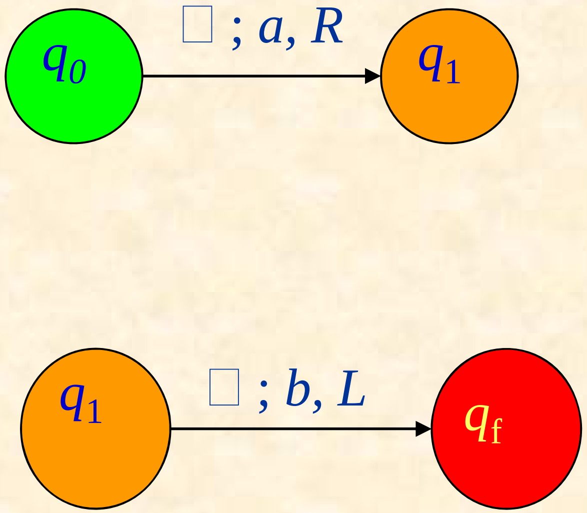
9.1.4. Diagramas de estados ou de transição de MT's

Os diagramas de estados são uma forma alternativa à escrita da função de transição

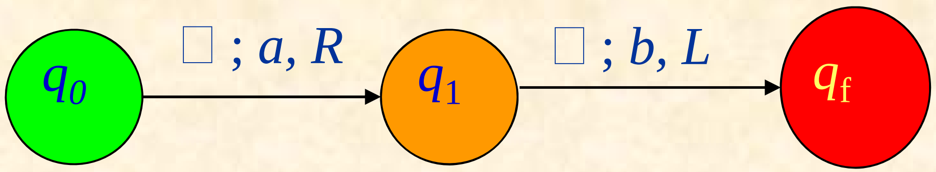
Consideremos a MT, com uma fita completamente em branco, que faz o seguinte:

- inicializada lendo uma casa em branco,
- escreve os caracteres *ab* e
- pára lendo o carácter *a*.

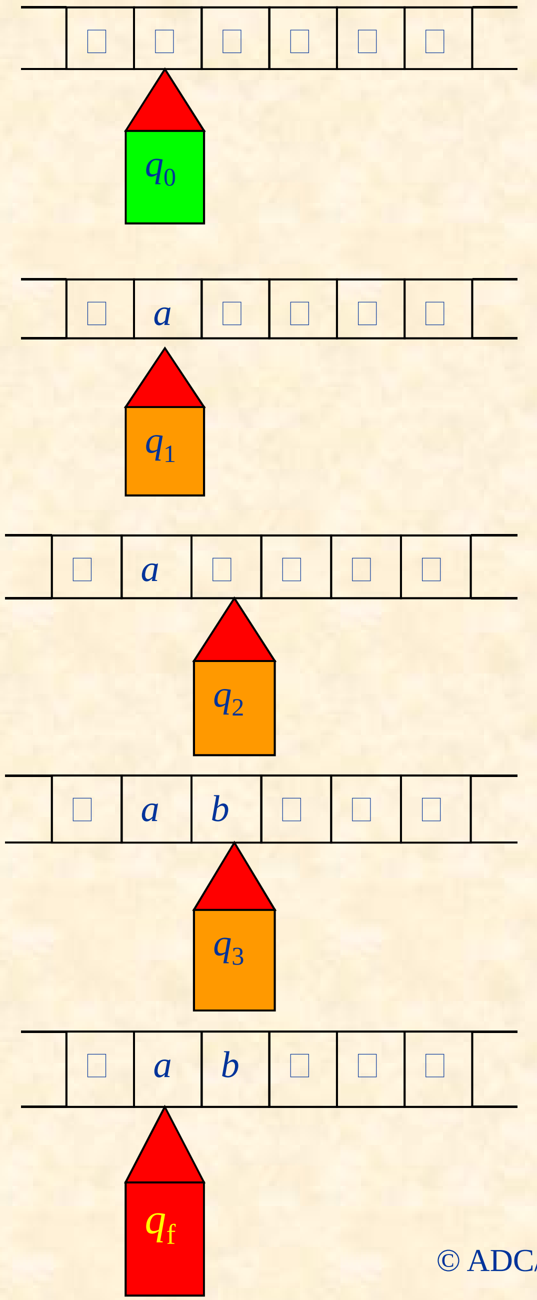


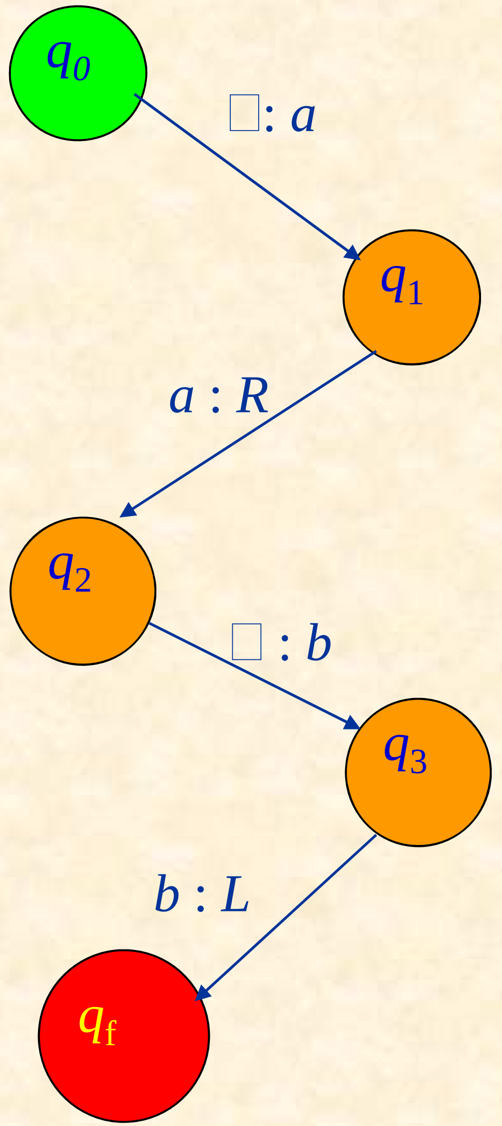


JFLAP

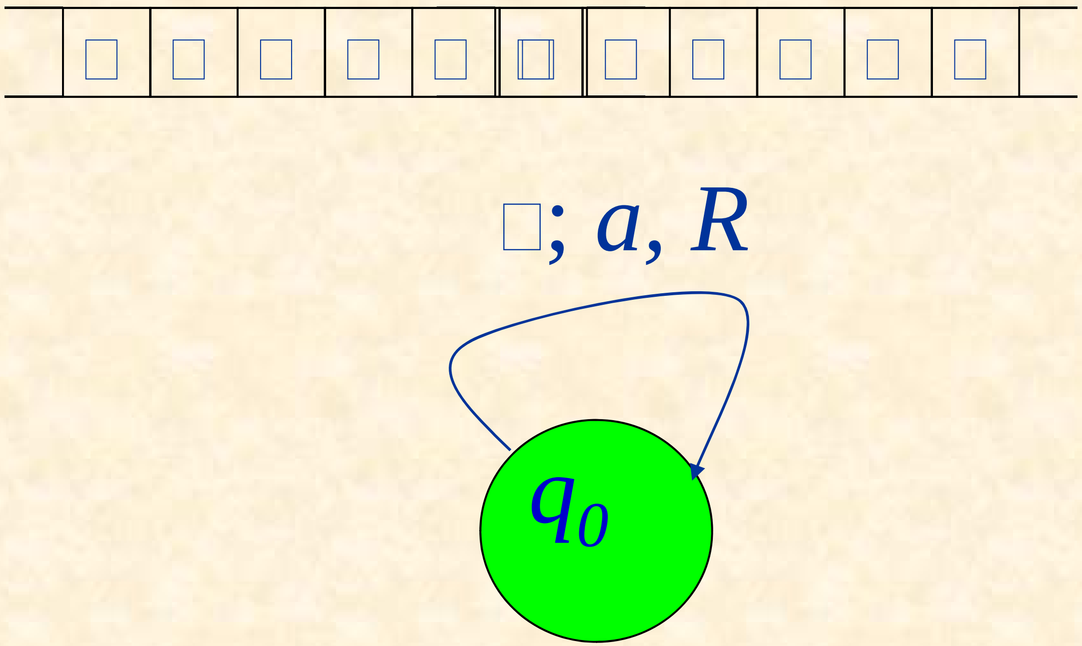


DEM





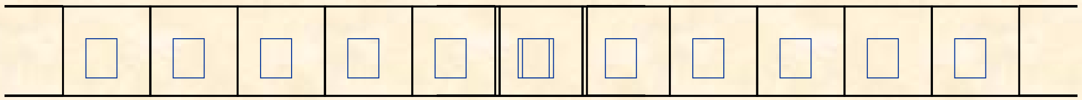
Que faz ???



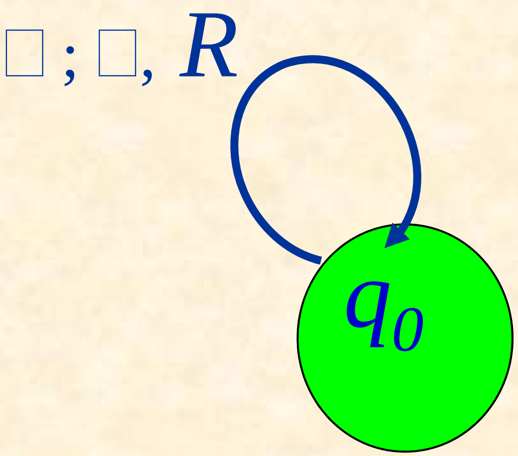
Fita inicial

JFLAP

Que faz ???

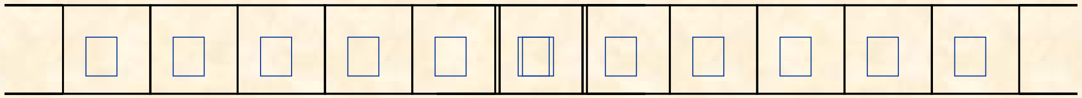


Fita inicial

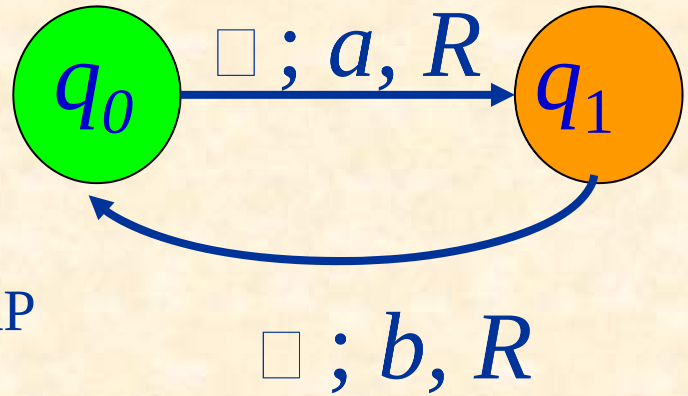


JFLAP

E esta ???



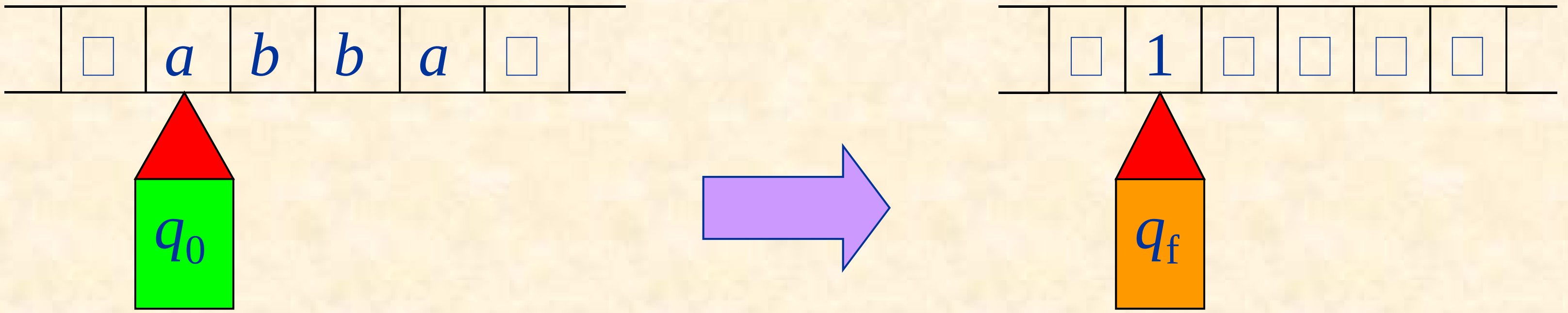
Fita inicial



JFLAP

Diagrama de estados da MT $n_a(w) = n_b(w)$ (aceitador)

- aceita as cadeias em $\{a, b\}^+$ com um número de *a*'s igual ao nº de *b*'s em qualquer ordem
- inicializada no carácter mais à esquerda de *w*, pára lendo um simples 1 estando toda a restante fita em branco

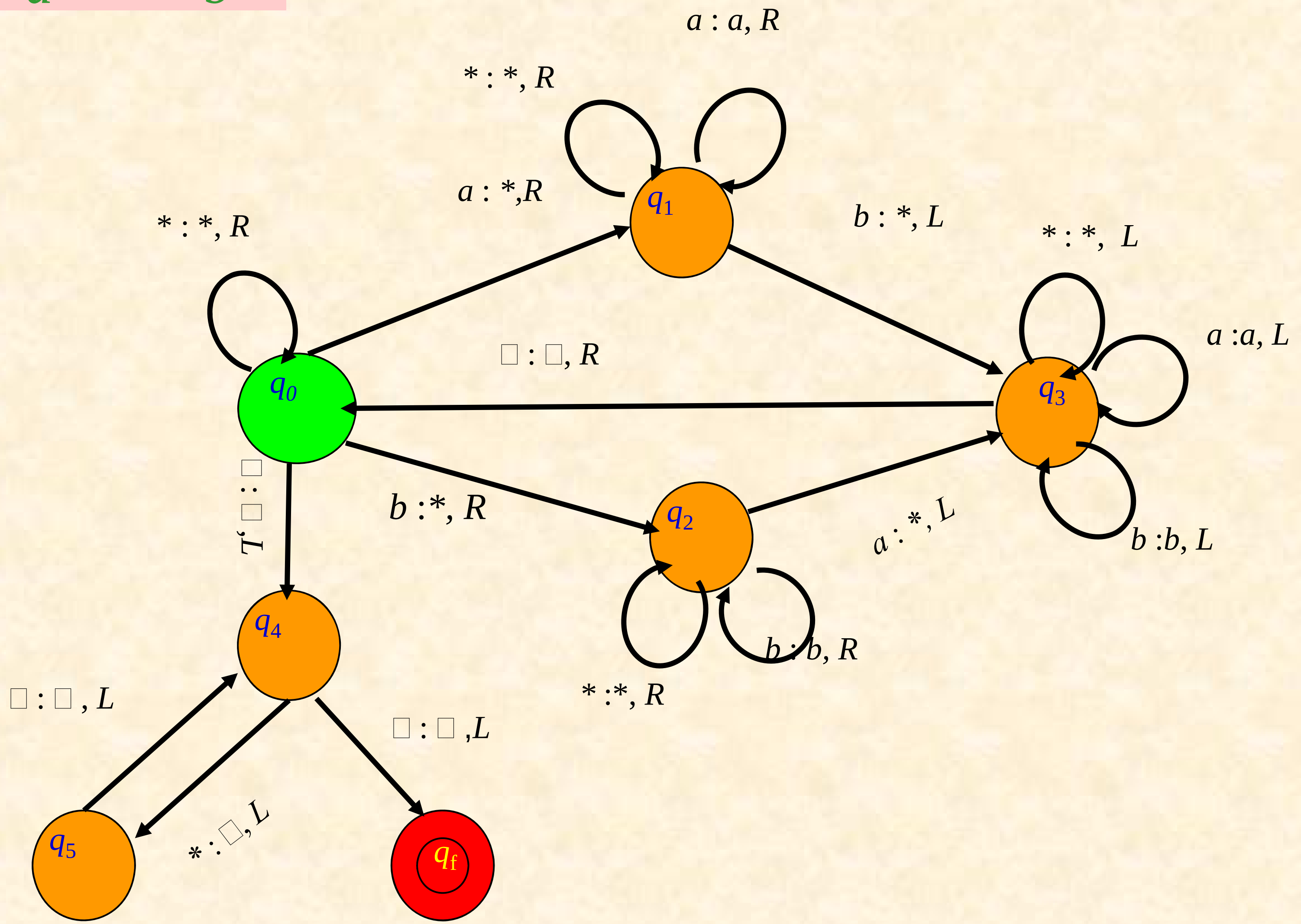


Algoritmo possível:

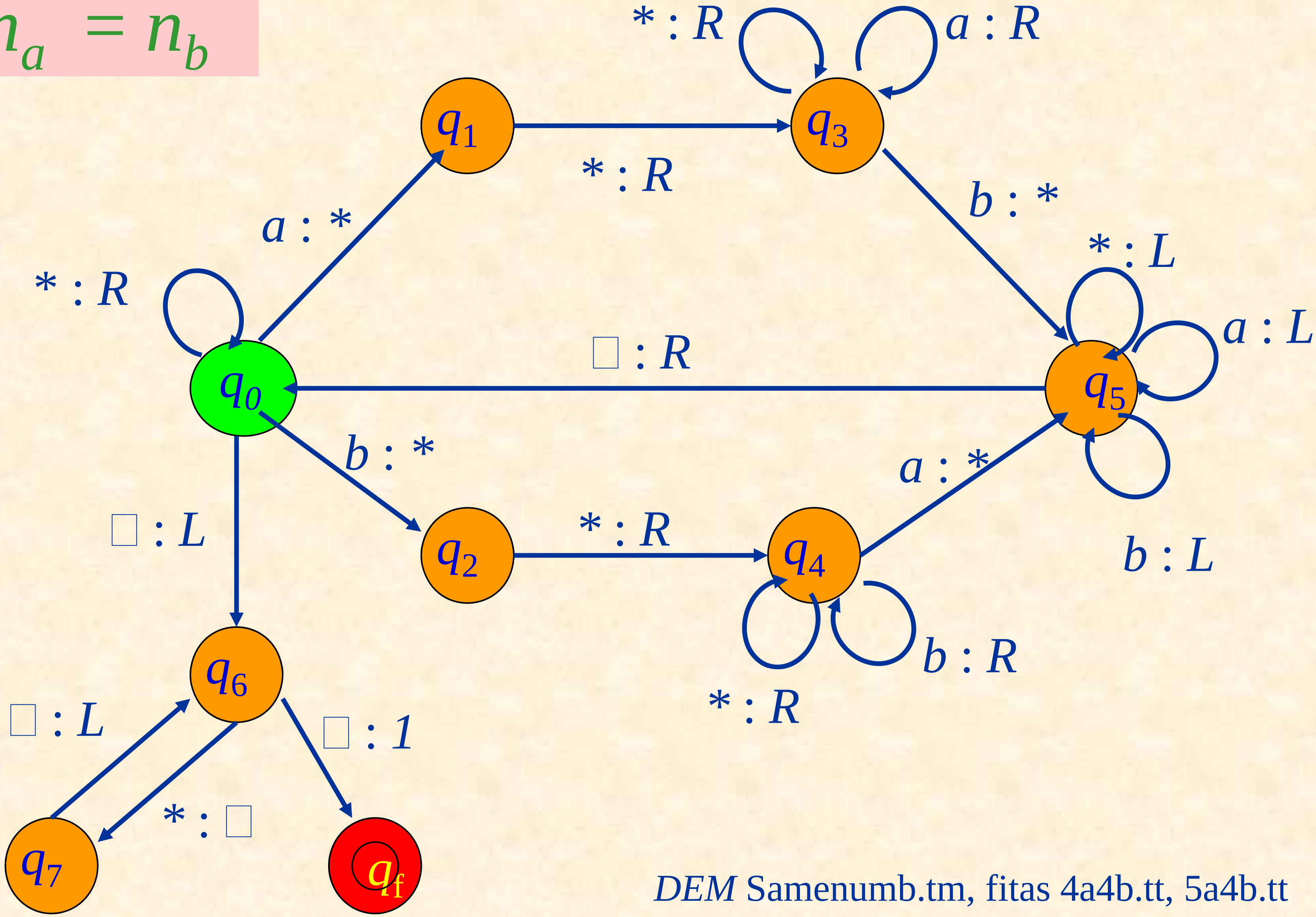
Eliminar pares *a*-*b* até que:

- ou não reste nenhum *a* nem nenhum *b*, caso em que a cadeia é aceite.
- ou restarem alguns caracteres não- emparelháveis, caso em que a cadeia não é aceite.

$n_a = n_b$



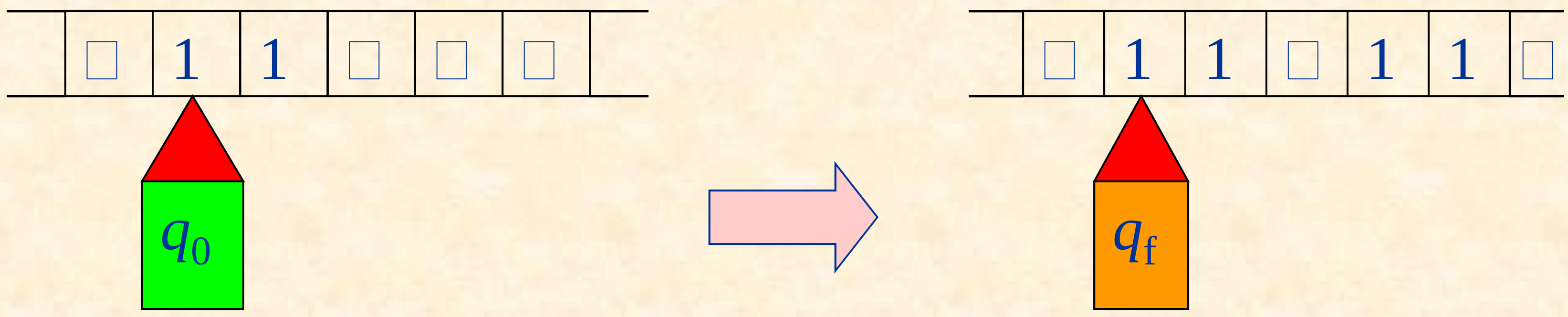
$n_a = n_b$



DEM Samenumb.tm, fitas 4a4b.tt, 5a4b.tt

Exemplo: a MT copiadora (transdutor).

- Inicia-se no primeiro 1 de uma cadeia de *n* 1's.
- Pára no primeiro 1 de uma cadeia ininterrupta de *n* 1's, seguida por um branco, seguido por uma cadeia ininterrupta de n 1's.



Copying Machine.tm in *DEM*, fita 8.tt

9.1.2. MT como aceitadores de linguagens.

- dada uma cadeia *w* do alfabeto de entrada, escrita na fita, **não contendo nem brancos nem λ** ,
- antes de *w* estão brancos e depois de *w* brancos estão ,
- a MT inicializa-se em q_0 com a cabeça de W/R posicionada no carácter mais à esquerda de *w*,
- dá-se uma sequência de movidas ,
- *w* é aceite se a MT entrar num estado final e parar (*halt*),
- *w* não é aceite se a MT parar (*halt*) num estado não final, ou se entrar num ciclo infinito.

Definição 9.3

Seja $M = (Q, \Sigma, \Gamma, \delta, q_0, \square , F)$ uma MT.
A linguagem aceite por M é

$$L(M) = \{ w \in \Sigma^+ : q_0 w \overset{*}{\vdash} x_1 q_f x_2$$

$$\text{para algum } q_f \in F \text{ e } x_1, x_2 \in \Gamma^* \}$$

Se a cadeia tivesse brancos, a MT não saberia quando ela terminaria e teria que continuar a percorrer a fita até ao infinito.

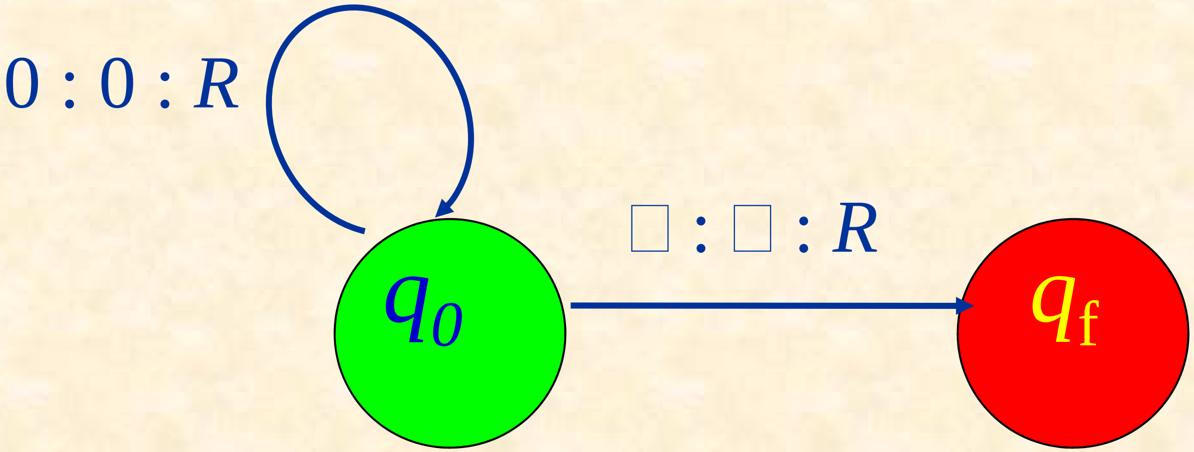
Exemplo $\Sigma = \{0, 1\}$

$L = L(00^*)$, todas as cadeias com um ou mais zeros (sem 1's).

- ler o primeiro carácter. É zero ? Se sim continuar; se não, parar num estado não final.
- ler o segundo carácter. É zero ? Se sim continuar; se não, parar num estado não final.
- ... ...
- encontrou um branco sem parar num estado não final ? Se sim, parar num estado final.

$$\delta(q_0, 0) = (q_0, 0, R)$$

$$\delta(q_0, \square) = (q_f, \square, R)$$



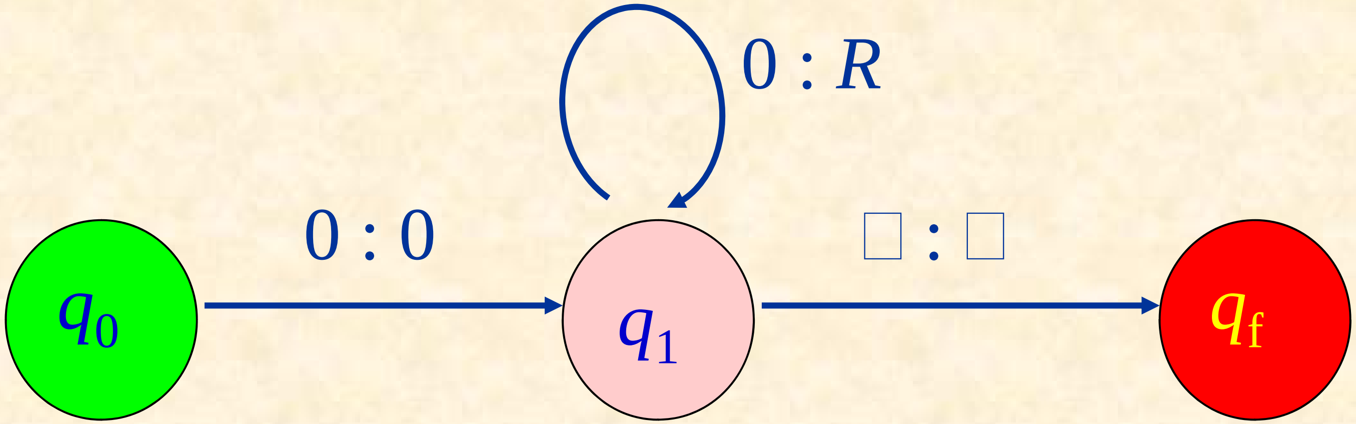
JFLAP

Seguindo Taylor e *DEM* teremos que criar mais estados:

$$\delta(q_0, 0) = (q_1, 0)$$

$$\delta(q_1, 0) = (q_1, R)$$

$$\delta(q_1, \square\ \square) = (q_f, \square)$$



DEM

- Note-se que δ é uma função parcial

- Não é definida por exemplo $\delta(q_0, 1)$

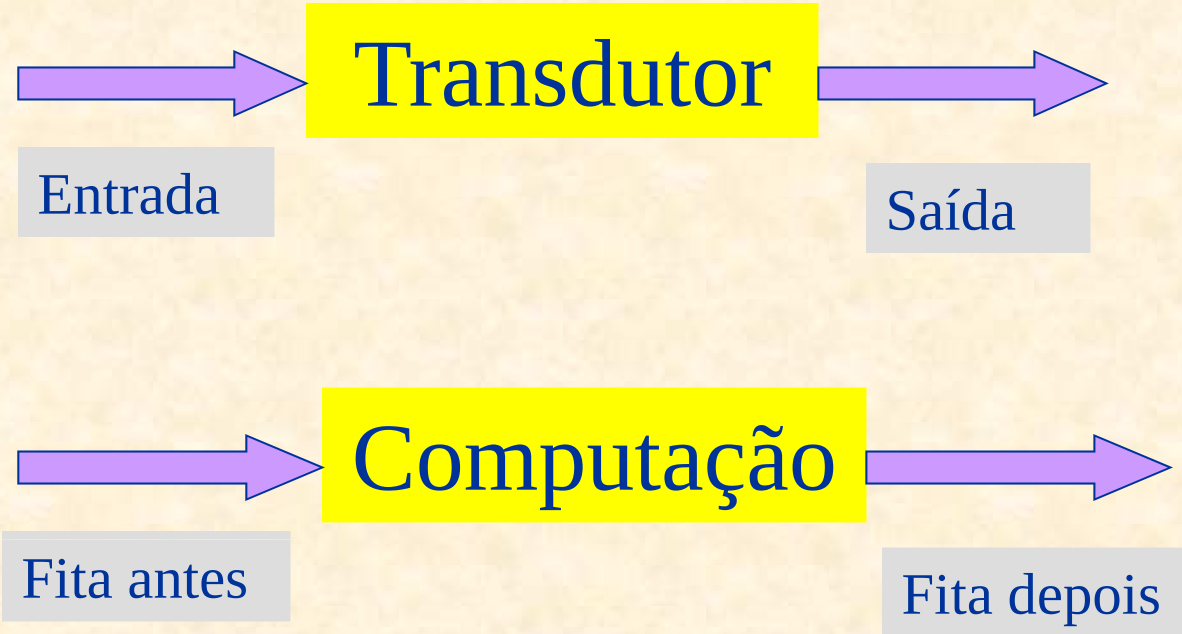
- Se aparece um 1, no estado q_0, a MT pára (halt) **para sempre**, indicando que w não é aceite (a menos que q_0 fosse um estado final)

- As MT podem reconhecer algumas linguagens que não são livres de contexto.

- São máquinas mais poderosas do que os autómatos estudados anteriormente.

9.1.3. TM como transdutores

um transdutor: transforma uma entrada numa saída



MT M implementa uma função f

$$\hat{w} = f(w),$$

se

$$q_0 w \overset{*}{\vdash}_M q_f \, f(w)$$

sendo q_f algum estado final.

Definição 9.4. Função computável

f com o domínio D diz-se **Turing – computável** ou simplesmente **computável** se existir alguma MT tal que para toda a $w \in D$:

$$q_0 w \overset{*}{\vdash}_M q_f\, f(w), \quad q_f \in F$$

Todas as funções matemáticas usuais, por mais complicadas que sejam, são Turing-computáveis.

Exemplo. Adição de números inteiros (representação unária),
ex. 9.9 Linz.

$$\overbrace{1\;1\;1\;1\;1}^{x\,=\,5}\;0\;\overbrace{1\;1\;1\;1}^{x\,=\,4}\;\square\;\square$$

$$\overbrace{1\;1\;1\;1\;1\;1\;1\;1\;1}^{x\,+\,y\,=\,9}\;\square\;\square\;\square$$

Outros exemplos de Máquinas de Turing

Função	Ficheiro do *DEM*
Conversor de binário em unário	Convert.tm
Factorial, n !	Factoria.tm
Multiplicação, $f(n,m) = nm$	Multiply.tm
Potência n^n	N^n.tm
Raiz quadrada (de quadrados perfeitos)	Sqrt.tm
Divisão por 2	div2single.tm
Reversão de uma cadeia em $\{a,b^*\}$	Reverse Word.tm

9.2. Combinações de máquinas de Turing para tarefas complicadas

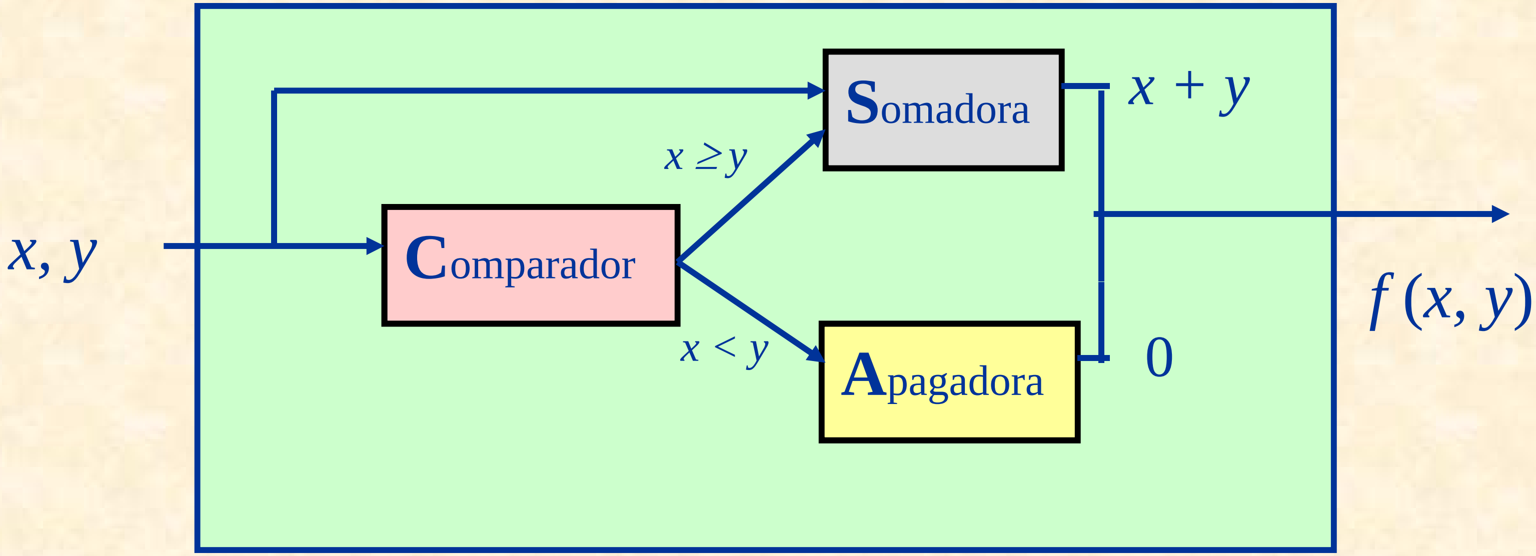
- Muitas funções complicadas podem-se decompor em sequências de funções simples computáveis

- Tal conceito também se pode aplicar ao nível da Turing - computabilidade.

- Pode-se projectar uma máquina de Turing composta por combinações de máquinas de Turing elementares a fim de realizar operações mais complexas.

Exemplo

$$f(x, y) = \begin{cases} x + y\,, & \text{se } x \geq y \\ 0, & \text{se } \; x < y \end{cases}$$



Combinações de máquinas de Turing no *DEM*

Um nó pode ser uma TM. Por exemplo *Convert.tm* e *N^n.tm*, que tem três sub-máquinas de Turing. Para as ver, no menu **View,** seleccionar *Submachines.*

Através de combinações, é possível computar com MT's operações matemáticas complexas.

9.3. A Tese de Turing (1930’s) (conjectura)

Qualquer computação que possa ser implementada por processos mecânicos (i.e., por uma máquina) pode também ser implementada por uma máquina de Turing

Alguns argumentos a favor da tese de Turing:

- Qualquer computação que possa ser feita por qualquer computador digital existente também pode ser feita por uma TM.

- Ninguém conseguiu ainda encontrar um problema, resolúvel por um qualquer algoritmo, para o qual não possa ser escrito um programa para uma TM.

- Foram propostos modelos alternativos para a computação TM, mas nenhum deles é mais poderoso do que a TM.

A Tese de Turing é (ainda) uma lei

básica das Ciências da Computação !

Definição 9.5. Algoritmo

Um algoritmo para uma função $f : D \to R$ é uma Máquina de Turing M tal que,

- dado como entrada um qualquer $d \in D$ na sua fita
- pára *(halts)* com uma resposta correcta para $f(d) \in R$ na sua fita

para todo o $d \in D$,

$$q_0 d \overset{*}{\vdash}_M q_f\, f(d), \quad q_f \in F$$

Bibliografia

An Introduction to Formal Languages and Automata, Peter Linz, 3rd Ed., Jones and Bartelett Computer Science, 2001.

Models of Computation and Formal Languages, R. Gregory Taylor, Oxford University Press, 1998.

Introduction to Automata Theory, Languages and Computation, 2nd Ed., John Hopcroft, Rajeev Motwani, Jeffrey Ullman, Addison Wesley, 2001.

Elements for the Theory of Computation, Harry Lewis and Christos Papadimitriou, 2nd Ed., Prentice Hall, 1998.

Introduction th the Theory of Computation, Michael Sipser, PWS Publishing Co, 1997.

http://www.turing.org.uk/turing/Turing.html (Alan Turing Home Page)

http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Turing.html