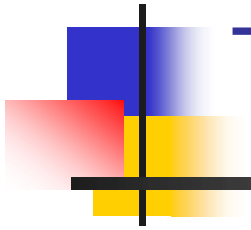


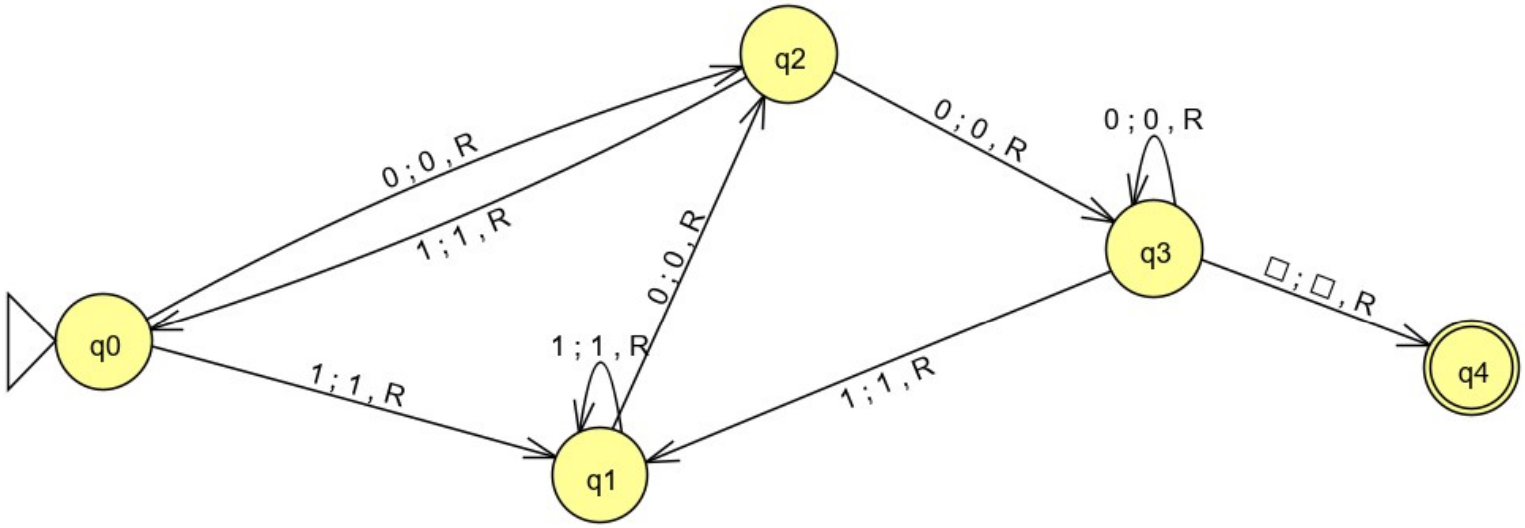
Teoria da Computação

Unidade 3 – Máquinas Universais

Referência – Teoria da Computação (Divério, 2000)

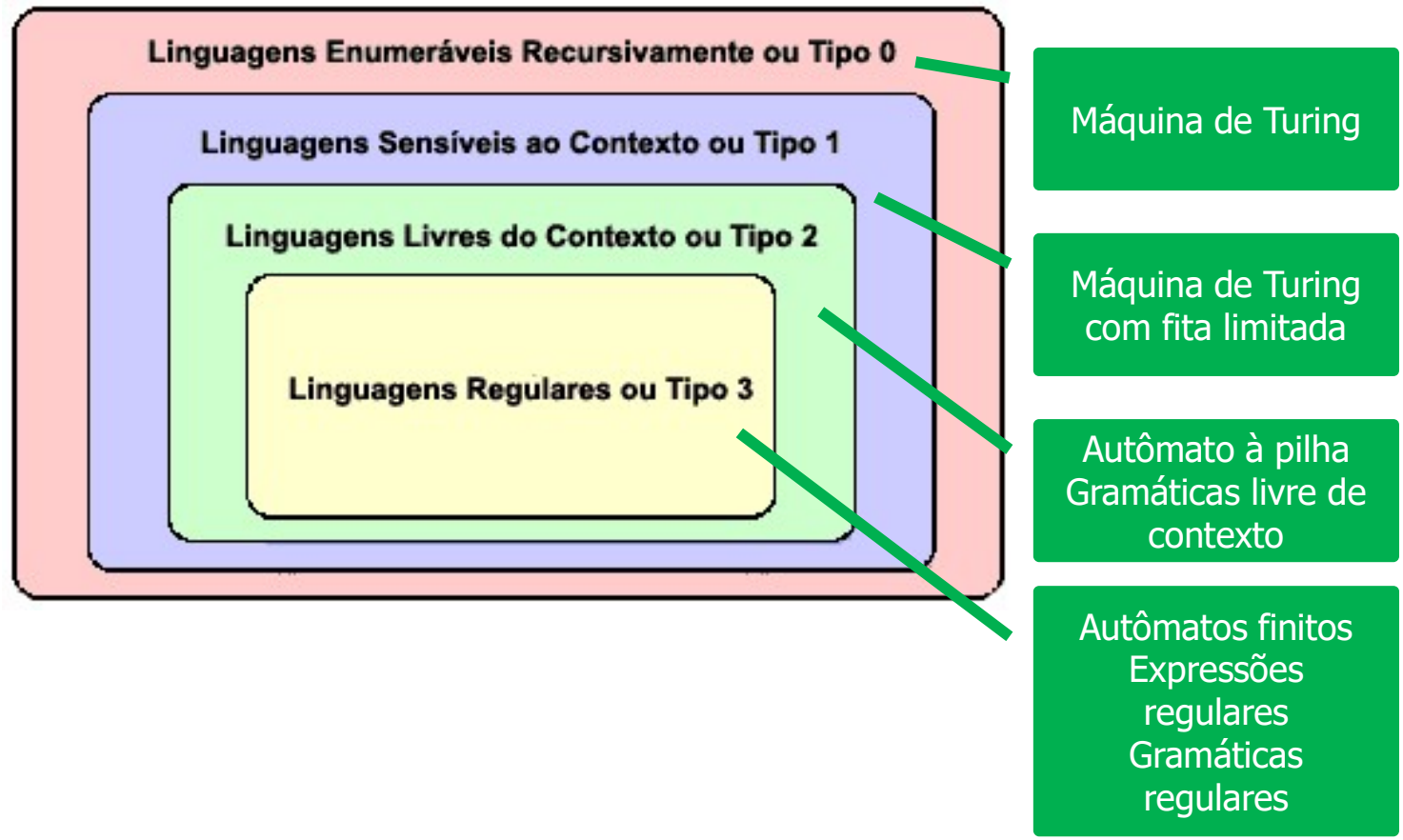
L={(0,1)*00} de forma que você pode usar uma Máquina de *Turing* que não altera os símbolos da fita e sempre move a direita.

MT_(0,1)*00=({0,1},{q_0,q_1,q_2,q_3},Π,q_0,{q3}, {0,1},□,□)

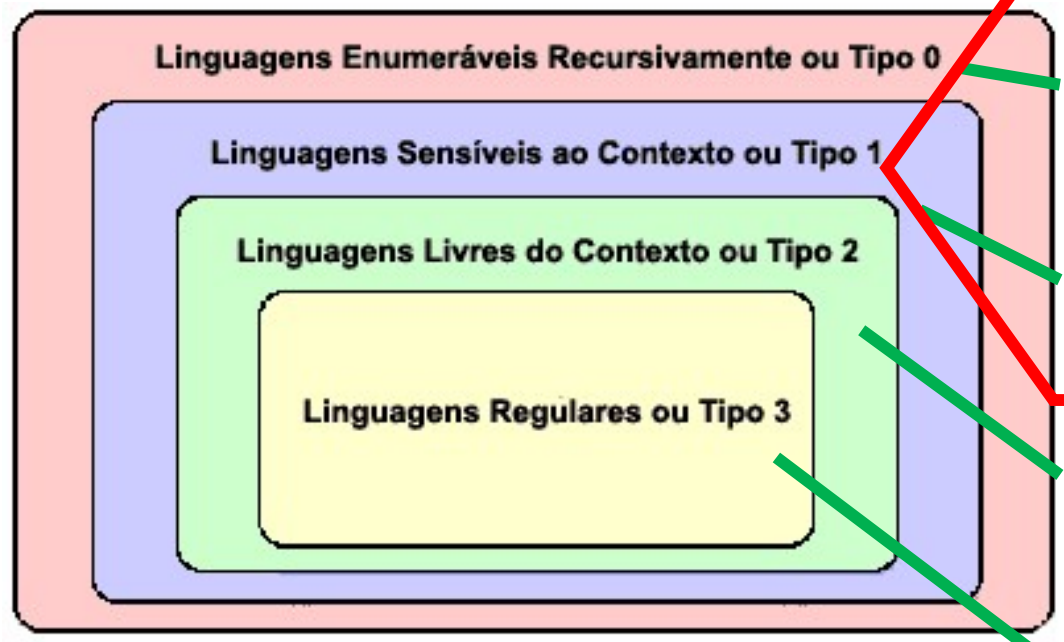


00	Accept
010100	Accept
1111	Reject
00000	Accept
101010100	Accept
10100	Accept

A linguagem de uma máquina de *Turing*



A linguagem de uma máquina de *Turing*



Máquina de Turing

Máquina de Turing com fita limitada

Autômato à pilha
Gramáticas livre de contexto

Autômatos finitos
Expressões regulares
Gramáticas regulares

Usos da máquina de *Turing*

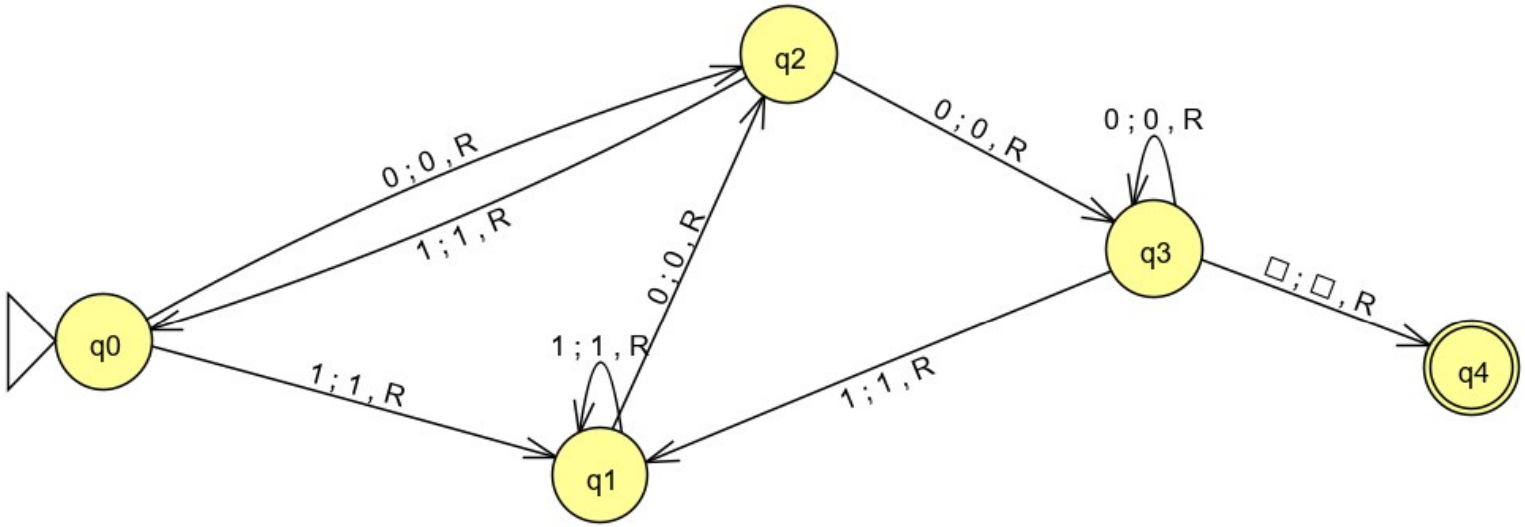
1. como reconhecedor de linguagens
   - Esta cadeia pertence à linguagem L(M)? ou esta é uma solução do problema?
     - Visto na aula passada
2. para calcular funções (resolver problemas)
   - Resolução do problema.
3. para processar problemas de decisão
   - sim/não

Bibliografia

HOPCROFT, J. E.; ULLMAN, J. D.; MOTWANI, R. ***Introdução à Teoria de Autômatos, Linguagens e Computação***. Rio de Janeiro: Elsevier, 2002.

L={(0,1)*00} de forma que você pode usar uma Máquina de *Turing* que não altera os símbolos da fita e sempre move a direita.

$MT\_(0,1)^*00=(\{0,1\},\{q_0,q_1,q_2,q_3\},\Pi,q_0,\{q3\},\{0,1\},\square,\square)$



MT como reconhecedor de linguagens

00	Accept
010100	Accept
1111	Reject
00000	Accept
101010100	Accept
10100	Accept

$L = \{waw \mid w \in \{0,1\}^*\}$

$MT\_(0,1)^*00 = (\{0,1,a\}, \{q_0, q_1, q_2, q_3, q_4, q_5\}, \Pi, q_0, \{q_5\}, \{X\}, \square, \square)$

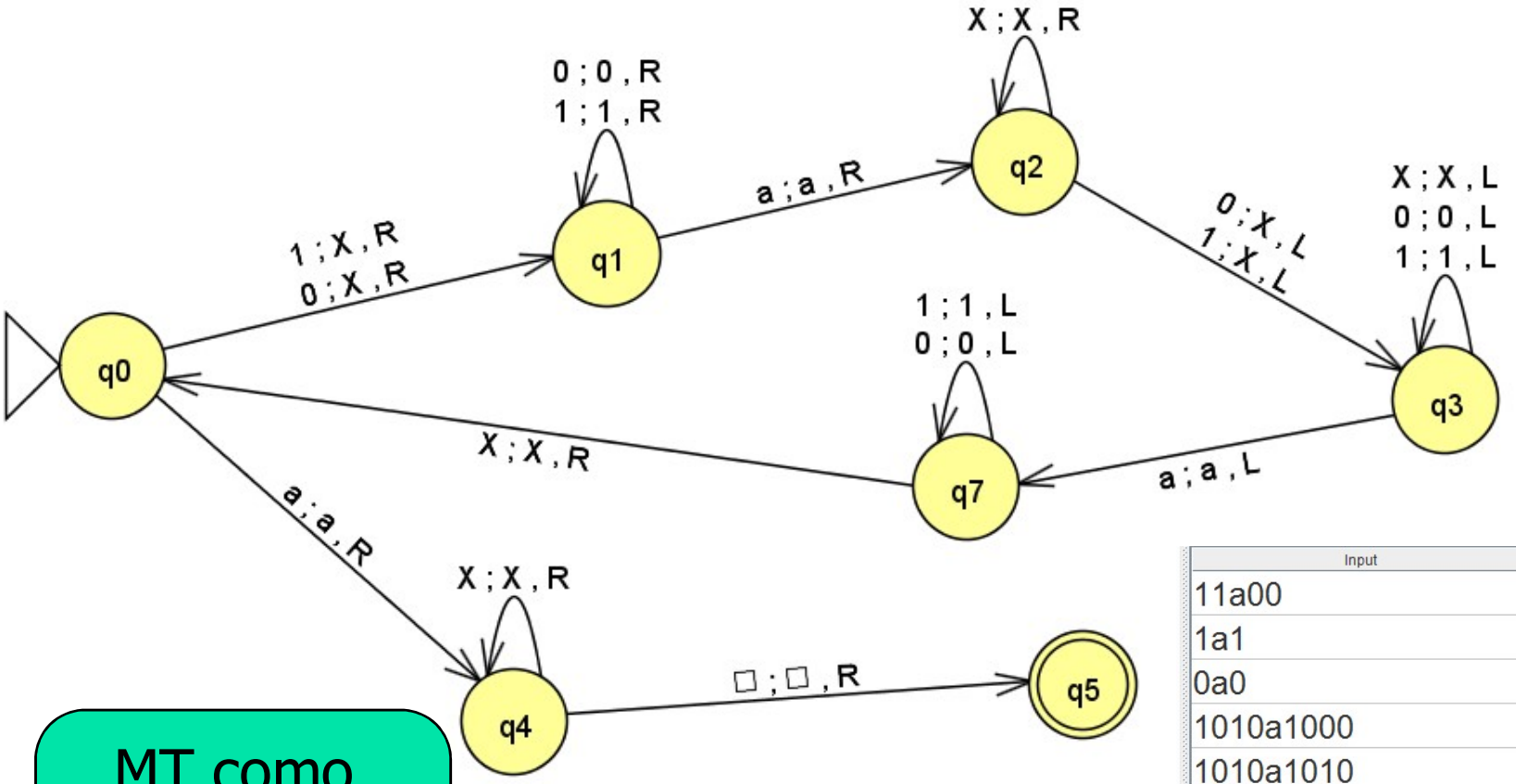
Usos da máquina de *Turing*

MT como reconhecedor de linguagens

Input	Resu
11a00	Accept
1a1	Accept
0a0	Accept
1010a1000	Accept
1010a1010	Accept
1a000	Reject
00a00	Accept
10a10	Accept
100a100	Accept
1a00000	Reject
1aaa111	Reject

$L = \{waw \mid w \in \{0,1\}^*\}$

$MT\_(0,1)^*00 = (\{0,1,a\}, \{q_0, q_1, q_2, q_3, q_4, q_5\}, \Pi, q_0, \{q_5\}, \{X\}, \square, \square)$



MT como reconhecedor de linguagens

Input	Resu
11a00	Accept
1a1	Accept
0a0	Accept
1010a1000	Accept
1010a1010	Accept
1a000	Reject
00a00	Accept
10a10	Accept
100a100	Accept
1a00000	Reject
1aaa111	Reject

L = {ww^r | w é palavra de {a,b}*}

MT_ww^r=({a,b},{q_0,q_1,q_2,q_3,q_4,q_5,q_6},Π,q_0,{q_6},{k},□,□)

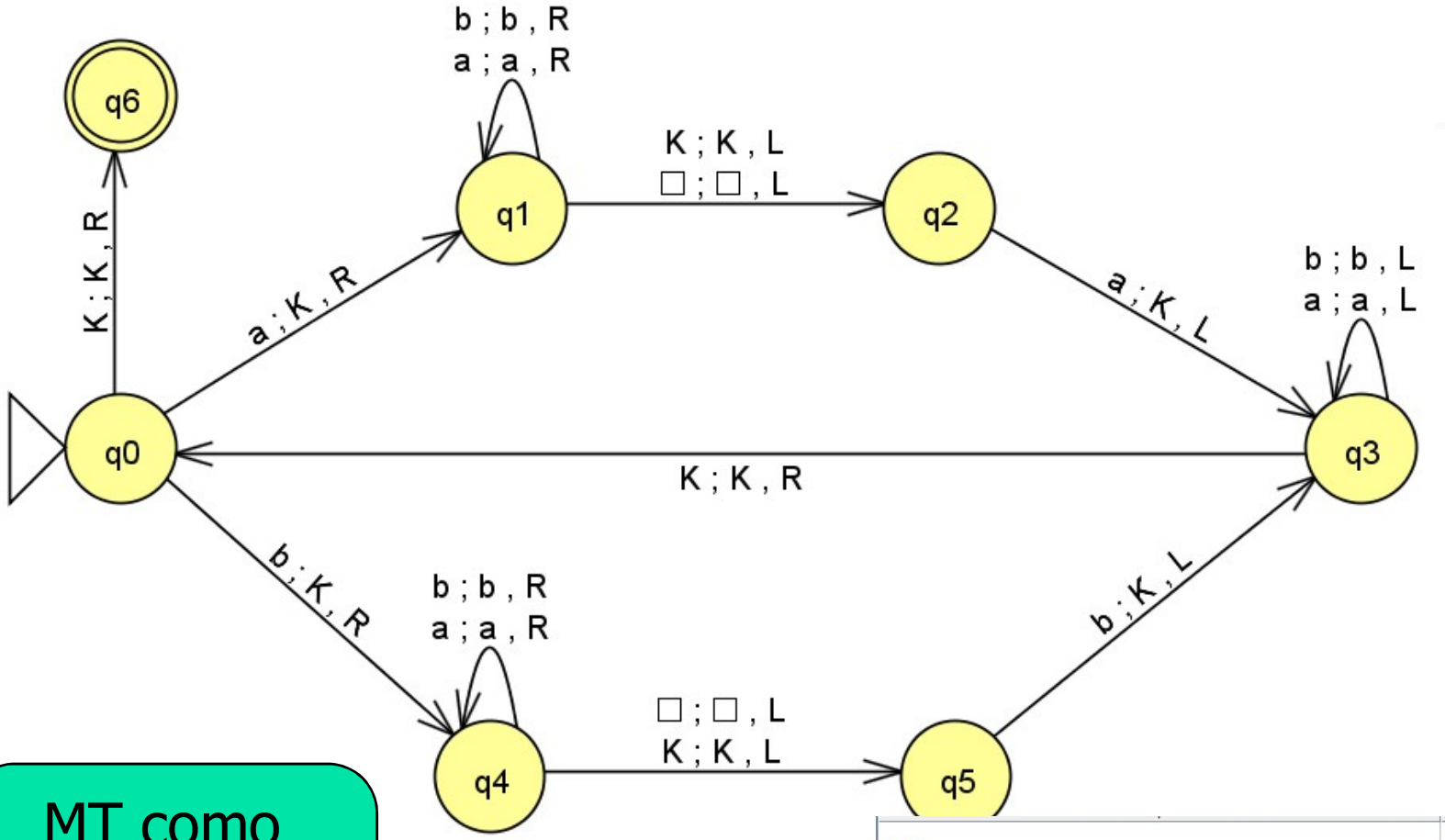
Usos da máquina de *Turing*

MT como reconhecedor de linguagens

aa	Accept
ab	Reject
abba	Accept
aaaaabbbb	Reject
aaaabbbb	Reject

L = {ww^r | w é palavra de {a,b}*}

MT_ww^r=({a,b},{q_0,q_1,q_2,q_3,q_4,q_5,q_6},Π,q_0,{q_6},{k},□,□)



MT como reconhecedor de linguagens

aa	Accept
ab	Reject
abba	Accept
aaaaabbbb	Reject
aaaabbbb	Reject

Uso da máquina de *Turing* para calcular funções (resolver problemas)

- Números inteiros são representados em vocabulário unário.
- O inteiro i >= 0 é representado pela cadeia $\mathbf{0^i}$
- Se a função tem k argumentos (i_1, i_2, ..., i_k) então esses inteiros são colocados na fita separados por **1's** como:

$$0^{i1}1\ 0^{i2}1\ \ldots\ 1\,0^{ik}$$

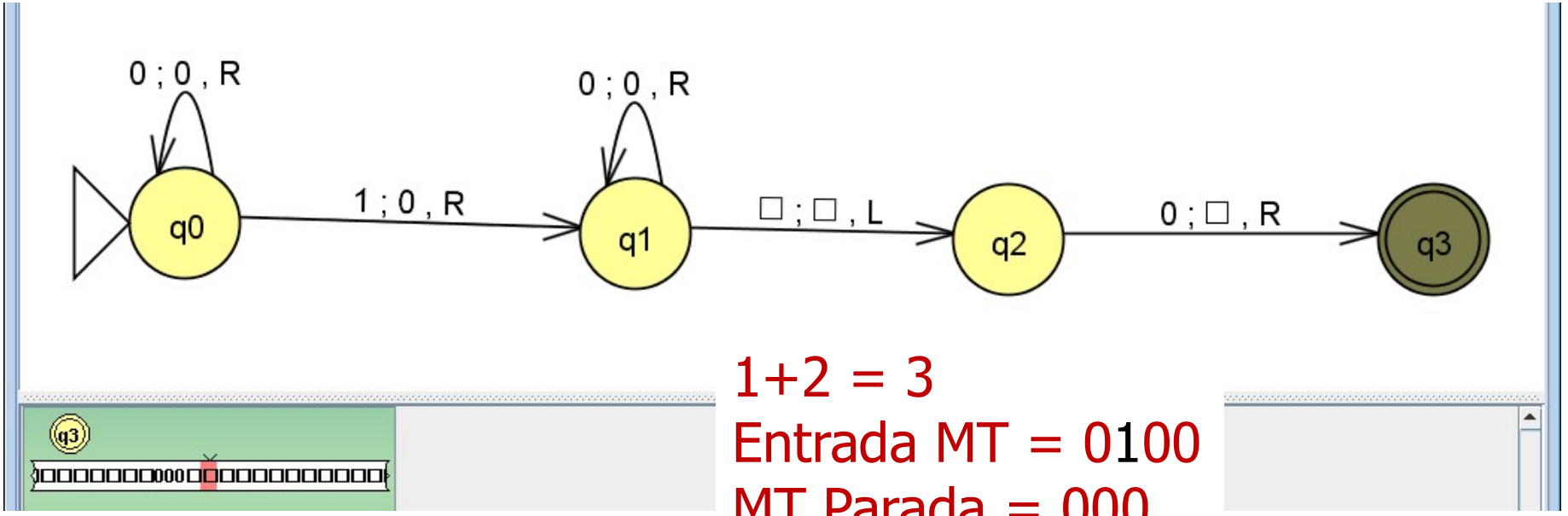
- O inverso também é possível.
- Se a máquina para (não importa em que estado) com a fita consistindo de 0^m para algum m, então dizemos que $f(i_1,i_2,\ldots i_k) = m$, onde f é uma função de k argumentos computados por essa MT

Uso da máquina de *Turing* para calcular funções (resolver problemas)

- MT que realiza a soma de dois números naturais não negativos → a+b
- Conteúdo inicial da fita: 0^a10^b
- MT parada: 0^{a+b}

Uso da máquina de *Turing* para calcular funções (resolver problemas)

- MT que realiza a soma de dois números naturais não negativos → a+b
- Conteúdo inicial da fita: 0^a10^b
- MT parada: 0^{a+b}



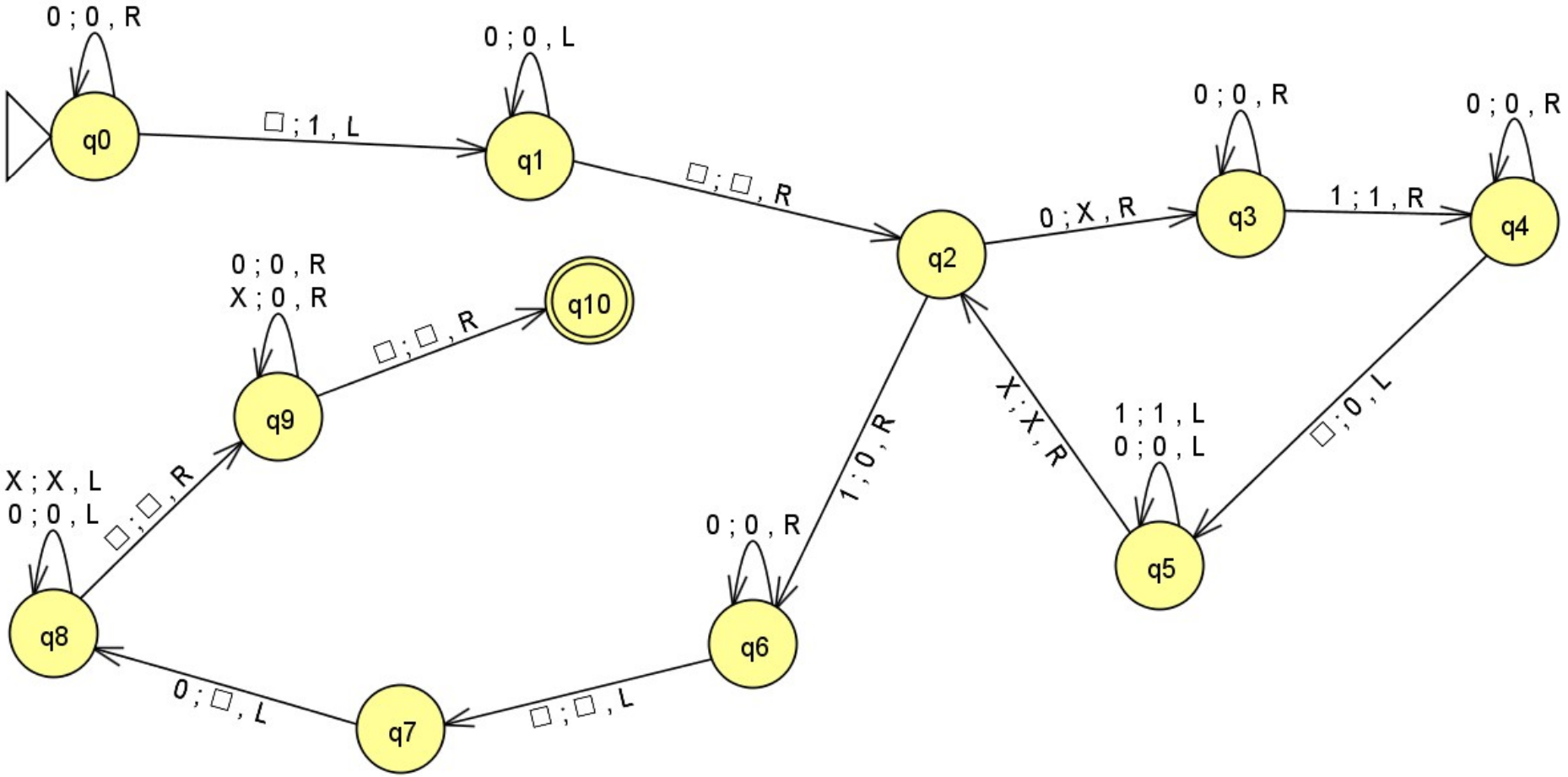
1+2 = 3
Entrada MT = 0100
MT Parada = 000

Uso da máquina de *Turing* para calcular funções (resolver problemas)

- MT que realiza a operação a*2
- Conteúdo inicial da fita: 0^a
- MT parada: 0^{a+a}
- Sugestão:
  - Grave 1 depois do último 0.
  - Grave o mesmo número de 0's da entrada, à direita do 1. Haverá necessidade de substituir os 0's originais por X
  - Substitua o 1 por 0 (nesse momento a cadeia da fita é $\mathbf{X^a0^{a+1}}$).
  - Continue movendo à direita sem mudar a fita, até que um B seja encontrado.
  - Mantenha o B e mova a esquerda para encontrar o último 0 mais a direita. Substitua esse 0 por B, para obter $\mathbf{X^a0^a}$.
  - Substitua todos os X por 0.

Uso da máquina de *Turing* para calcular funções (resolver problemas)

- MT que realiza a operação a*2

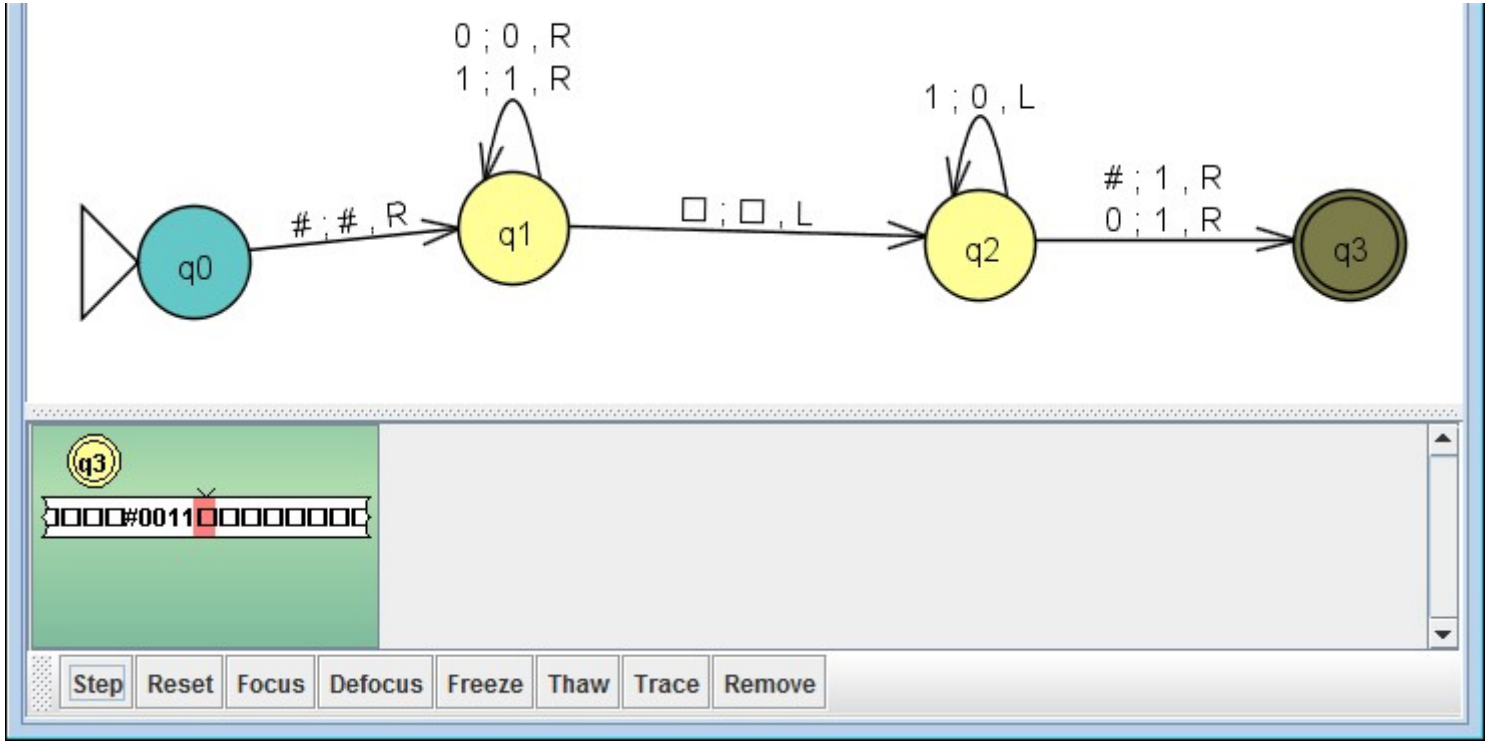


Uso da máquina de *Turing* para calcular funções (resolver problemas)

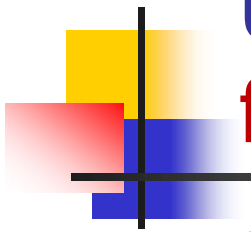
- MT que realiza a operação soma 1 em números representados em binário, começados por #
- Exemplos:
  - Conteúdo inicial da fita: #0010
  - MT parada: #0011
  - Conteúdo inicial da fita: #100
  - MT parada: #101
  - Conteúdo inicial da fita: #111
  - MT parada: #1000

Uso da máquina de *Turing* para calcular funções (resolver problemas)

- MT que realiza a operação soma 1 em números representados em binário, começados por **#**



Uso da máquina de *Turing* para calcular funções (resolver problemas)

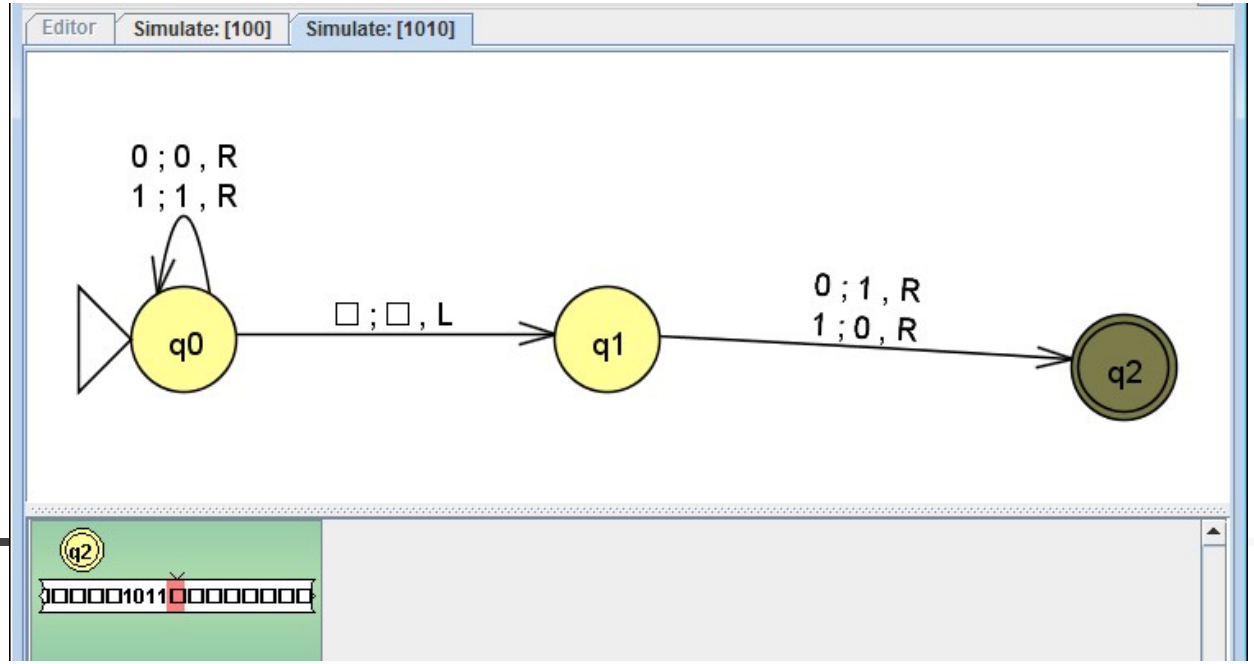


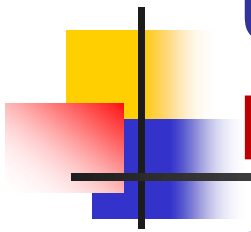
- MT que aceita números representados em binário, e produza a saída de acordo com as seguintes condições:
  - Se o número for impar, subtrai 1
  - Se o número for par, soma 1

- Exemplos:
  - Conteúdo inicial da fita: #101
  - MT parada: #100

  - Conteúdo inicial da fita: #1001
  - MT parada: #1000

Uso da máquina de *Turing* para calcular funções (resolver problemas)

- MT que aceita números representados em binário, e produza a saída de acordo com as seguintes condições:
  - Se o número for impar, subtrai 1
  - Se o número for par, soma 1
- 



Uso da máquina de *Turing* para processar problemas de decisão

- Quando a MT é utilizada para decidir problemas (S/N)

Uso da máquina de *Turing* para processar problemas de decisão

1. MT que decide se um número representado na base binária é par ou impar

Fica em q0 enquanto par

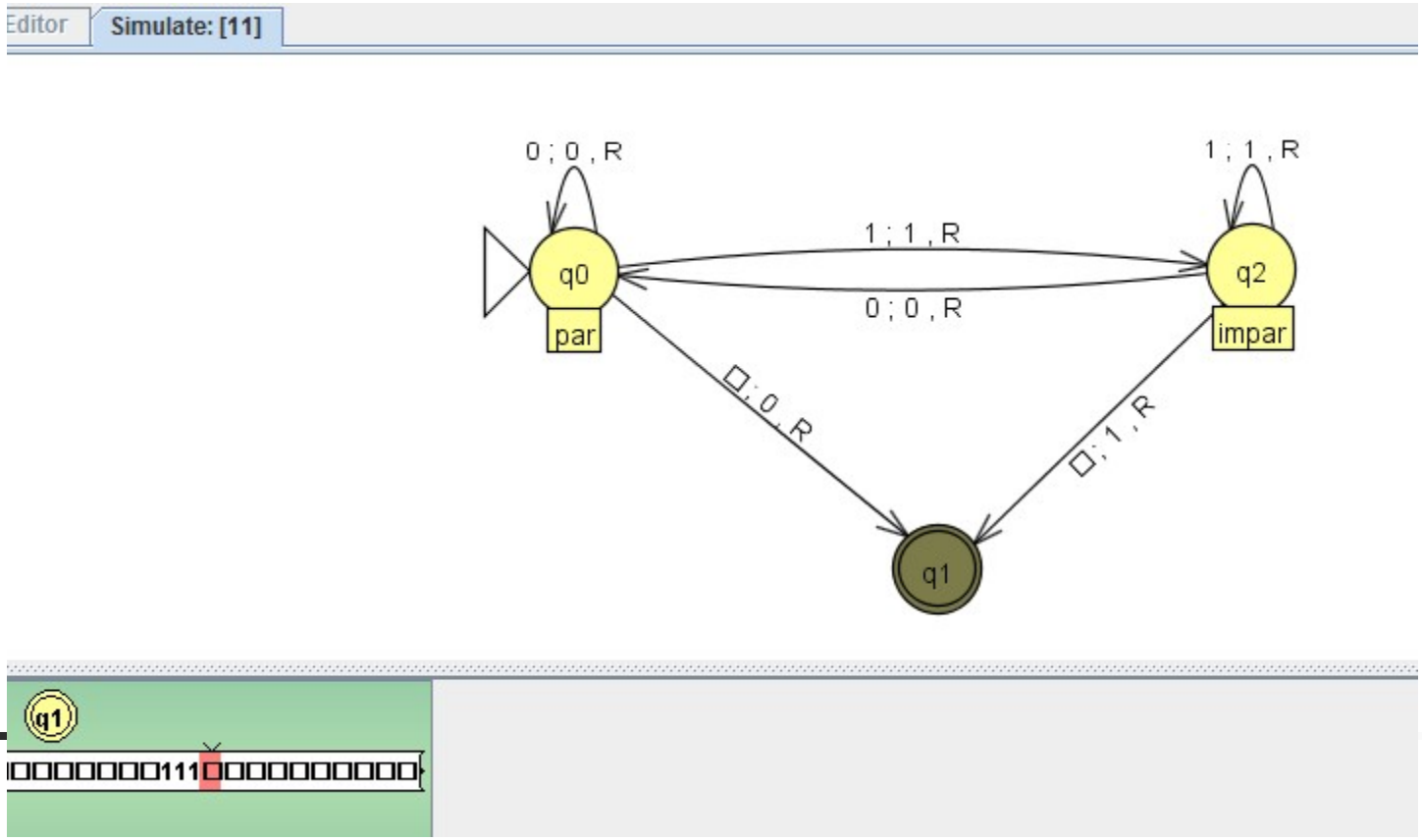
Fica em q1 enquanto impar

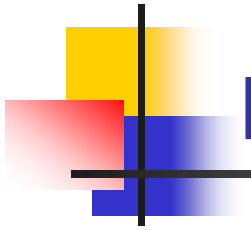
Se Par para e escreve 0

Se Impar para e escreve 1

Uso da máquina de *Turing* para processar problemas de decisão

1. MT que decide se um número representado na base binária é par ou impar





Máquina de *Turing* – Exercícios

1. Construir uma MT que decida se uma sequência de parênteses é bem formada.
   - Escreve 0 se mal formada
   - Escreve 1 se bem formada

Dica: considere que a cadeia de parênteses é iniciada por # para facilitar a checagem.

Ideia: Procurem por um) e substitua por X e em seguida voltar a esquerda procurando o (mais próximo para substituir por X também.

Máquina de *Turing* – Exercícios

1. Construa uma máquina de Turing que receba uma cadeia de 1's e 0's: M = (Q, {0,1}, Γ, δ, q0, F) que resolva o comando: ***Se 1 então soma_elementos senão zera_número***

Ou seja,

- (1) se a MT recebe uma cadeia do tipo "***1***011111011" então deve calcular a soma de "11111" por "11", que dá "1111111" e ficar com a fita: "10111111011111111"
- (2) se receber a cadeia "***0***011111011", então deve zerar o número e ficar com a fita: "**0**01111101100".

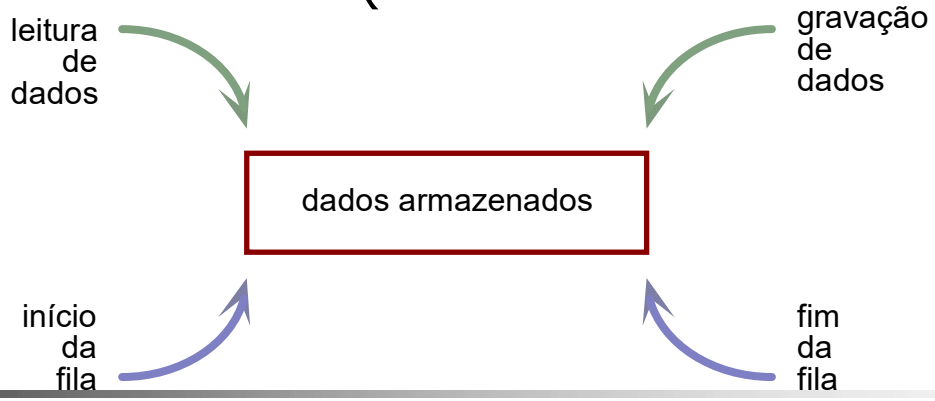
Máquina de *Turing* - Exercícios

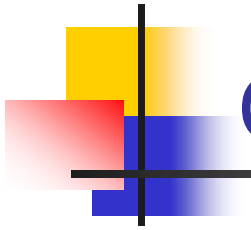
2. Construa uma máquina de *Turing* que receba como entrada uma cadeia da linguagem da expressão regular #0 (0 + 1) (0 + 1)*# representando um número *n* na base 2 delimitado pelo símbolo "#" – por exemplo, #010# - e produza, como saída, a cadeia correspondente ao número *n + 2* na base 2, também delimitado por "#" – resultando neste caso em #100#

3. Projete uma MT que receba como entrada uma sequência binária e produza como resultado o valor binário multiplicado por 4;

Outros Modelos de Máquinas Universais

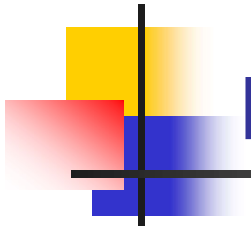
- Máquina de Post
  - A principal característica da Máquina de Post é de utilizar uma estrutura de dados do tipo *fila* para entrada, saída e memória de trabalho.
  - Estruturalmente, o primeiro valor gravado é também o primeiro a ser lido (uma leitura exclui o dado lido).





Outros Modelos de Máquinas Universais

- Máquina de Post e Máquina Norma são equivalentes à Máquina de Turing
  - Infinitos registradores (Norma), fita infinita (Turing) e fila (Post) são estruturas que podem simular uma às outras.

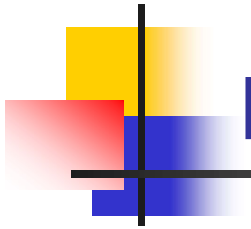


Máquina de Post

- Uma MP consiste de duas partes:

a) Variável X

- Trata-se de uma variável do tipo fila e é utilizada como entrada, saída e memória de trabalho.
- A variável X não possui tamanho nem limite fixos. Seu comprimento é igual ao comprimento da palavra corrente armazenada.
- Os símbolos podem pertencer ao alfabeto de entrada ou a { # }, único símbolo auxiliar.
- Inicialmente, o valor de X é a palavra de entrada. Caso X não contenha símbolos, a entrada é vazia, representada por ε.

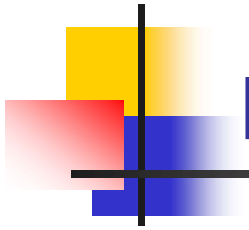


Máquina de Post

- Uma MP consiste de duas partes:

b) Programa

- **É uma sequência finita de instruções, representado como um diagrama de fluxos, no qual cada vértice é uma instrução.**
- **As instruções podem ser de quatro tipos: partida, parada, desvio (leitura com teste) e atribuição.**



Máquina de Post

- Definição

$$M = (\Sigma, D, \#)$$

Σ *alfabeto de símbolos de entrada*;

D *programa* ou *diagrama de fluxos* construído a partir de componentes elementares - partida, parada, desvio e atribuição;

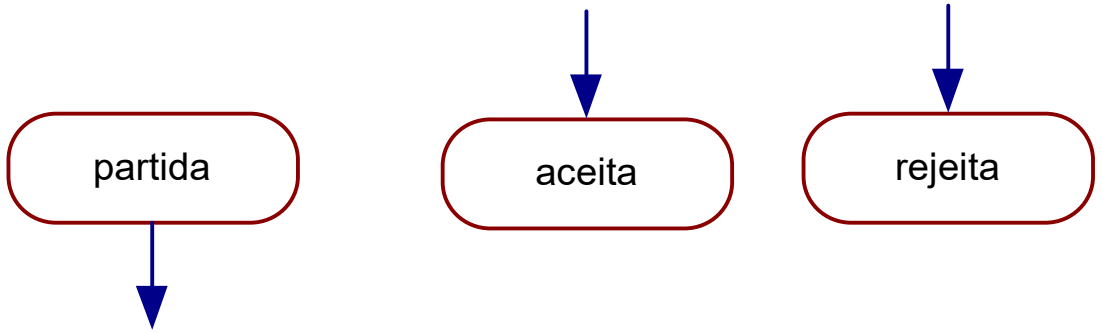
\# *símbolo auxiliar*.

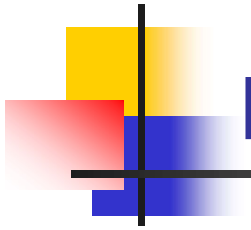
Máquina de Post

- Diagrama de Fluxos (Componentes)

a) ***Partida:*** Existe somente uma instrução de início em um programa

b) ***Parada:*** Existem duas alternativas de instruções de parada em um programa, uma de aceitação e outra de rejeição:





Máquina de Post

- Diagrama de Fluxos (Componentes)
  - c) ***Desvio ou (leitura com Teste):***

$$X \leftarrow ler\ (X)$$

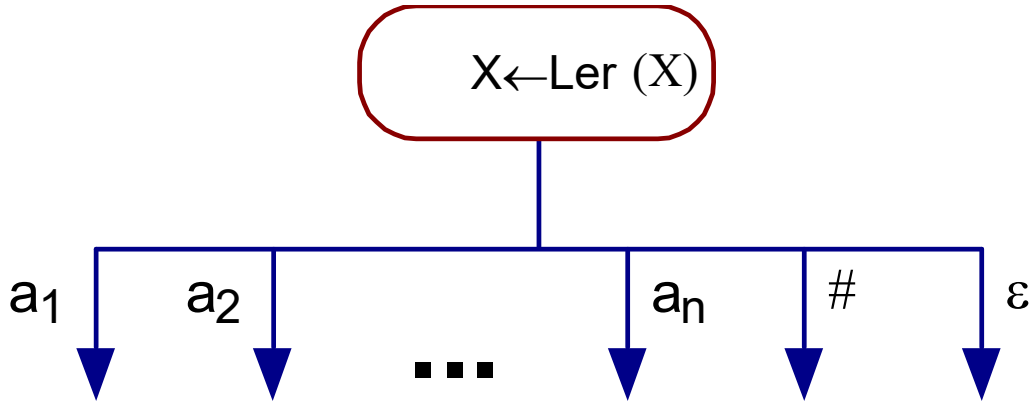
  - **lê o símbolo mais à esquerda da palavra armazenada em X, retirando o primeiro símbolo e desviando o fluxo do programa de acordo com o símbolo lido;**
  - **fluxo do programa é determinado de acordo com o símbolo mais à esquerda da palavra.**
  - **deve ser prevista a possibilidade de X conter a palavra vazia.**

Máquina de Post

c) ***Desvio ou (leitura com Teste):***

$$X \leftarrow ler\ (X)$$

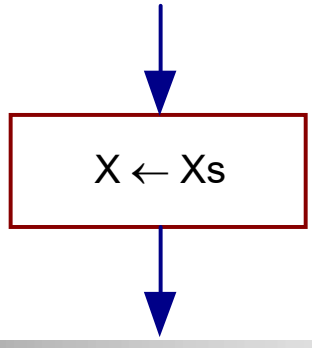
- **Se o $\sum$ (entrada) é n, então existem n+2 arestas de desvios condicionais, pois se deve incluir as possibilidades # e ε**

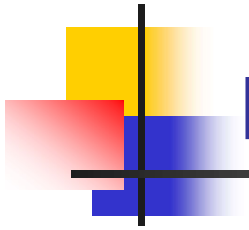


Máquina de Post

d) ***Atribuição***. *X←Xs*

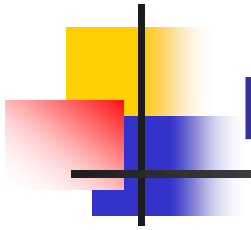
- É uma instrução de concatenação, gravando o símbolo indicado (pertencente a $\Sigma \cup \{ \# \}$) à direita da palavra armazenada na variável X (fim da fila).
- A operação de atribuição é representada a seguir, supondo que $s \in \Sigma \cup \{ \# \}$.





Máquina de Post

- Diagrama de Fluxos
  - Existe somente uma partida, mas podem existir diversas (zero ou mais) instruções de parada (aceitação / rejeição) ou ficar em *loop infinito*
  - Em um desvio, se X contém ε, então segue o fluxo correspondente. Caso contrário, lê o símbolo mais à esquerda da palavra em X e remove-o após a decisão da próxima instrução



Máquina de Post

- Exemplo – Duplo Balanceamento

  $$Duplo\_Bal = \{ a^n b^n \mid n \geq 0 \}$$

  - A Máquina de Post:

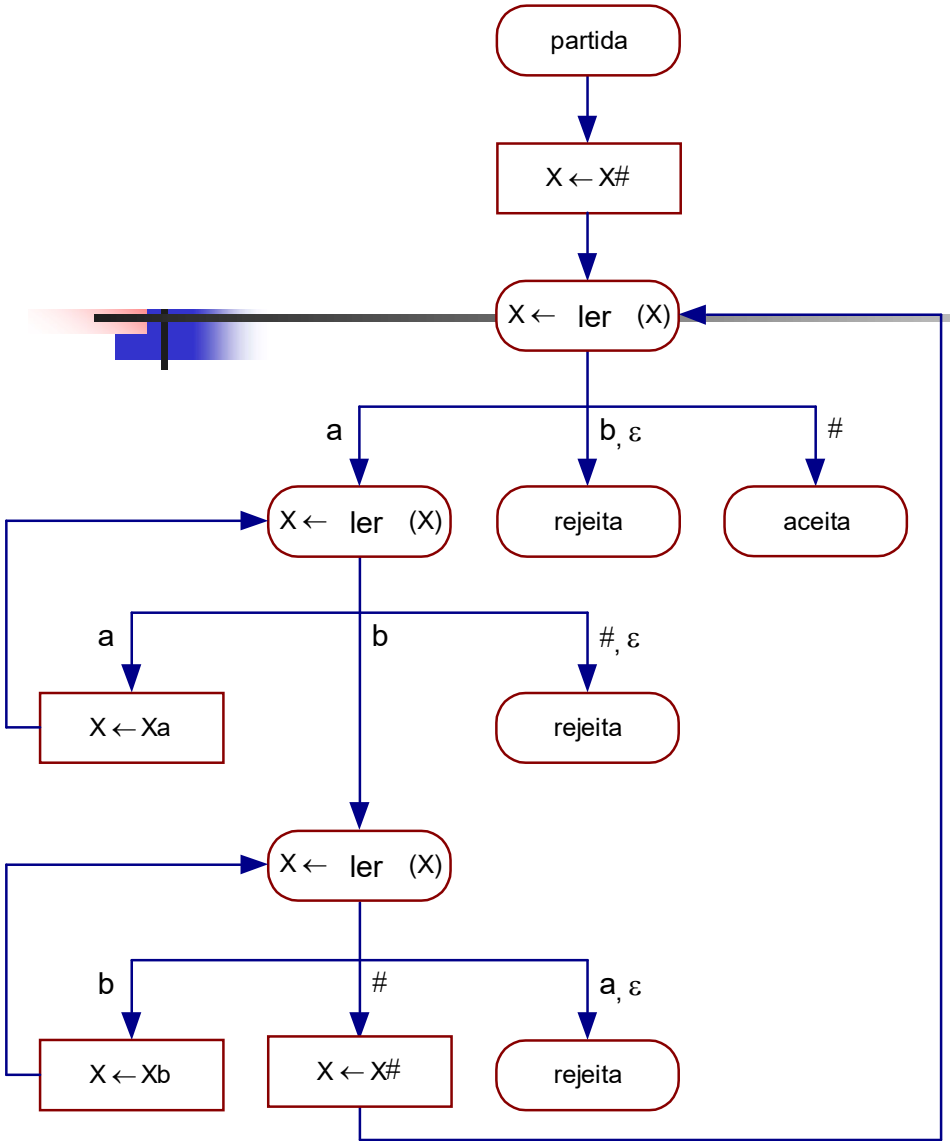
    $$Post\_Duplo\_Bal = (\{ a, b \}, D, \#)$$

  - onde D é, ...(fluxograma)..., tal que:

ACEITA(Post_Duplo_Bal) = Duplo_Bal

REJEITA(Post_Duplo_Bal) = Σ^* - Duplo_Bal

LOOP(Post_Duplo_Bal) = $\varnothing$.



· O algoritmo lê e remove o primeiro símbolo **a**;

· Realiza uma varredura circular em busca do correspondente **b**.

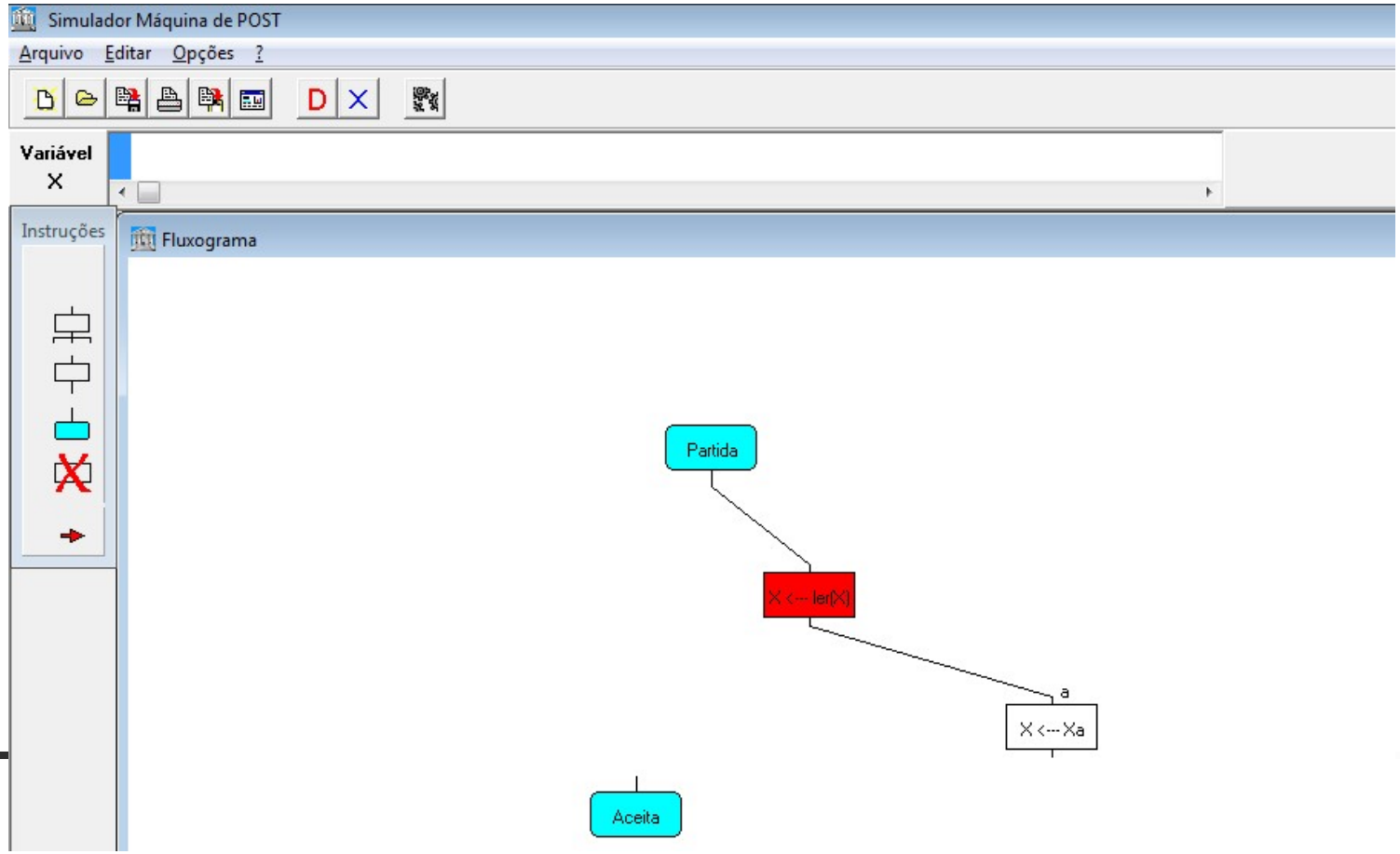
· Essa varredura é realizada através de sucessivas leituras (e remoções), armazenando o símbolo lido à direita de **X**.

· Ao encontrar o **b**, este é removido, e uma nova varredura circular é realizada até o fim da palavra de entrada (identificado pelo símbolo auxiliar **#**, atribuído a **X** no início do processamento).

· Este ciclo é repetido até restar a palavra vazia ou ocorrer alguma condição de rejeição.

Máquina de *Post* - Simulador

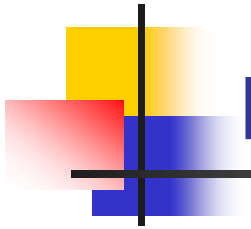
http://deexatas.blogspot.com/2015/07/usando-o-simulador-da-maquina-de-post.html



Máquina de *Post* - Exercícios

1. Desenvolver uma máquina de *Post*, sobre o alfabeto {(,)}, que verifique se uma sequência de parênteses é bem formada. Exemplos:

Entrada – Fila	Saída – Fila	Status
( )	indiferente	aceita
) (	indiferente	rejeita
(( ) ( ))	indiferente	aceita
(( ))) ( )	indiferente	rejeita
β	indiferente	aceita



Máquina de *Post* - Exercícios

2. Exercícios 3.6

3. Projete uma Máquina de *Post* para reconhecer o conjunto de todas as cadeias de O's e 1's terminando por 00 L={(0,1)*00}.

4. Projete uma Máquina de *Post* para reconhecer o conjunto de todas as cadeias de entrada representada pela linguagem L= {(00,1)*01}.

5. Projete uma Máquina de *Post* para reconhecer se quantidade de 1's é par. Exemplo: Entrada 11 é aceita; Entrada 111 é rejeitada

Máquina de *Post* - Exercícios

6. Desenvolver uma máquina de Post, sobre o alfabeto {a, b, c}, que reconheça a linguagem $L = \{a^n b^n c \mid n \geq 1\}$. A seguir, são apresentados alguns exemplos de entradas possíveis de serem fornecidas pelo usuário com seus respectivos resultados.

Entrada – Fila	Saída – Fila	Status
aabbc	indiferente	aceita
aabbcc	indiferente	rejeita
aabbbc	indiferente	rejeita
abc	Indiferente	aceita
β	indiferente	rejeita

7. Adição entre 2 números (em notação unária).
8. Multiplicação entre 2 números (em notação unária).