Teoria da Computação

Subrotinas e Extensões de Máquinas de Turing

Thiago Alves

Subrotina

- Podemos combinar subprogramas para construir programas
- Podemos fazer o mesmo com máquinas de Turing

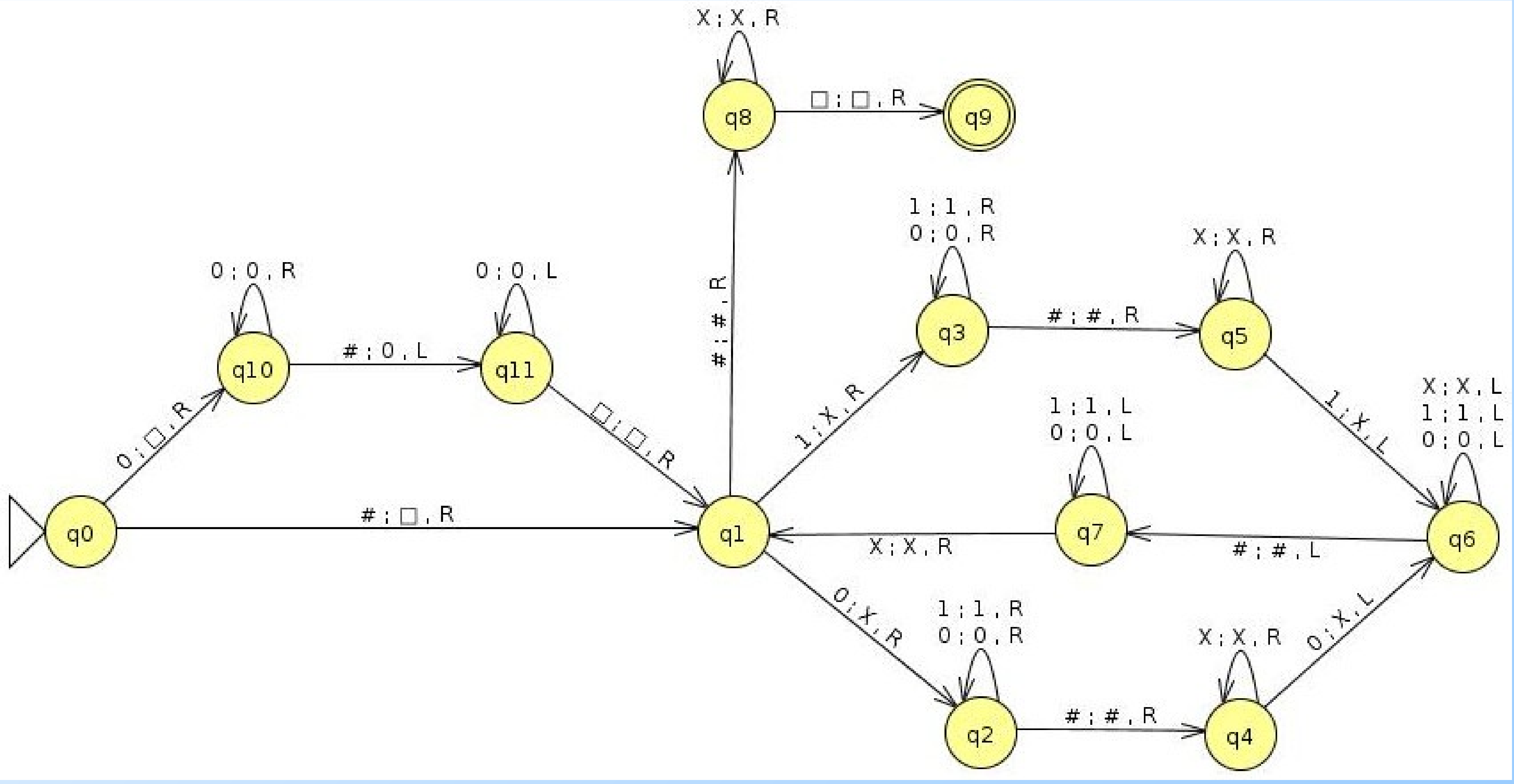
Subrotina

- Seja L = $\{0^i\#0^j\#0^k \mid i,j,k \geq 0 \text{ e } i + j = k\}$
- Vamos definir uma máquina de Turing para decidir L
- Vamos usar a máquina de Turing M que decide $\{w\#w \mid w \in \{0,1\}^*\}$

Subrotina

- Seja $L = \{0^i\#0^j\#0^k \mid i,j,k \geq 0 \text{ e } i + j = k\}$
- Vamos definir uma máquina de Turing para decidir L
- Podemos trocar o primeiro 0 por B e o primeiro # por 0
- E usar a máquina de Turing M que decide $\{w\#w \mid w \in \{0,1\}^*\}$

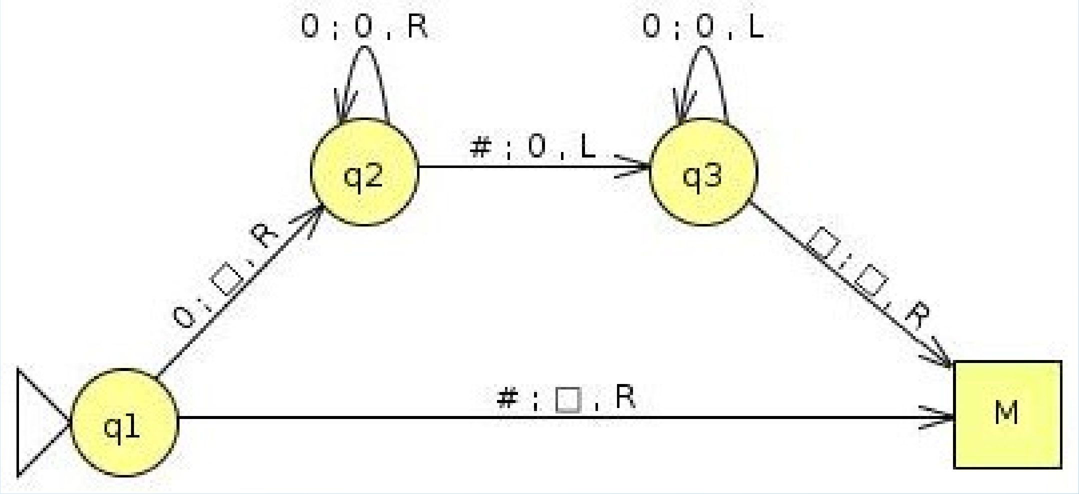
Subrotina



X ; X , R
q8
□ ; □ , R
q9
1 ; 1 , R
0 ; 0 , R
X ; X , R
0 ; 0 , R
0 ; 0 , L
; # , R
q3
; # , R
q5
q10
; 0 , L
q11
1 ; X , R
1 ; X , L
X ; X , L
1 ; 1 , L
0 ; 0 , L
1 ; 1 , L
0 ; 0 , L
0 ; □ , R
□ ; □ , R
q7
q0
; □ , R
q1
X ; X , R
; # , L
q6
0 ; X , R
1 ; 1 , R
0 ; 0 , R
X ; X , R
0 ; X , L
q2
; # , R
q4

Subrotina

- Podemos encapsular a máquina de Turing M

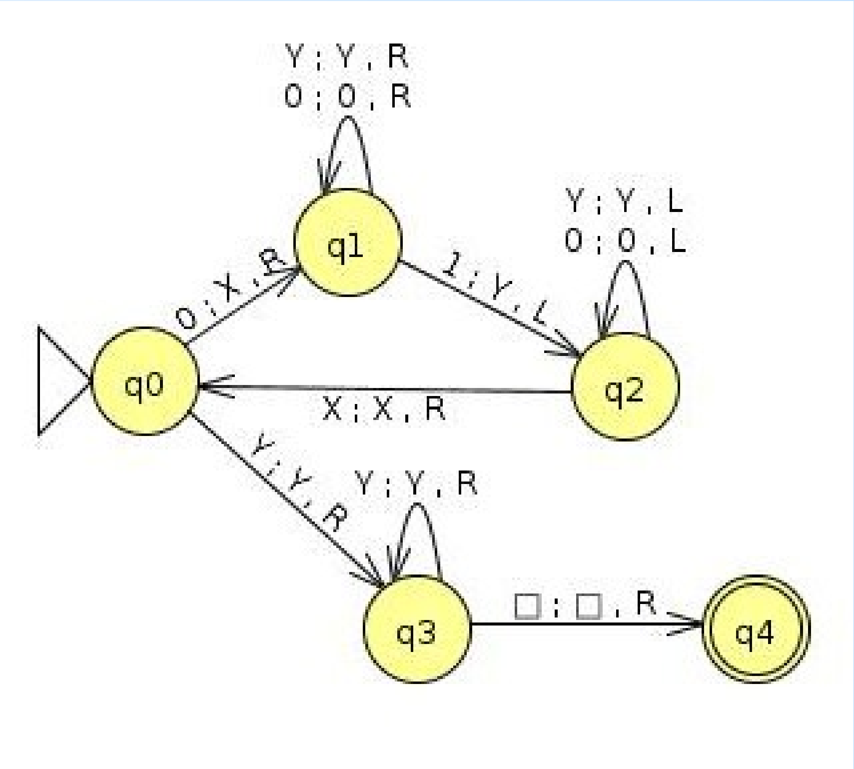


Exemplo

- Seja A = $\{0^n 1^n 2^n \mid n>0\}$
- Mostre que A é recursiva usando a máquina de Turing M_{0n1n} que decide $\{0^n 1^n \mid n > 0\}$

Exemplo

- Máquina de Turing M_{0n1n}:

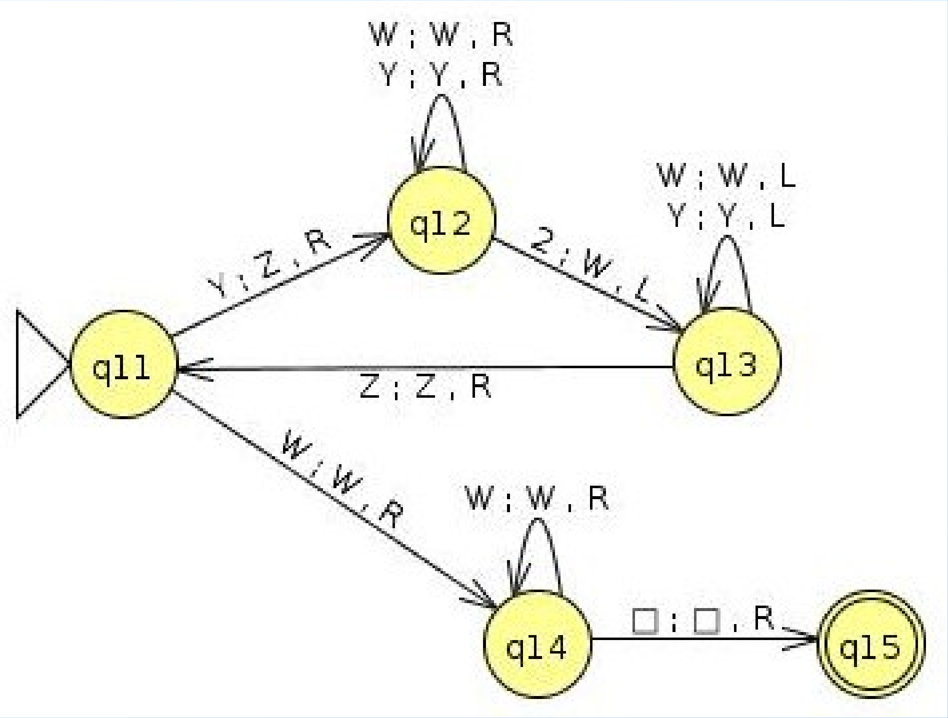


Exemplo

- $A = \{0^n 1^n 2^n \mid n>0\}$
- Podemos usar a máquina M_{0n1n}
- E deixar a fita como X...XY...Y2...2
- Usar uma outra máquina MYn2n para verificar se a quantidade de Y's é igual a de 2's

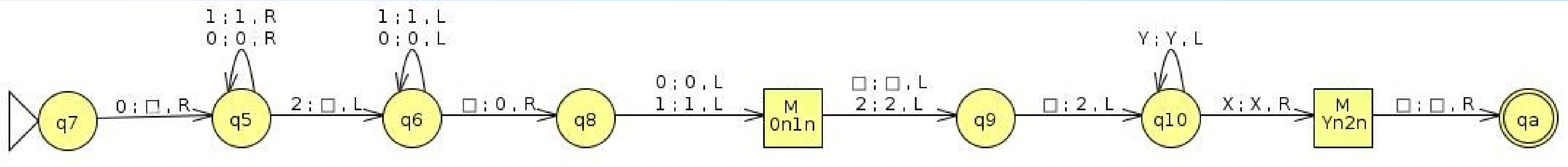
Exemplo

- Máquina de Turing para decidir $\{Y^n 2^n \mid n>0\}$



Exemplo

- Máquina de Turing para decidir a linguagem A = $\{0^n 1^n 2^n \mid n>0\}$



Máquinas de Turing Multifita

- Possui várias fitas
- Cada fita tem seu controle para leitura e escrita
- Inicialmente, a primeira fita tem a entrada e as outras apenas brancos
- A função de transição permite ler, escrever e mover em algumas ou todas as fitas ao mesmo tempo

Máquinas de Turing Multifita

- $\delta: Q \mathrm{x} \Gamma^k \rightarrow Q \mathrm{x} \Gamma^k \mathrm{x} \{L, R\}^k$
- $\delta(q_i, a_1, \ldots, a_k) = (q_j, b_1, \ldots, b_k, D_1, D_2, \ldots, D_k)$
- Em que k é o número de fitas

Máquinas de Turing Multifita

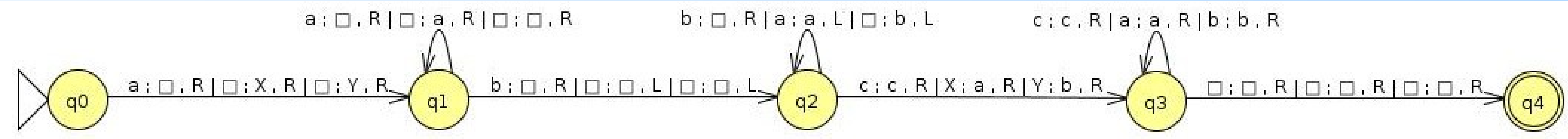
- Vamos definir uma máquina com três fitas para decidir a linguagem $L = \{a^n b^n c^n \mid n > 0\}$

Máquinas de Turing Multifita

- Vamos copiar os a's para a segunda fita e os b's para a terceira fita
- Depois comparar se possuem o mesmo tamanho

Máquinas de Turing Multifita

- $L = \{a^n b^n c^n \mid n > 0\}$



Teorema

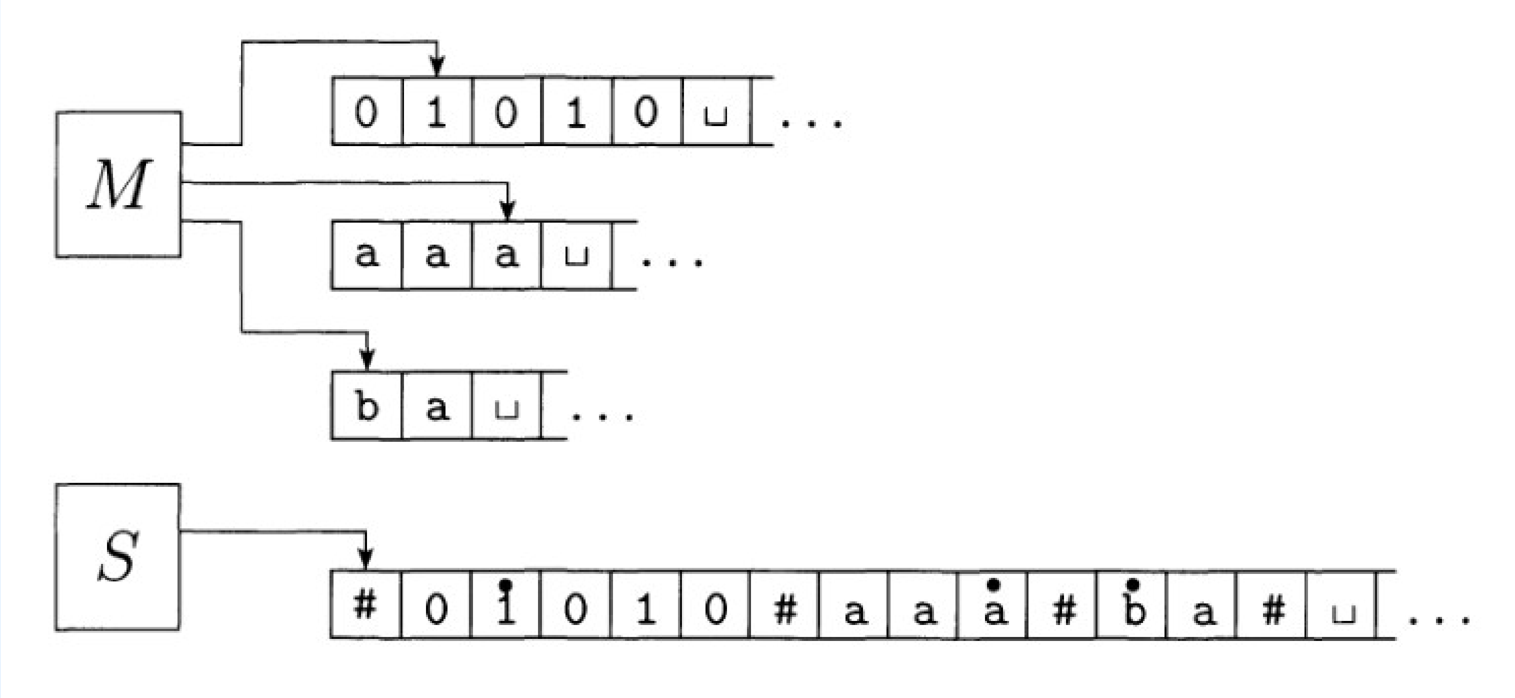
- Para toda máquina de Turing multifita M, existe uma máquina de Turing S tal que L(M) = L(S).

Prova

- Simular M com S
- S armazena as k fitas de M na sua fita única usando # como separador
- Para marcar o controle de cada fita, basta usar um novo símbolo para cada símbolo da fita

Prova

- Ideia da prova



Prova

- Descrição da máquina S:

$S =$ "On input $w = w_1 \cdots w_n$:

1. First S puts its tape into the format that represents all k tapes of M. The formatted tape contains

$$\#\dot{w}_1 w_2 \cdots w_n \#\dot{\sqcup}\#\dot{\sqcup}\# \cdots \#$$

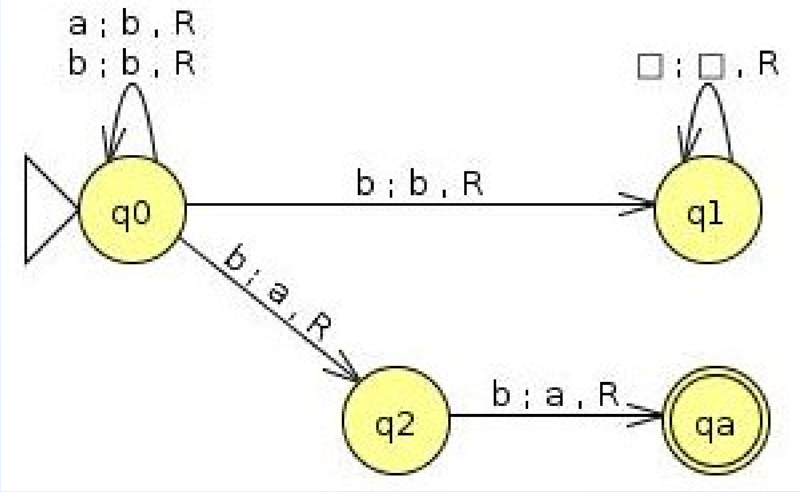
2. To simulate a single move, S scans its tape from the first #, which marks the left-hand end, to the $(k+1)$st #, which marks the right-hand end, in order to determine the symbols under the virtual heads. Then S makes a second pass to update the tapes according to the way that M's transition function dictates.
3. If at any point S moves one of the virtual heads to the right onto a #, this action signifies that M has moved the corresponding head onto the previously unread blank portion of that tape. So S writes a blank symbol on this tape cell and shifts the tape contents, from this cell until the rightmost #, one unit to the right. Then it continues the simulation as before."

Máquinas de Turing Não Determinísticas

- Em qualquer ponto da computação pode continuar de acordo com várias possibilidades
- Se alguma computação chegar em um estado final, a entrada é aceita
- δ: $Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L,R\}}$
- Exemplo:
  - $\delta(q_i, a)=\{(q_j,b_1,L),(q_l,b_2,R),(q_m,b_3,L)\}$

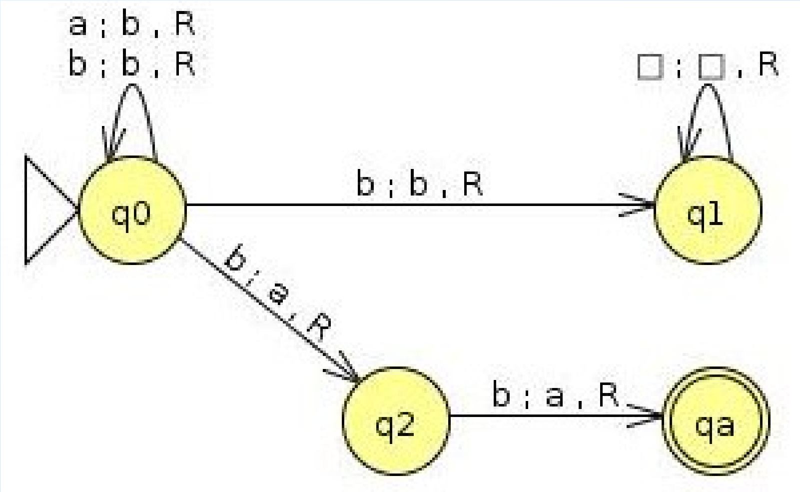
Máquinas de Turing Não Determinísticas

- Exemplo de máquina de Turing não determinística:



Exemplo

- Vamos executar com entrada abb

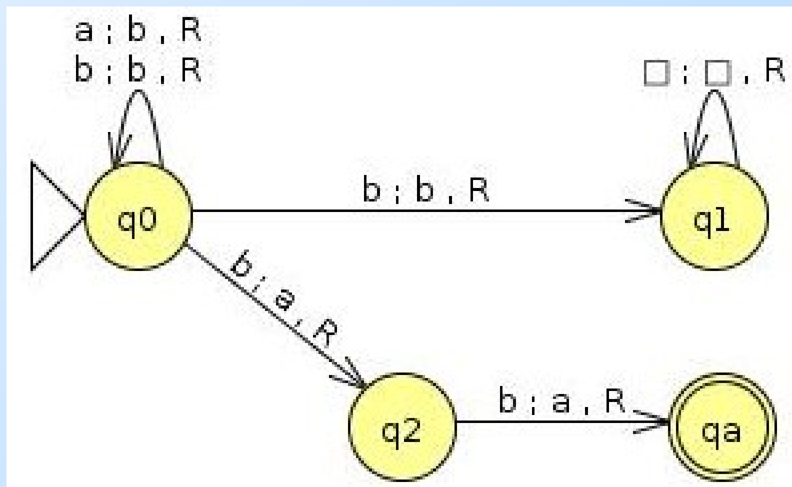


Exemplo

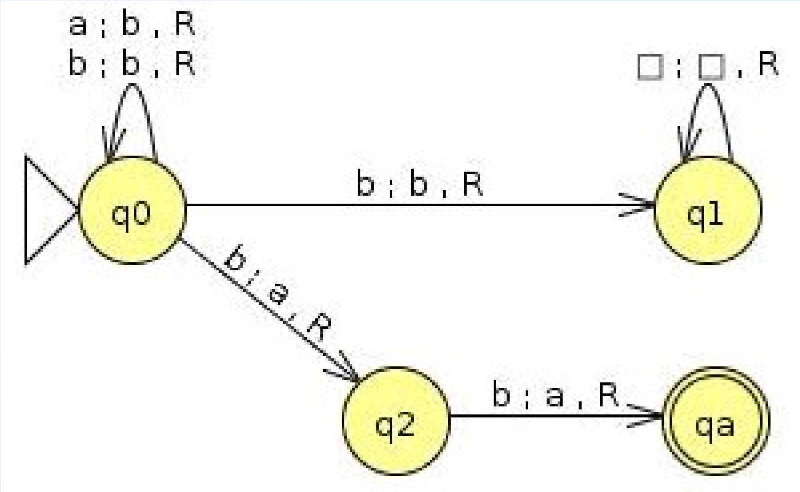
- $q_0abb \vdash bq_0bb \vdash baq_2b \vdash baaq_a$
- $q_0abb \vdash bq_0bb \vdash bbq_1b$
- $q_0abb \vdash bq_0bb \vdash bbq_0b \vdash bbbq_0B$
- $q_0abb \vdash bq_0bb \vdash bbq_0b \vdash bbbq_1B$ ...
- $q_0abb \vdash bq_0bb \vdash bbq_0b \vdash bbaq_2B$

Exemplo

- Vamos executar com entrada ab



Exemplo



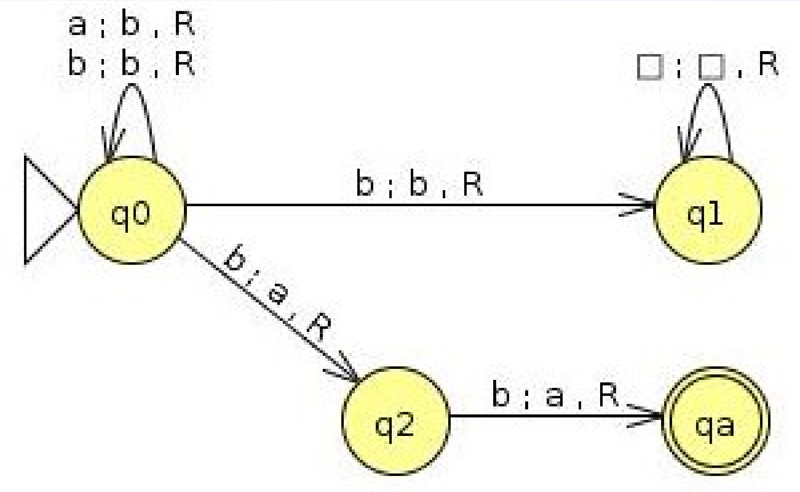
- $q_0ab \vdash bq_0b \vdash bbq_0B$
- $q_0ab \vdash bq_0b \vdash bbq_1B \vdash bbBq_1B$ ...
- $q_0ab \vdash bq_0b \vdash baq_2B$

Máquinas de Turing Não Determinísticas

- Uma máquina de Turing não determinística N aceita w se existe $q_0 w \vdash^* C$ em que C possui q_a
- N não aceita w se não existe $q_0 w \vdash^* C$ em que C tem q_a

Exemplo

- Seja N a máquina abaixo



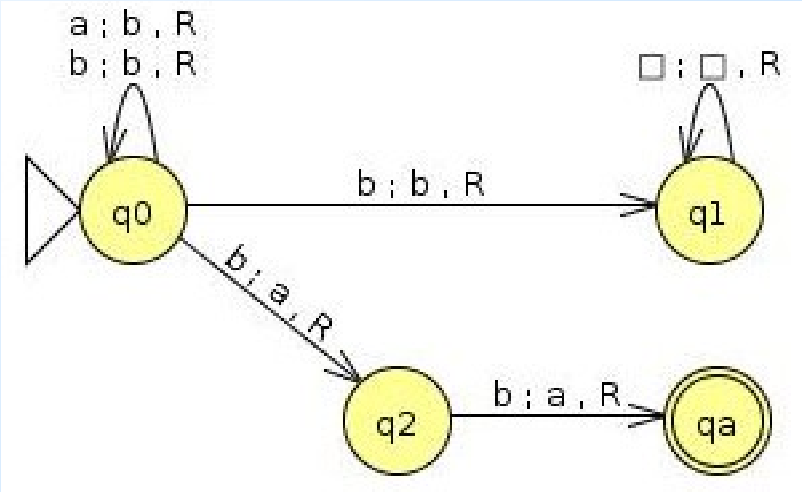
- N aceita abb
- N não aceita ab

Máquinas de Turing Não Determinísticas

- A linguagem de uma máquina de Turing não determinística N:
- $L(N) = \{w \mid \text{existe } q_0 w \vdash^* C \text{ em que } C \text{ tem } q_a\}$
- Seja B uma linguagem. Dizemos que N reconhece B quando L(N) = B.

Exemplo

- Seja N a máquina abaixo



- L(N) = {w ∈ {a,b}* | w possui a substring bb}

Máquinas de Turing Não Determinísticas

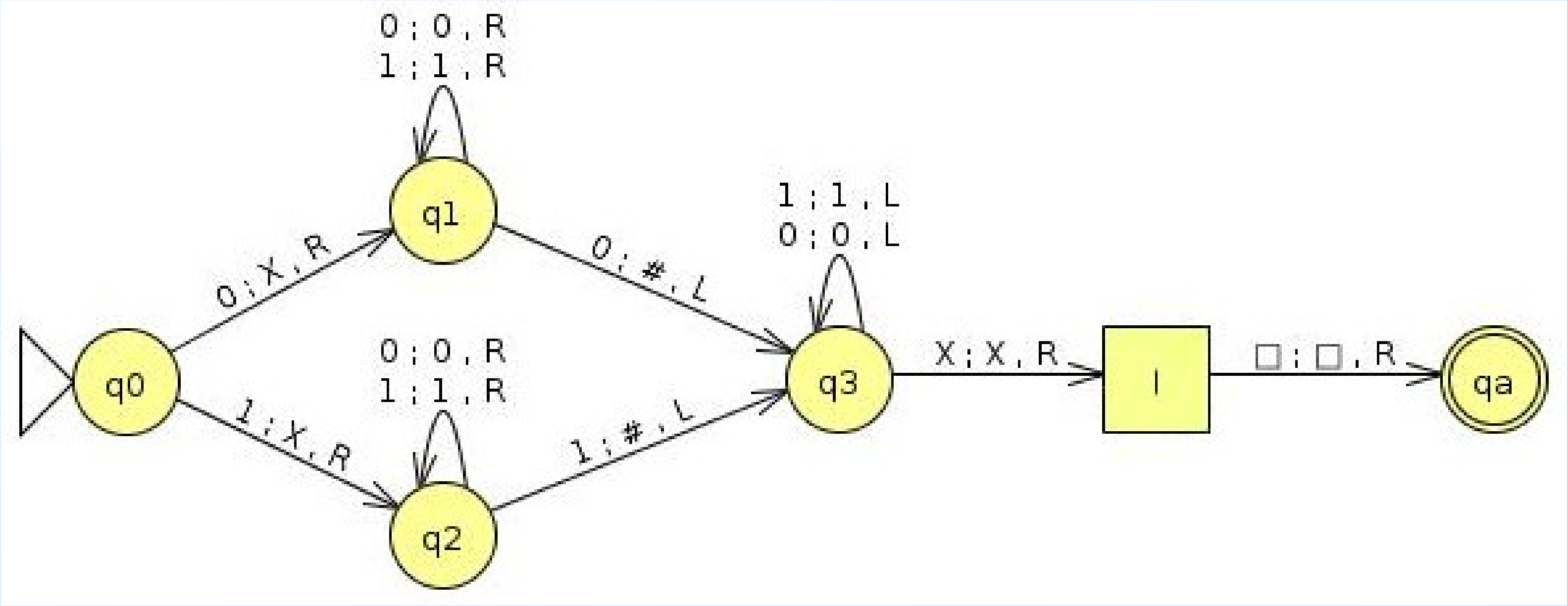
- Seja B uma linguagem. Dizemos que N decide B quando:
- L(N) = B
- w $\in$ B se e somente se N aceita w
- Quando N não aceita w, N pára em todas as execuções do não determinísmo

Exemplo

- Vamos definir uma máquina de Turing não determinística para decidir

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

Exemplo



- I decide {w#w | w ∈ {0,1}*}
- N decide L = {ww | w ∈ {0,1}*}

Teorema

- Para toda máquina de Turing não determinística N, existe uma máquina de Turing determinística D tal que L(N) = L(D).

Ideia da Prova

- Se alguma computação de N for de aceitação, D vai aceitar
- Testa em largura
- Simular N com D fazendo D testar cada uma das possibilidades de N
- Se nenhuma computação de N for de aceitação, D vai entrar em loop ou chegar em um estado de rejeição

Corolário

- Uma linguagem B é recursivamente enumerável se e somente se existe uma máquina de Turing não determinística N que reconhece B.

Corolário

- Uma linguagem B é recursiva se e somente se existe uma máquina de Turing não determinística N que decide B.