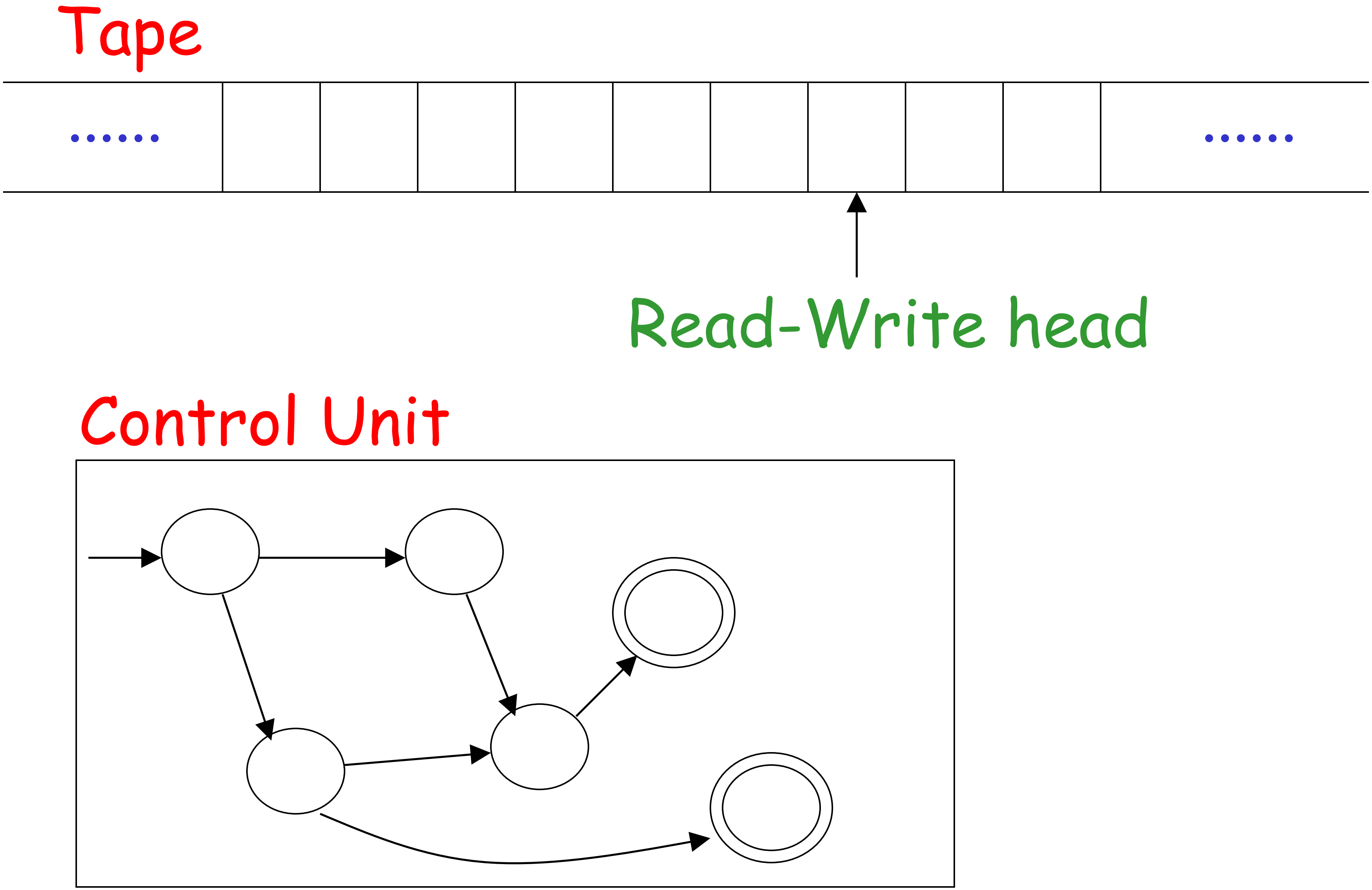
Walton Athletic Club, 1947

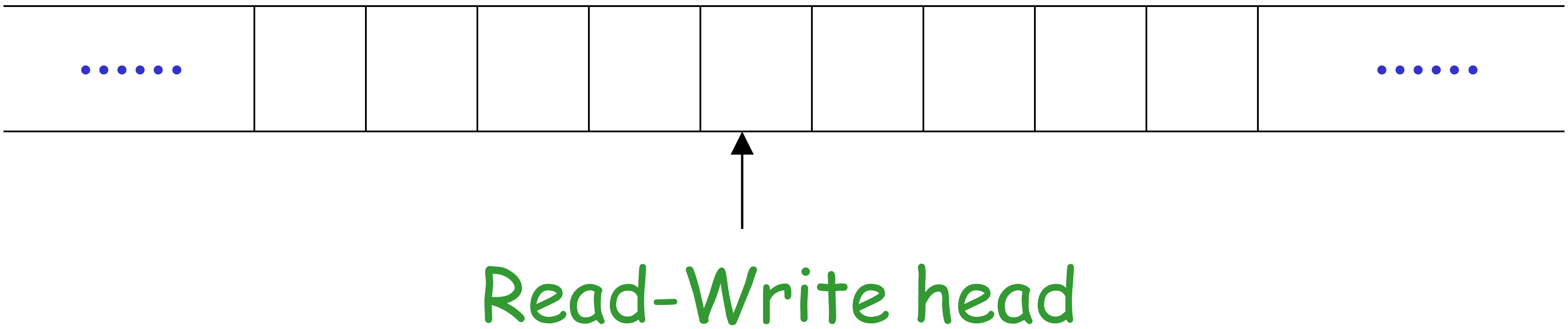
Macchine di Turing

Una macchina di Turing

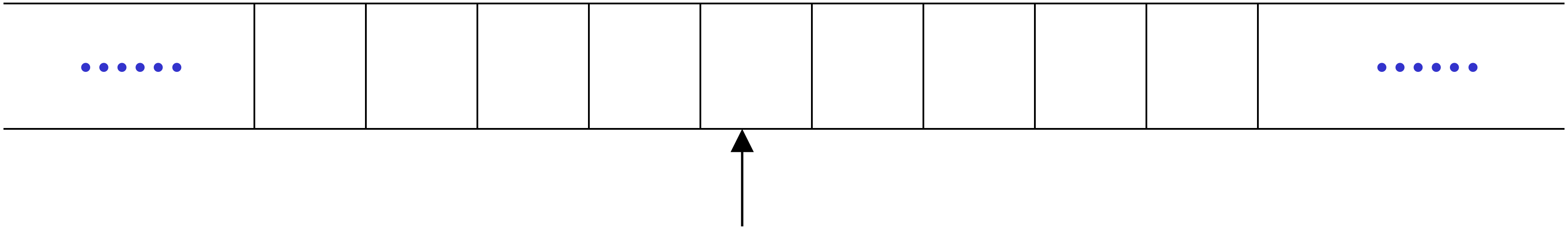


il Tape

No limiti - lunghezza potenzialmente infinita



Le testa si muove Left or Right



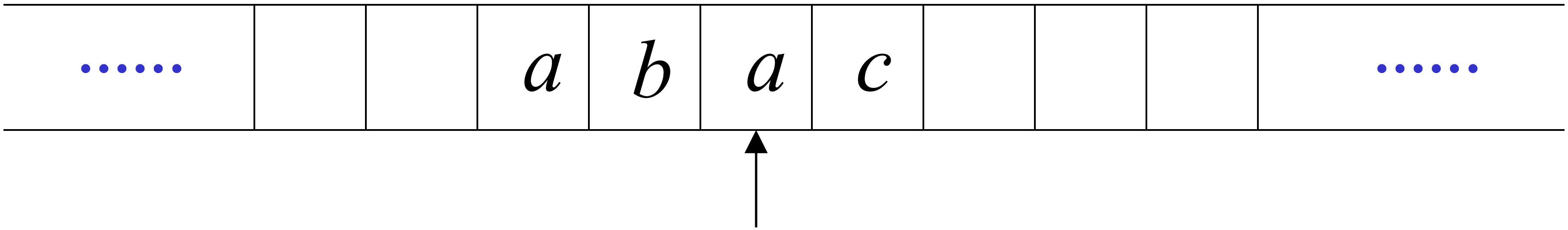
Read-Write head

la head ad ogni transizione (time step):

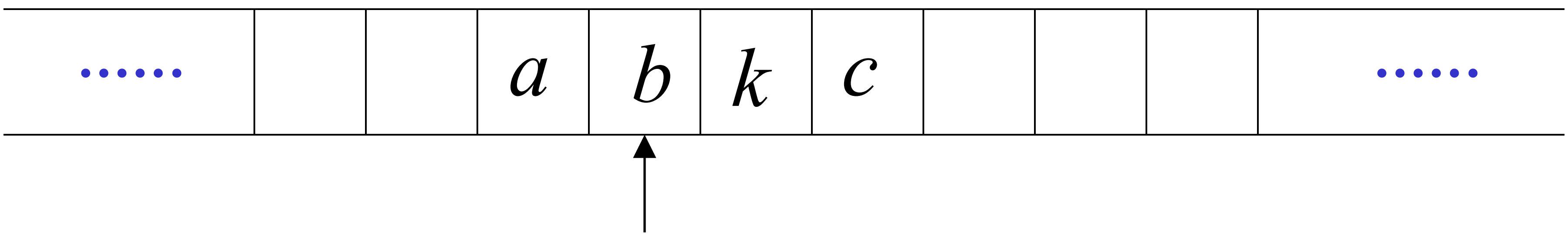
1. legge un simbolo
2. scrive un simbolo
3. si muove Left or Right

esempio:

Time 0

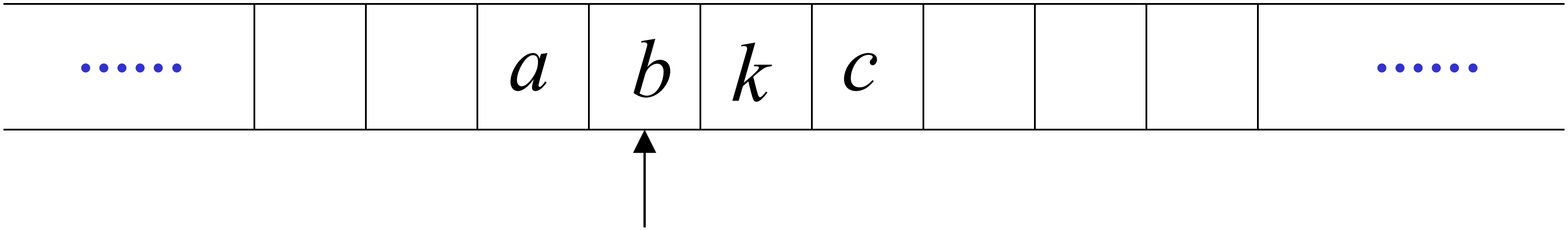


Time 1

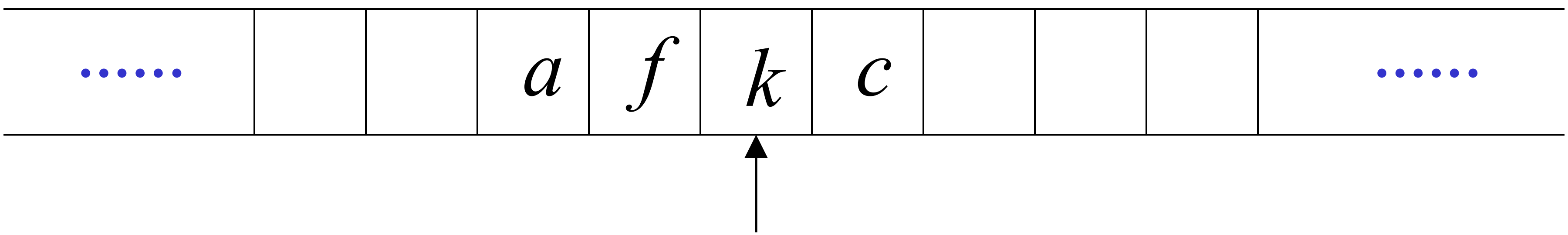


1. Reads a
2. Writes k
3. Moves Left

Time 1



Time 2

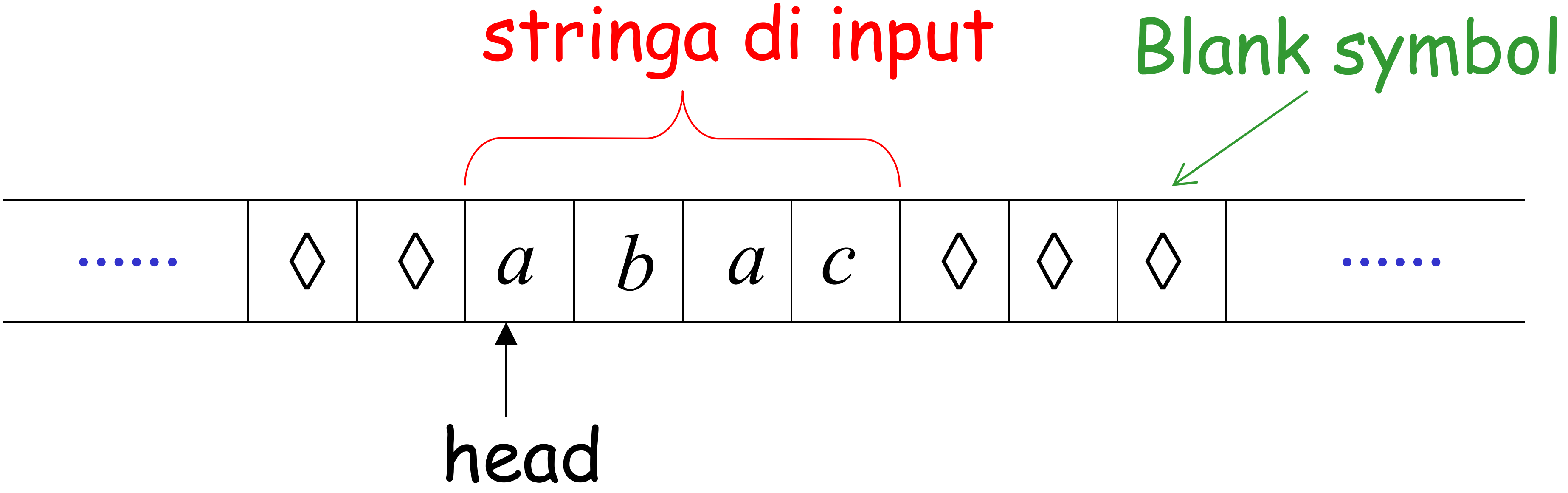


1. Reads b

2. Writes f

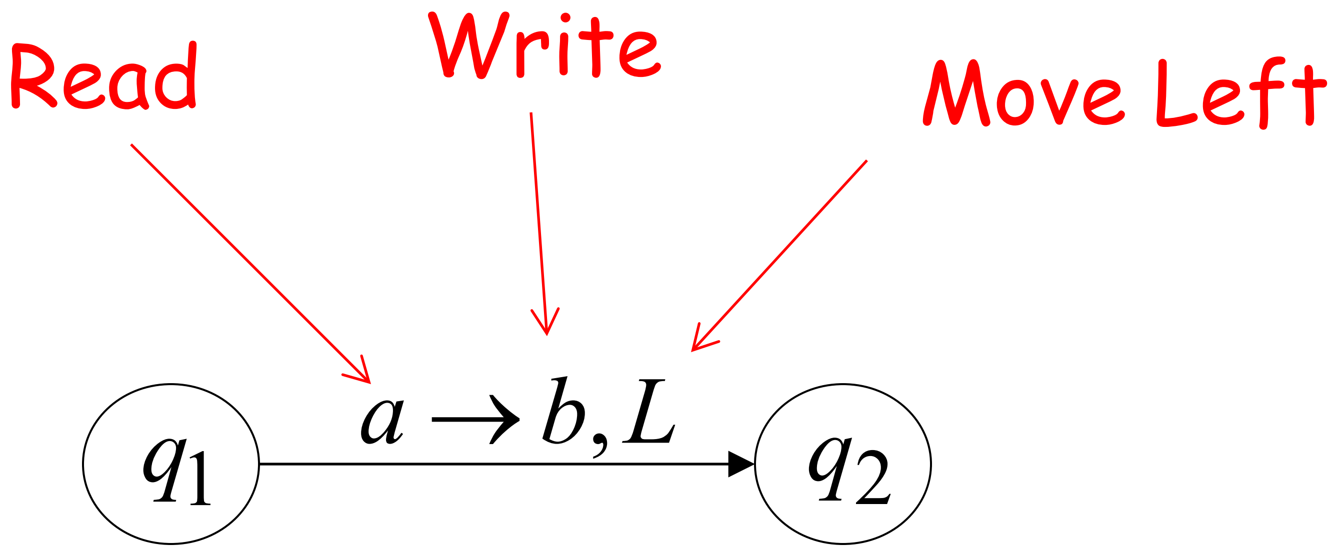
3. Moves Right

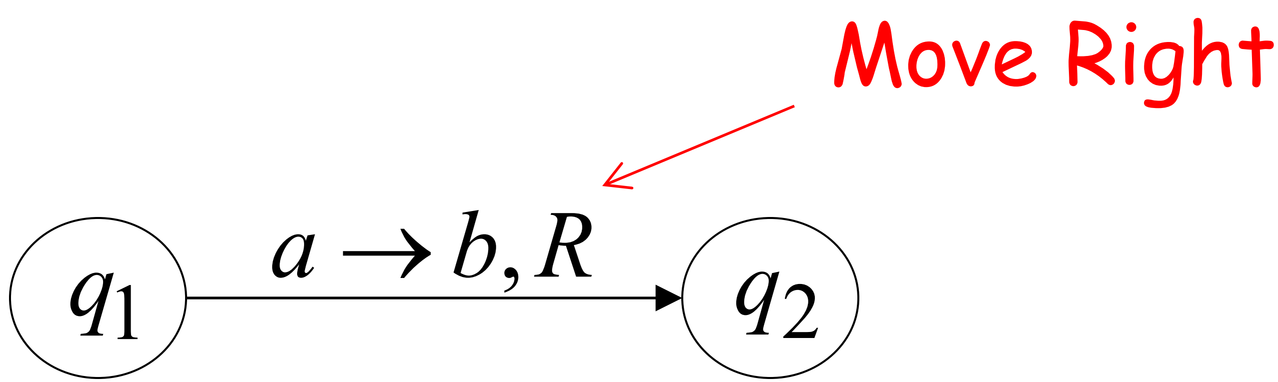
la String Input



Head parte dalla posizione più a sinistra della stringa di input

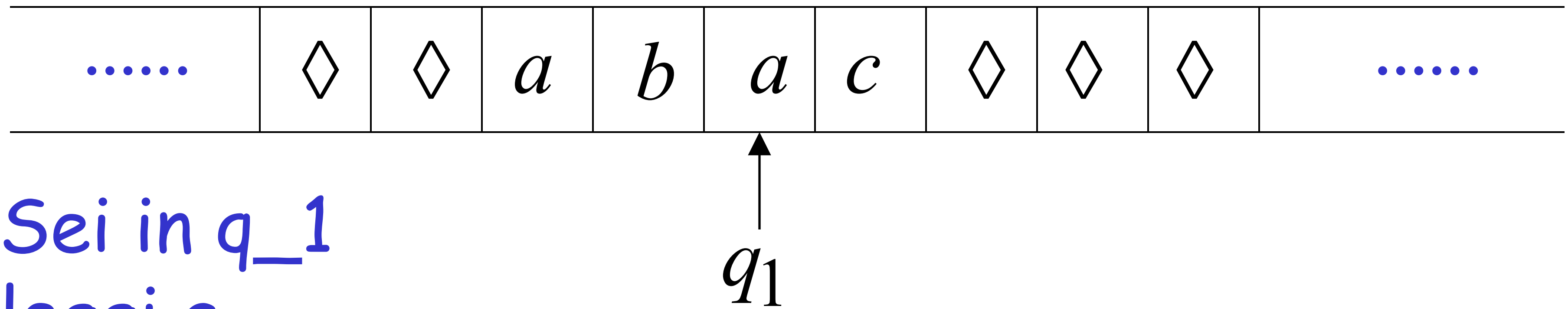
Stati & Transizioni





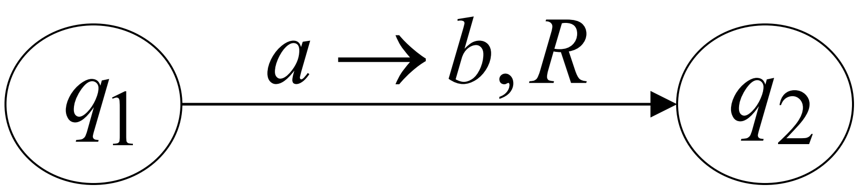
esempio:

Time 1

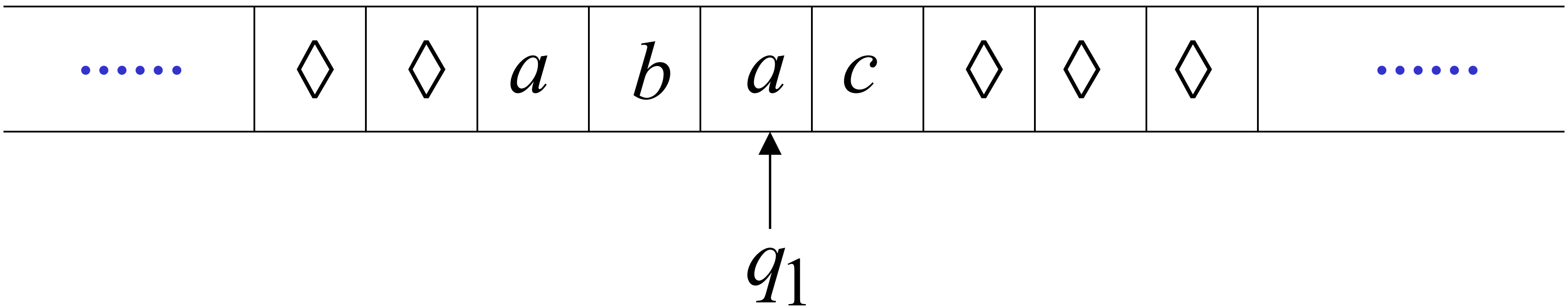


Sei in q_1
leggi a ,
cambia a in b
e vai a right

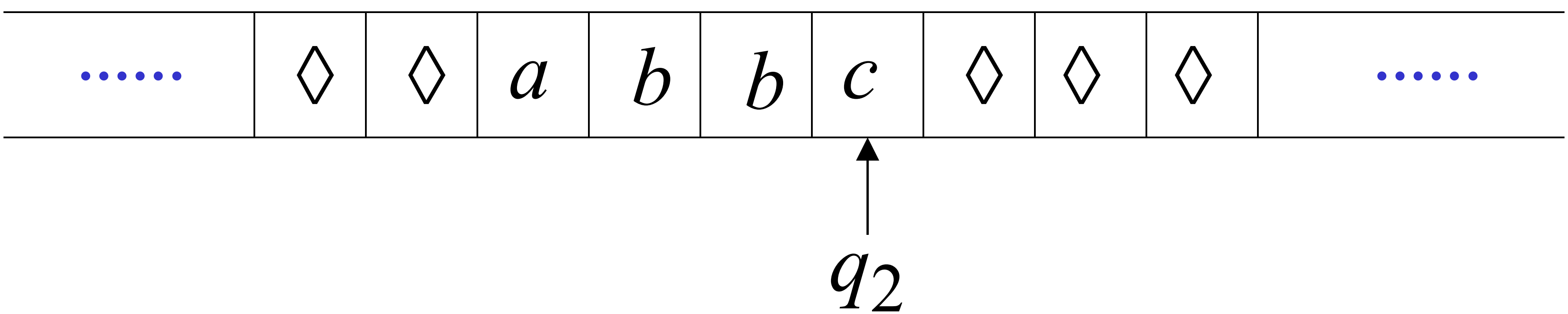
stato corrente

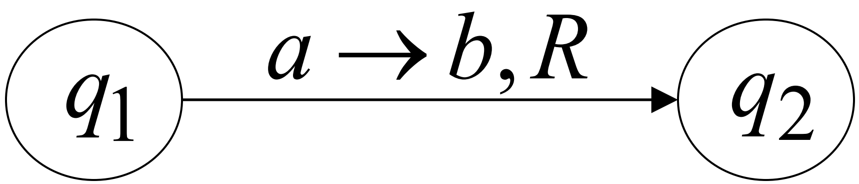


Time 1



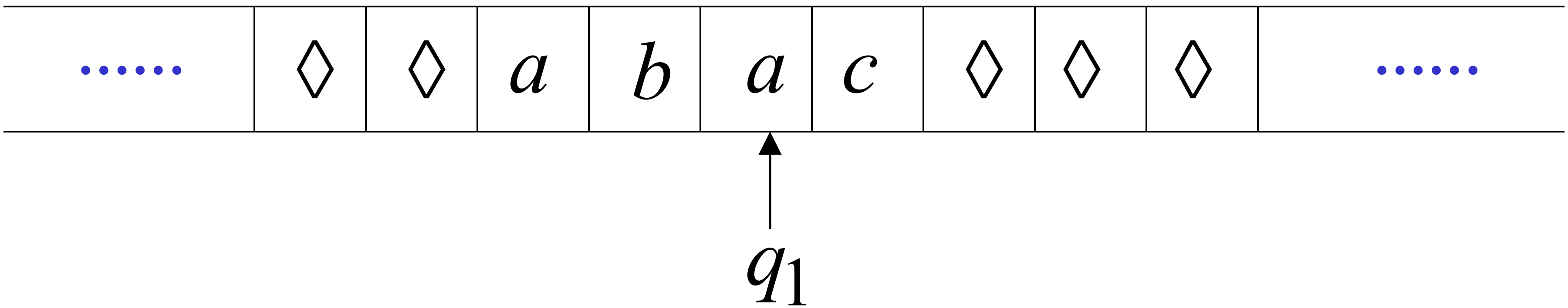
Time 2



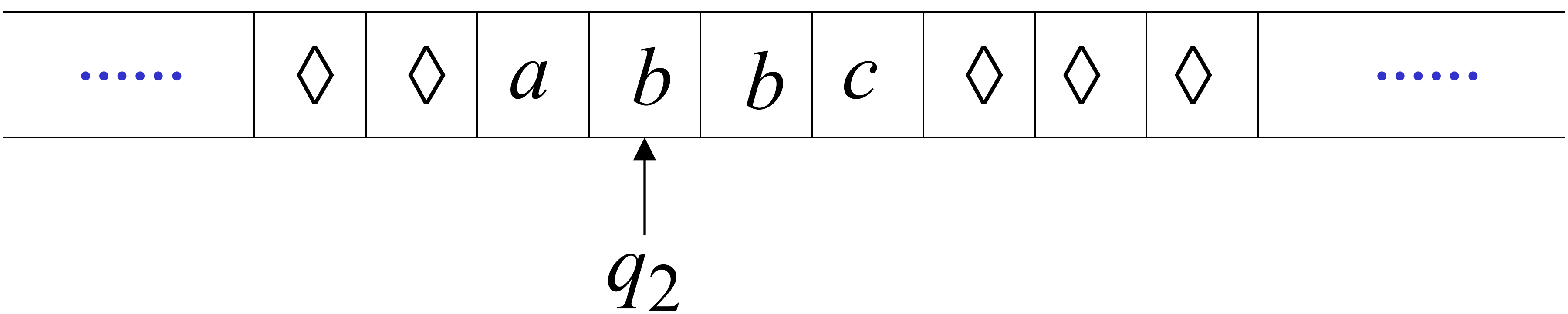


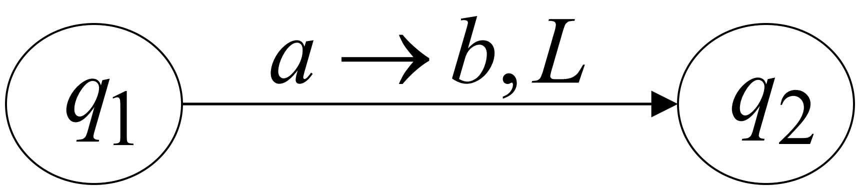
esempio:

Time 1



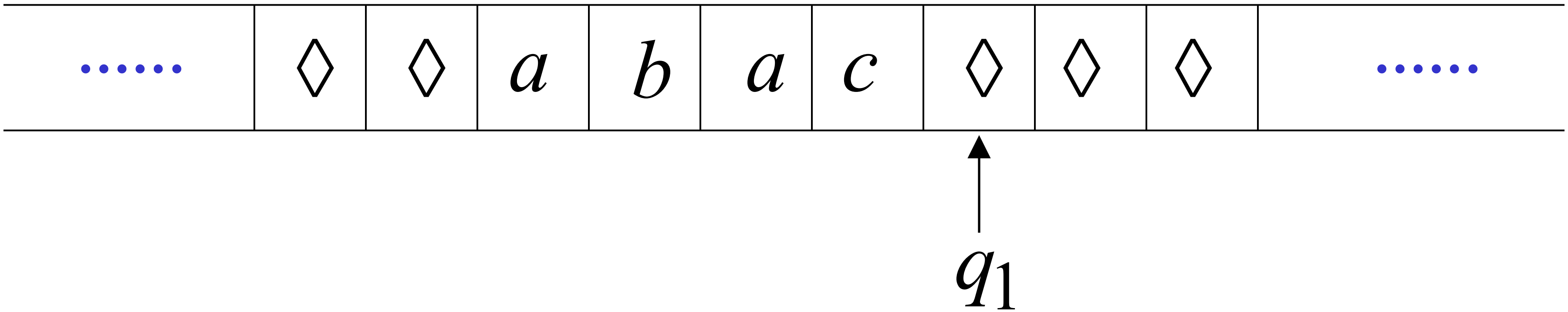
Time 2



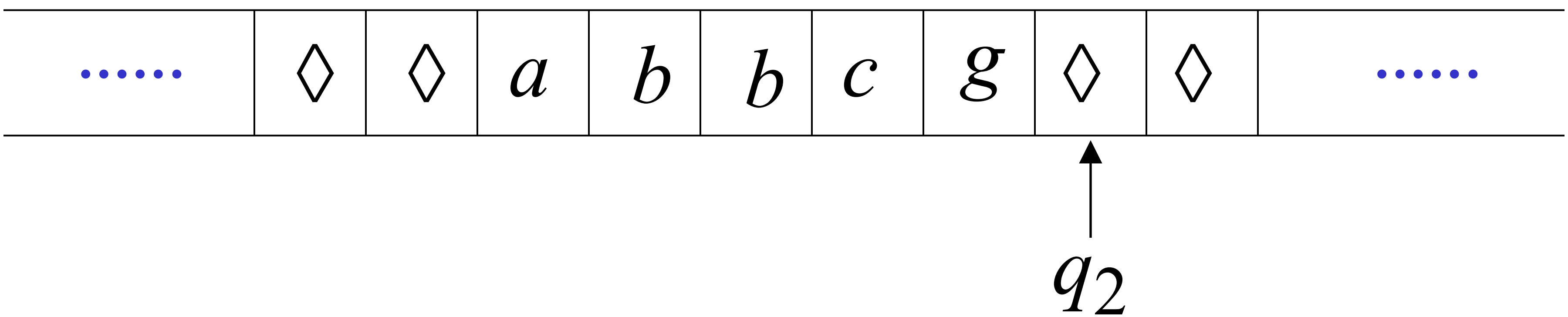


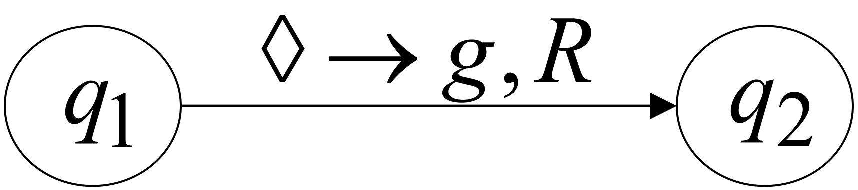
esempio:

Time 1



Time 2

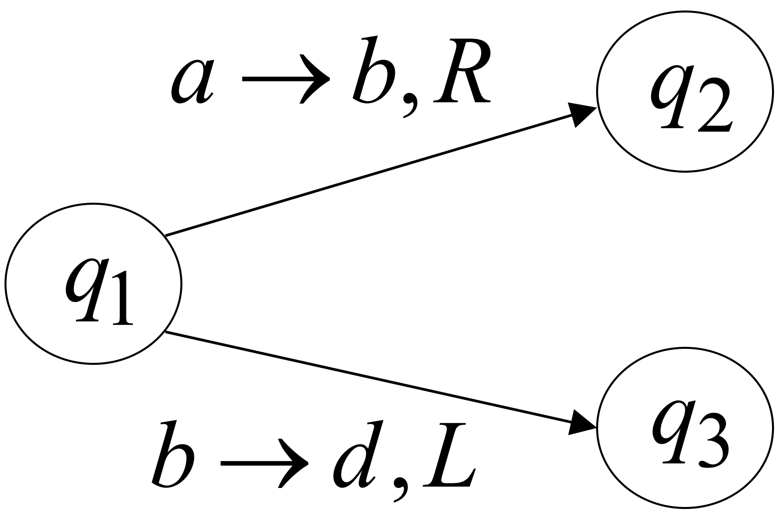




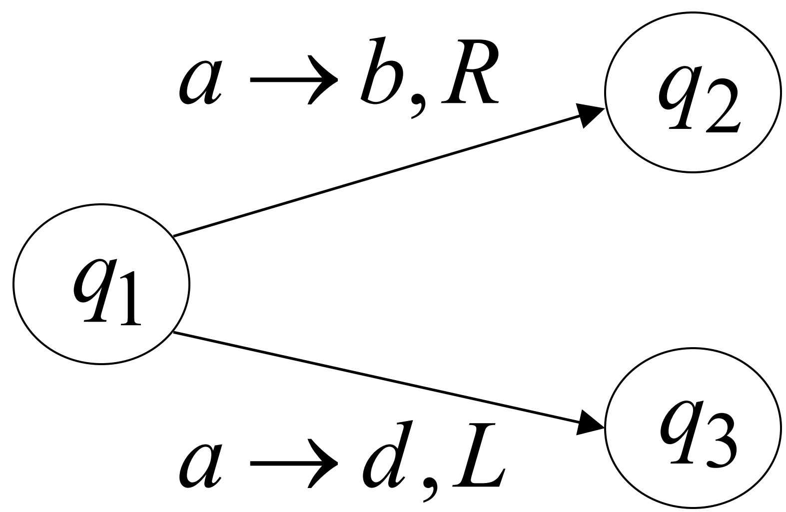
Determinismo

macchine di Turing sono deterministiche

<u>permesso</u>



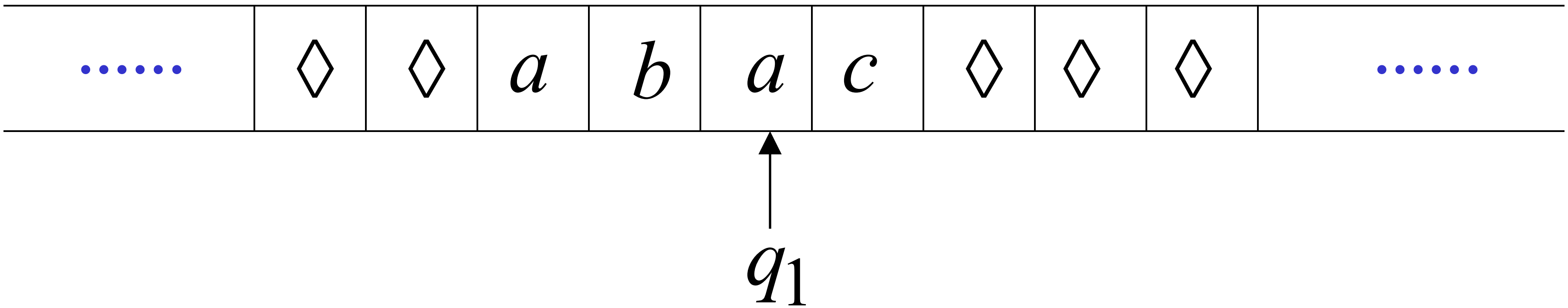
<u>Non permesso</u>

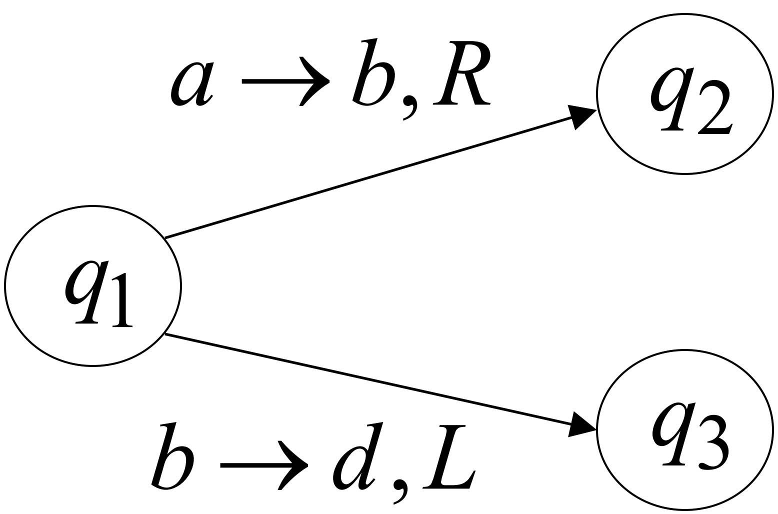


nessuna transizione lambda è permessa

Funzione di Transizione Parziale

esempio:





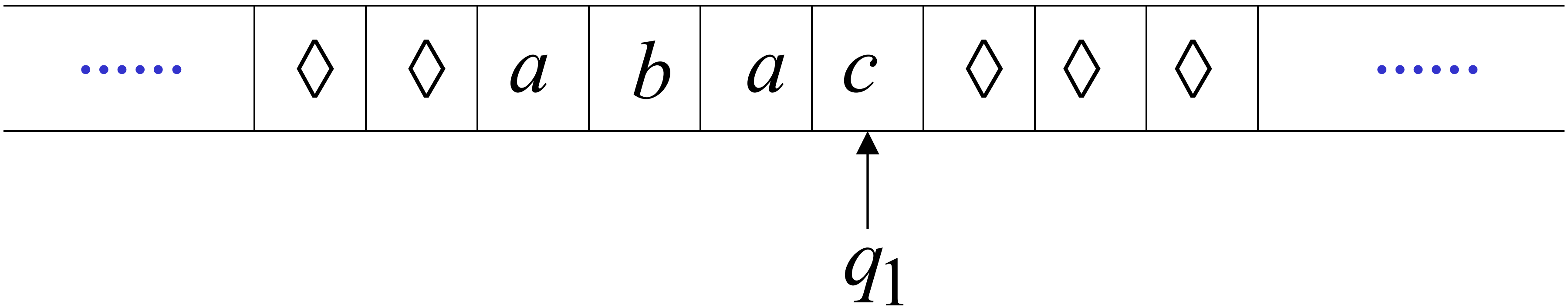
<u>permesso:</u>

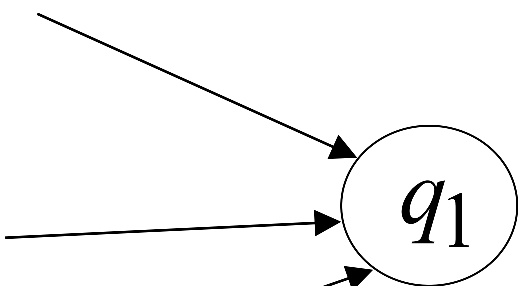
Nessuna transizione per simbolo input c

Halting

La macchina ***si ferma*** nello stato
in cui si trova
se non vi è nessuna transizione
da eseguire

Halt esempio 1:

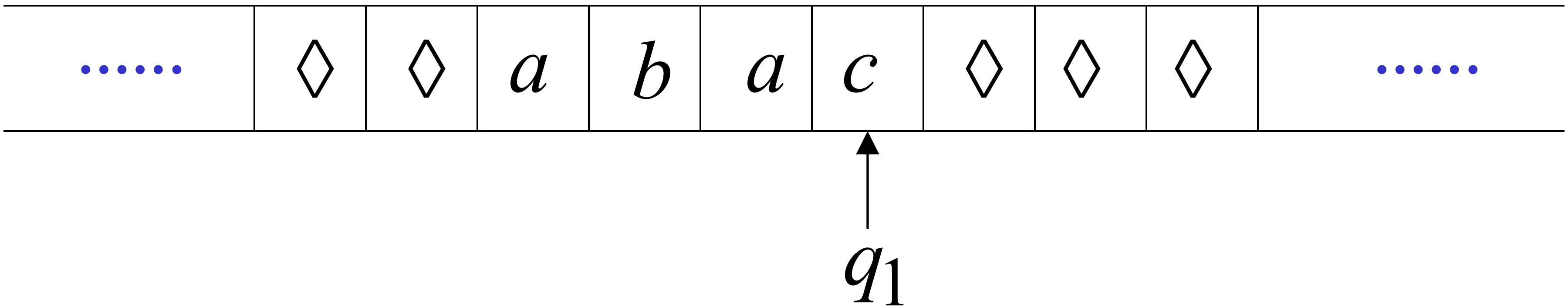


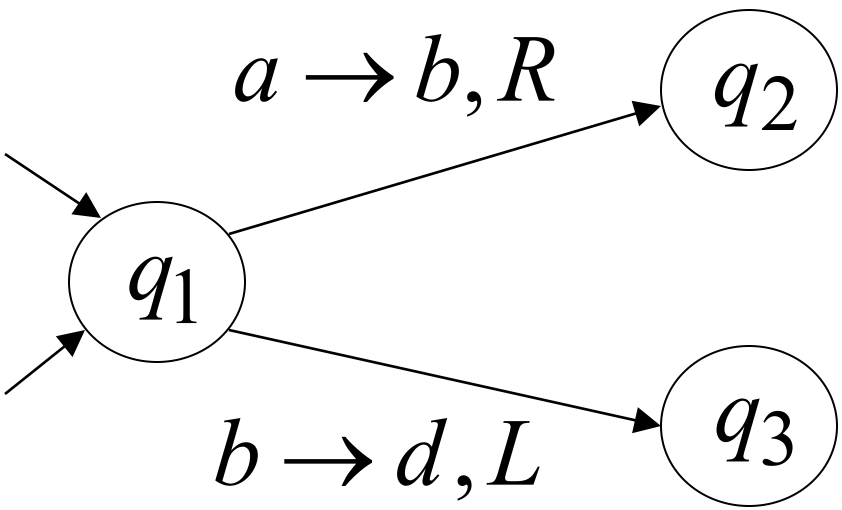


Nessuna transizione da q_1

HALT!!!

Halt esempio 2:

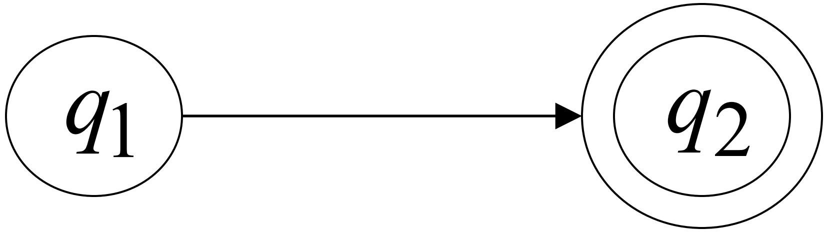




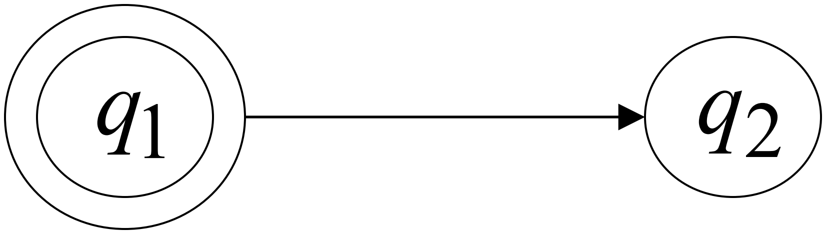
Nessuna transizione possibile da q_1 sul simbolo c

HALT!!!

Stati di accettazione



permesso



Non permesso

- Stati di accettazione non hanno transizioni in uscita
- La macchina si ferma e accetta.

Accettazione

Accettare stringa

In Input

se macchina si ferma in uno stato di accettazione

RIGETTARE stringa

In Input

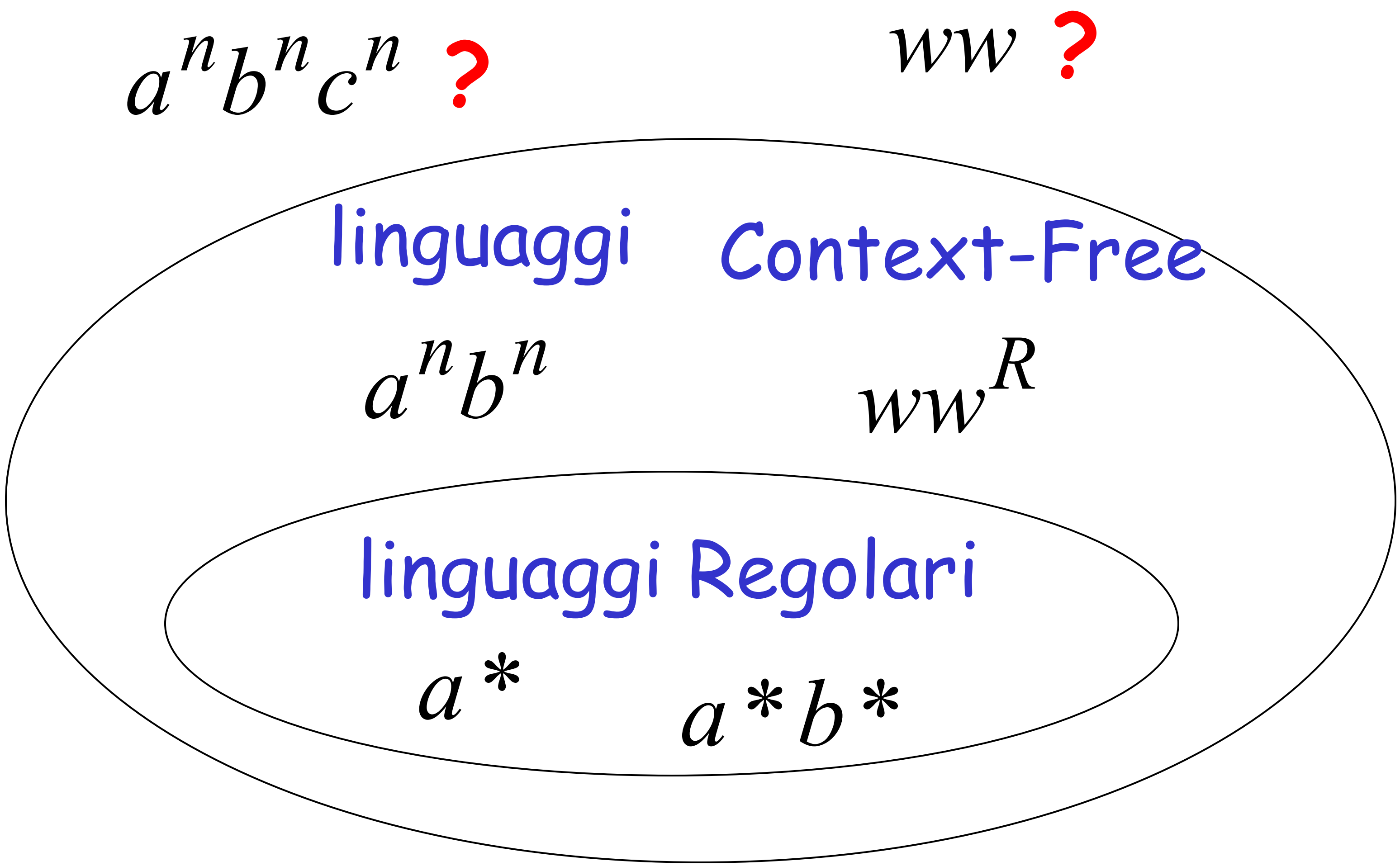
se macchina si ferma in uno stato di NON- accettazione o

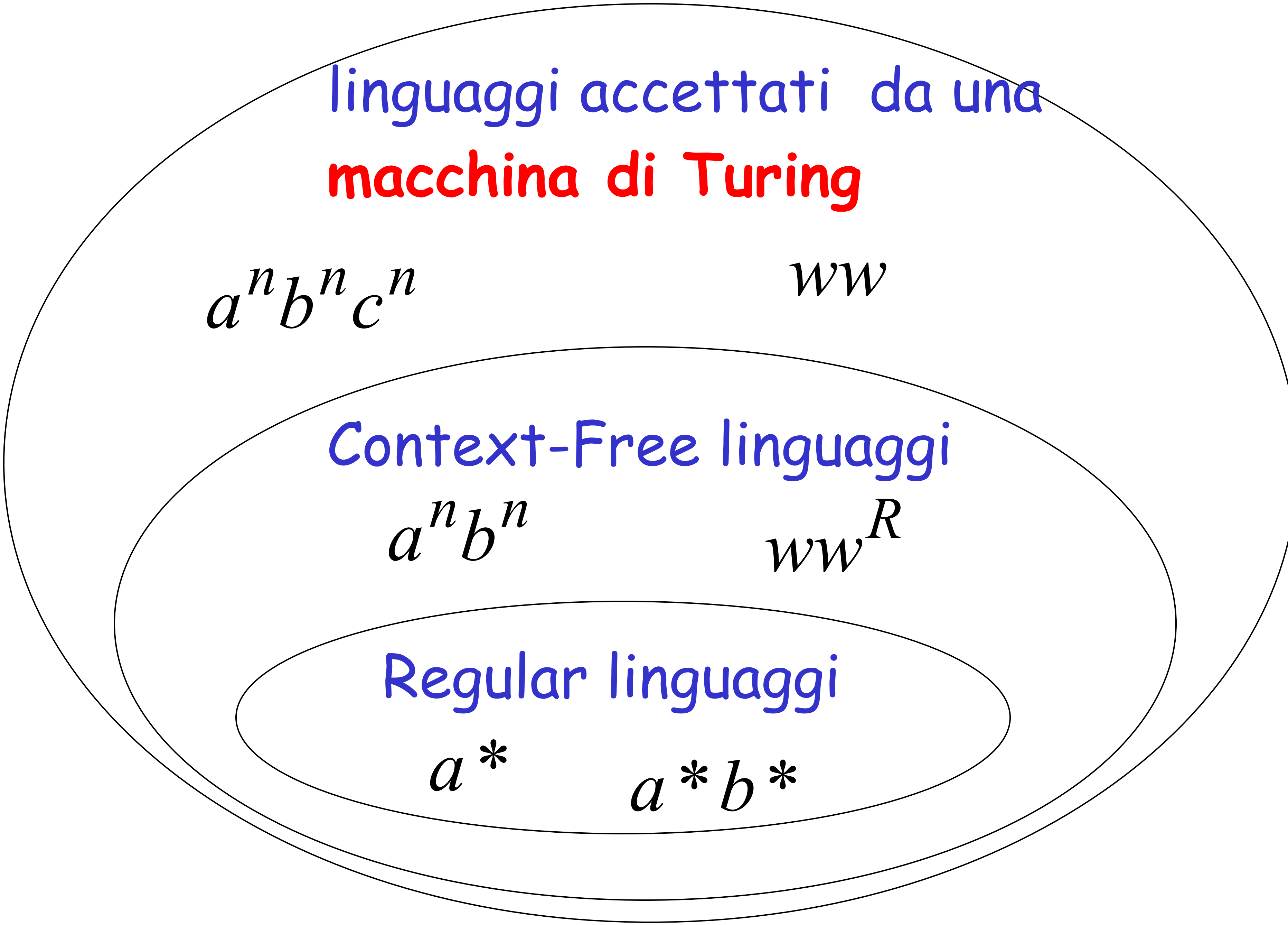
se la macchina entra in un *infinite loop*

Osservazione:

Nell'accettare una stringa di input,
non è necessario esaminare tutti
i simboli nella stringa.

La gerarchia dei linguaggi

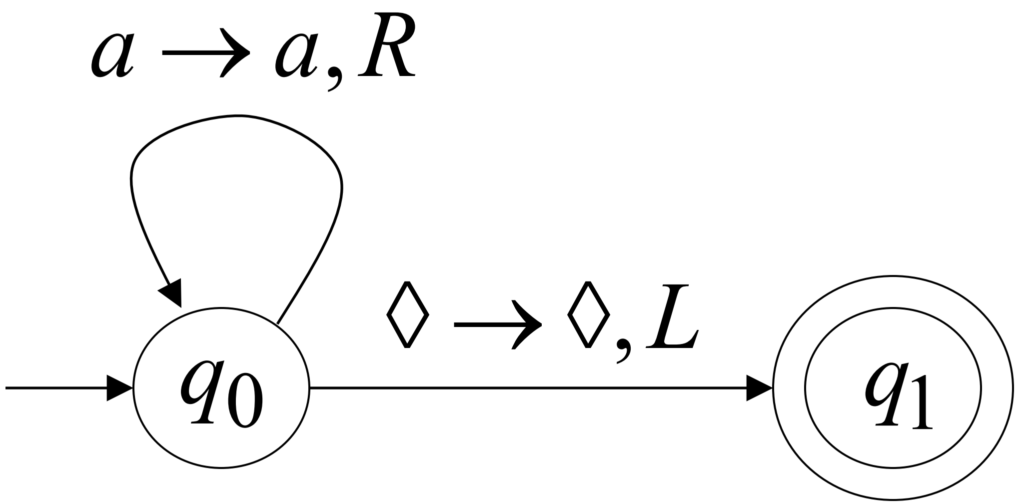



linguaggi accettati da una
macchina di Turing
$a^n b^n c^n$
ww
Context-Free linguaggi
$a^n b^n$
ww^R
Regular linguaggi
a^*
$a^* b^*$

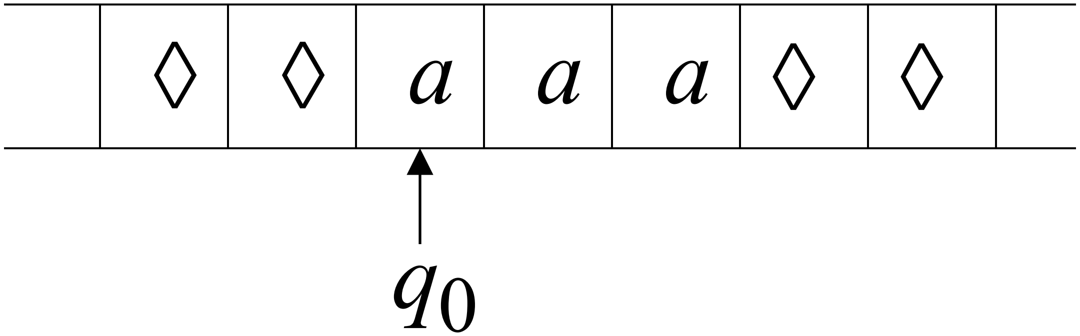
macchinaTuring esempio

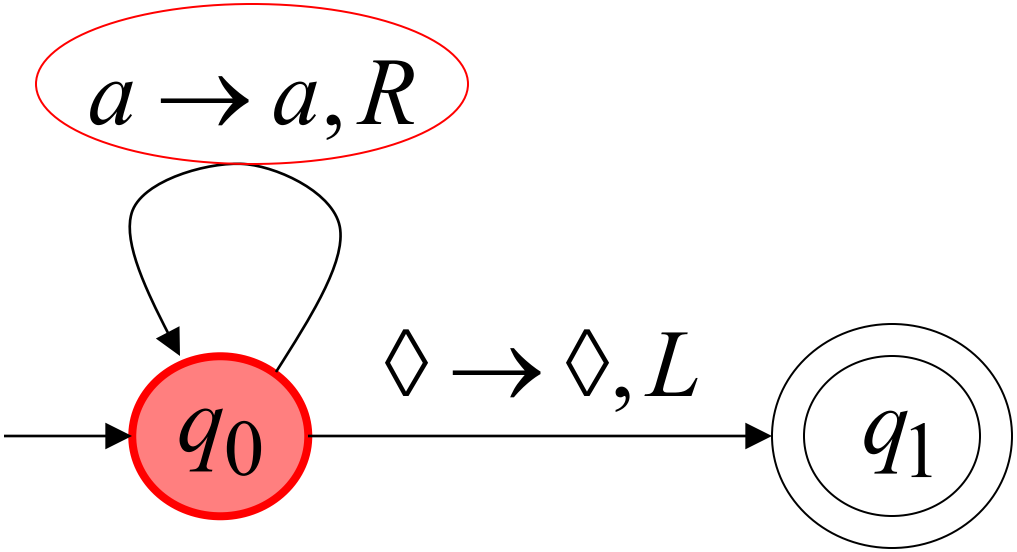
Input alphabet $\Sigma = \{a, b\}$

Accetta il linguaggio: a^*

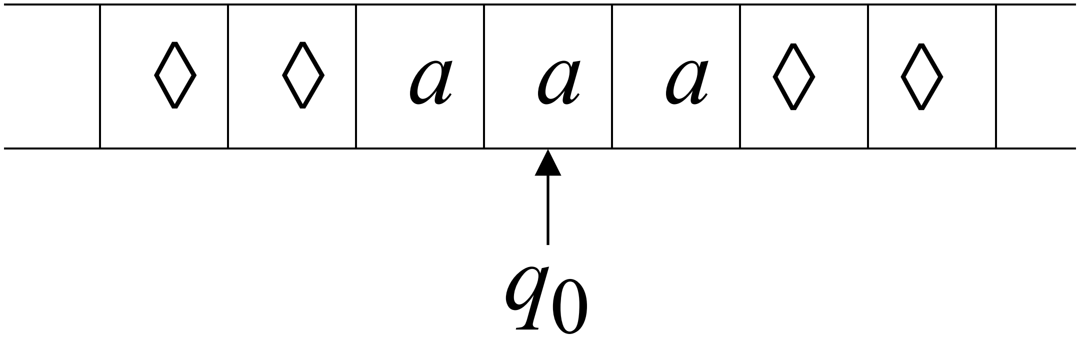


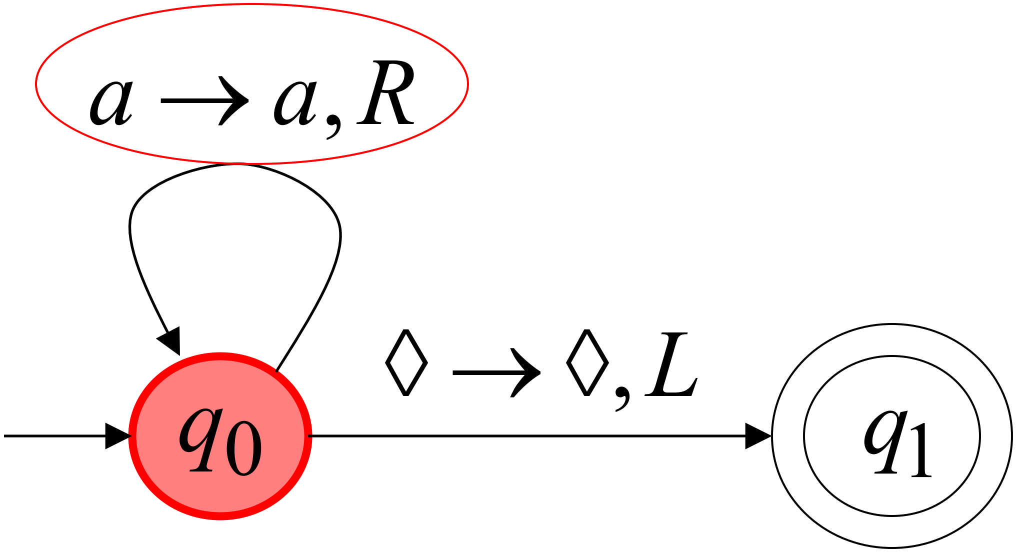
Time 0



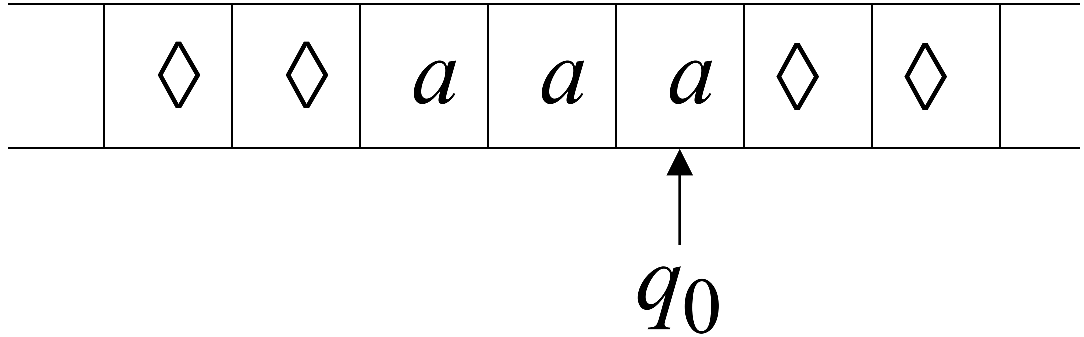


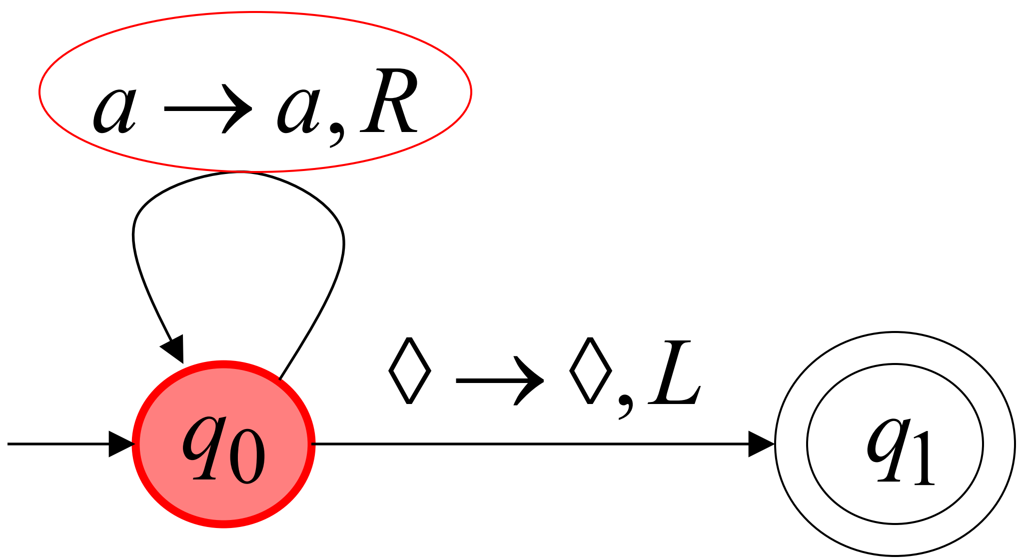
Time 1



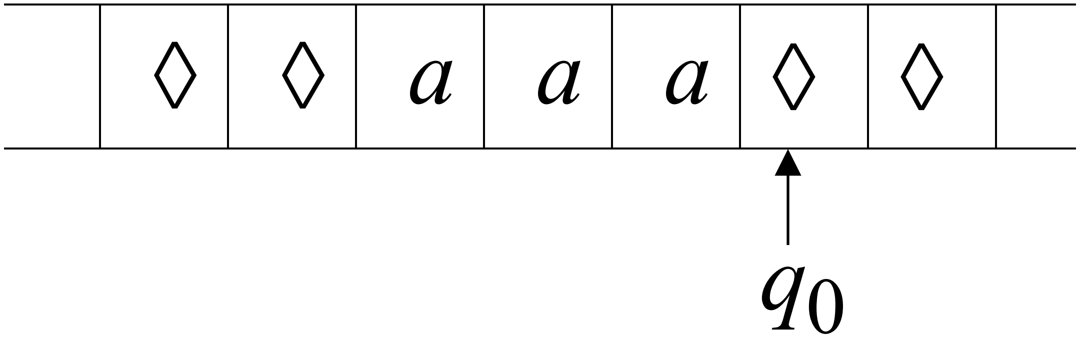


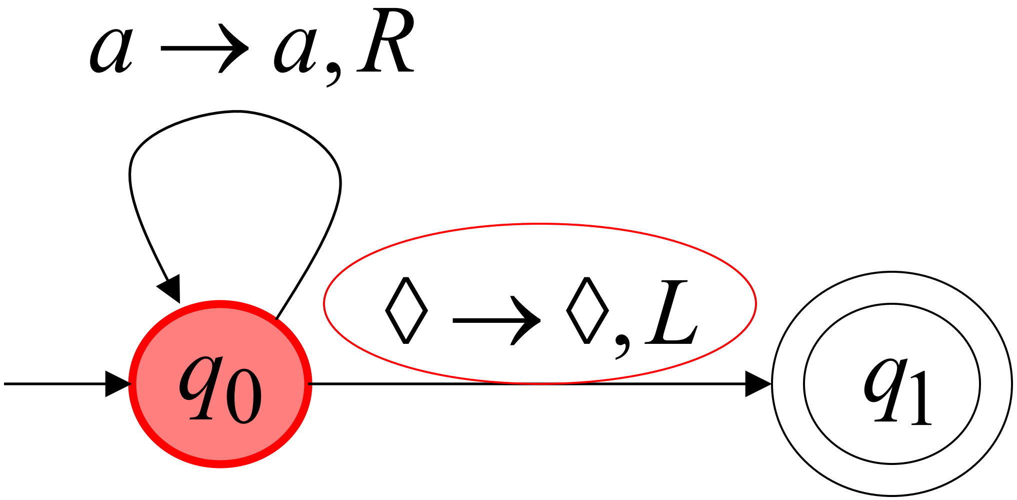
Time 2



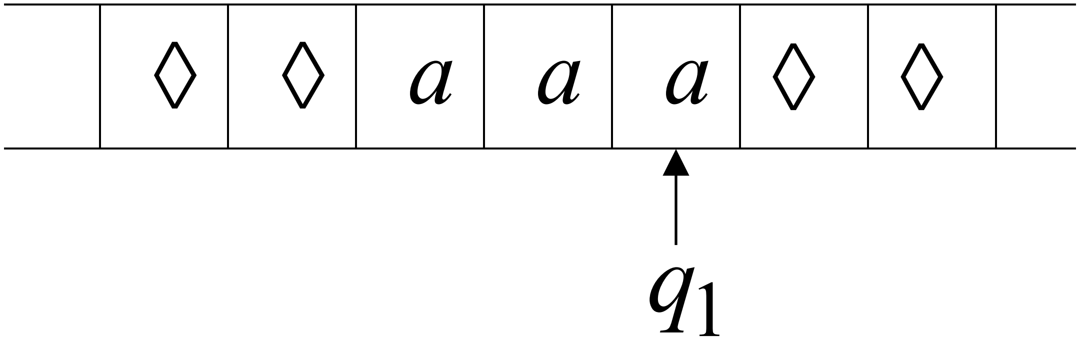


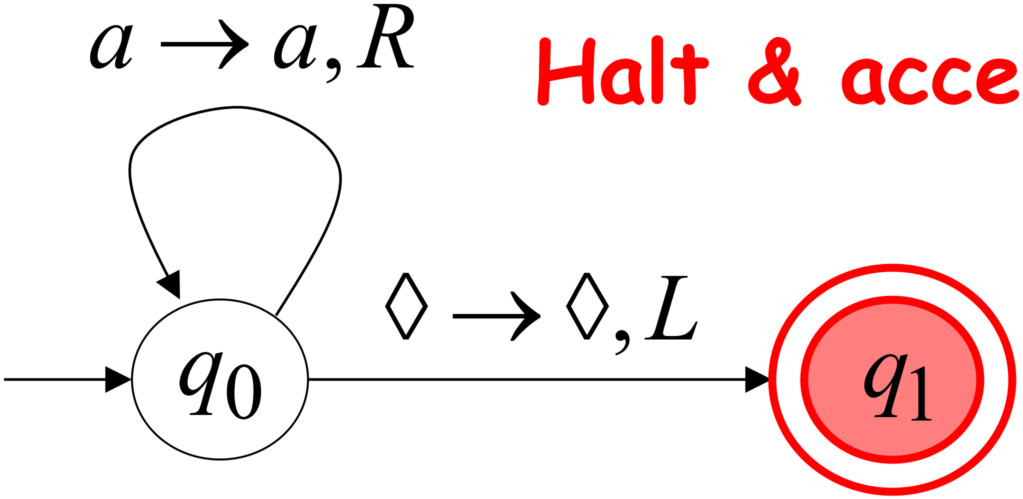
Time 3





Time 4

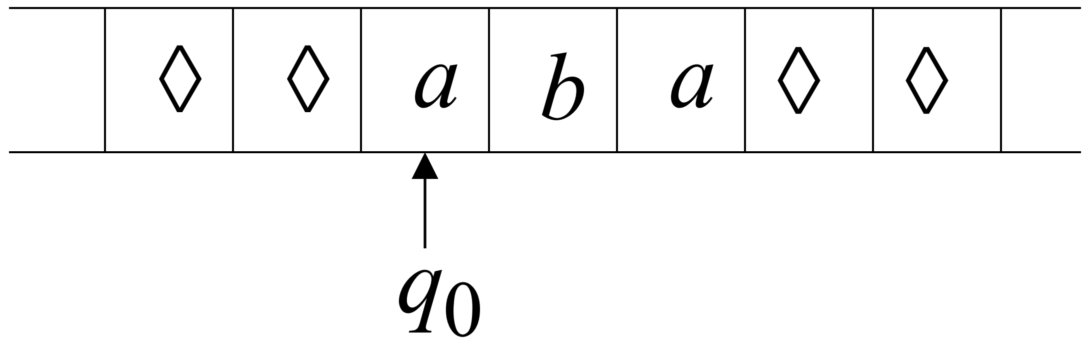


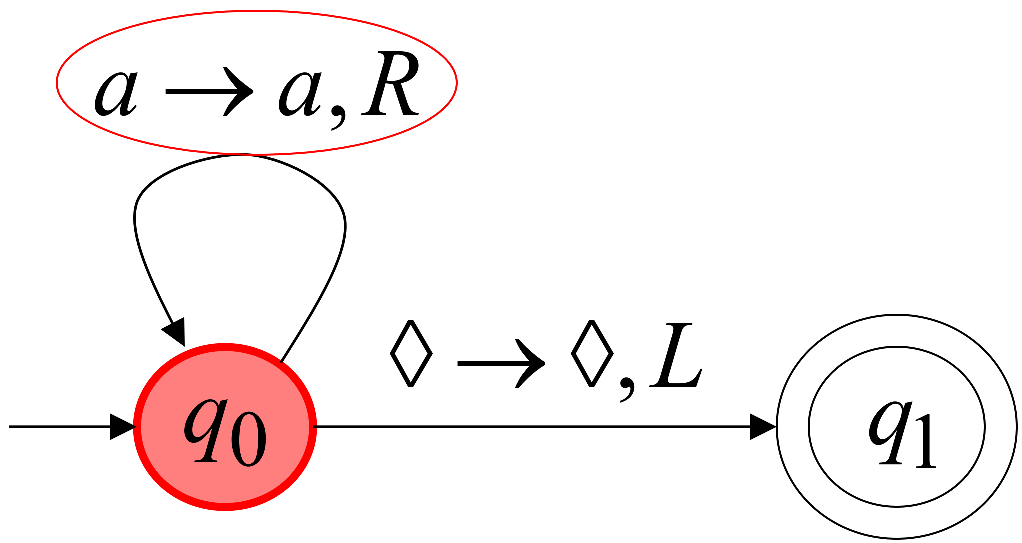


Halt & accettazione

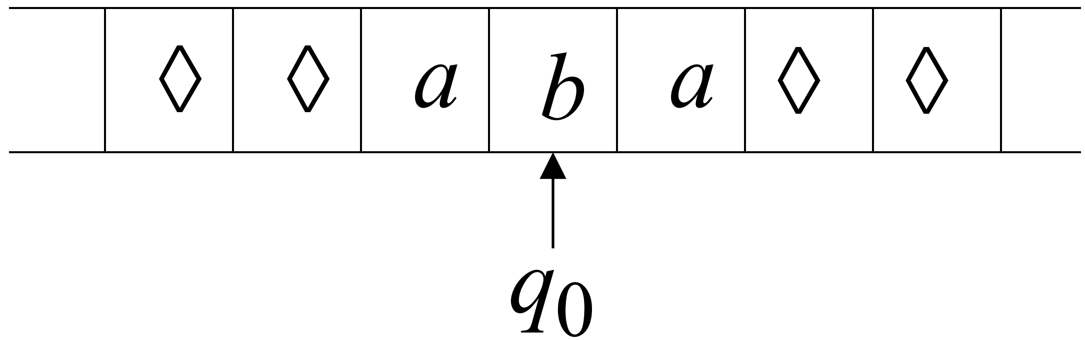
Rejection esempio

Time 0



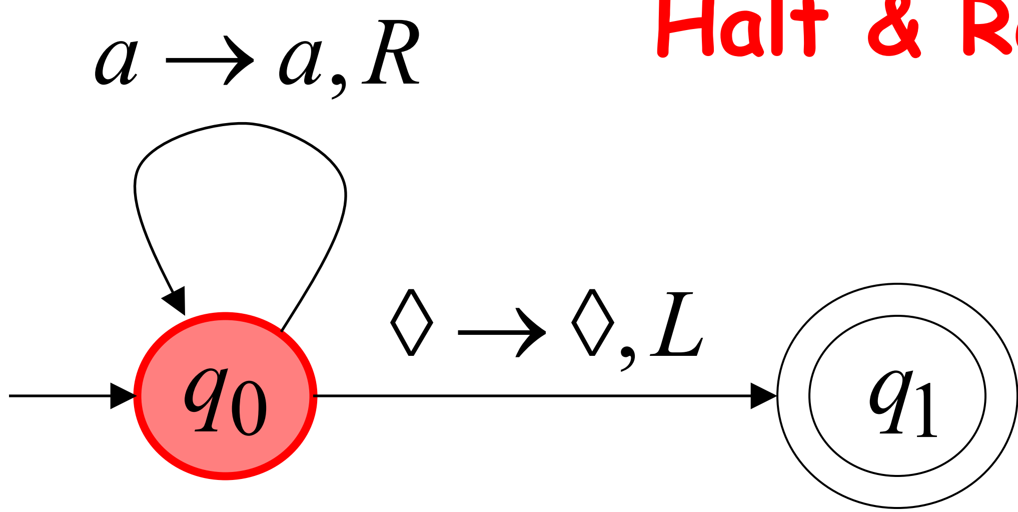


Time 1



Nessuna transizione

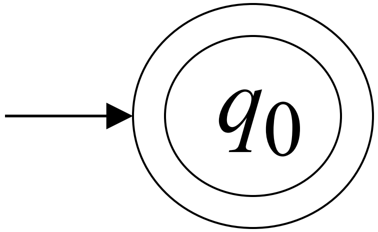
Halt & Reject



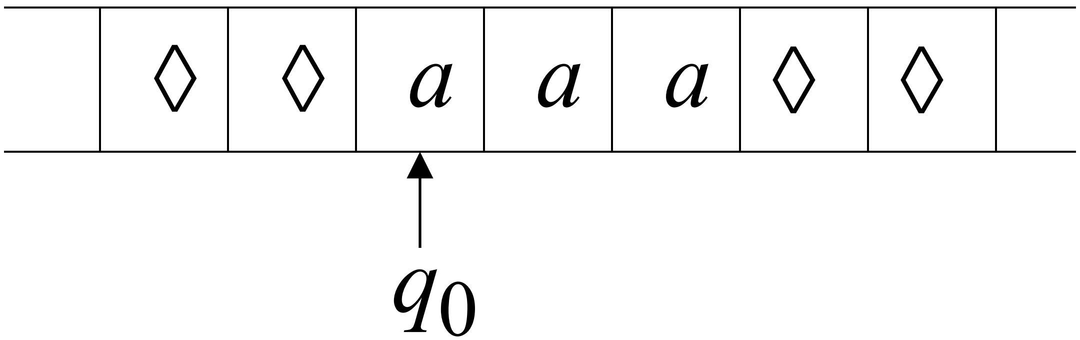
Una macchina semplice per linguaggio a^*

Ma con alfabeto di input $\Sigma = \{a\}$

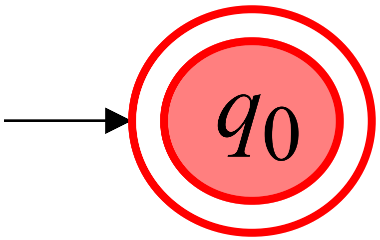
Accetta il linguaggio: a^*



Time 0



Halt & accettazione

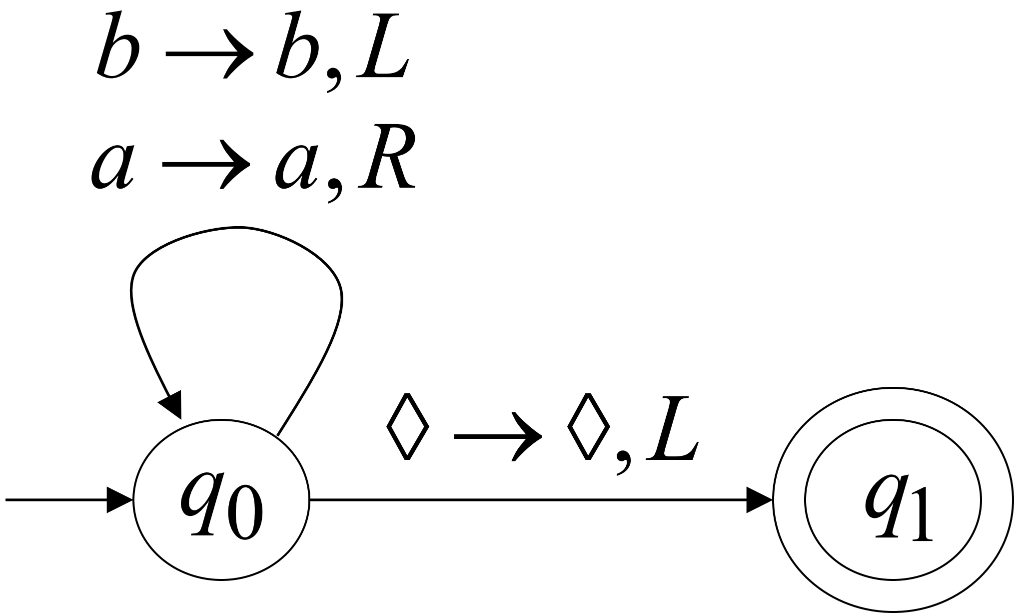


Non è necessario esaminare l'input

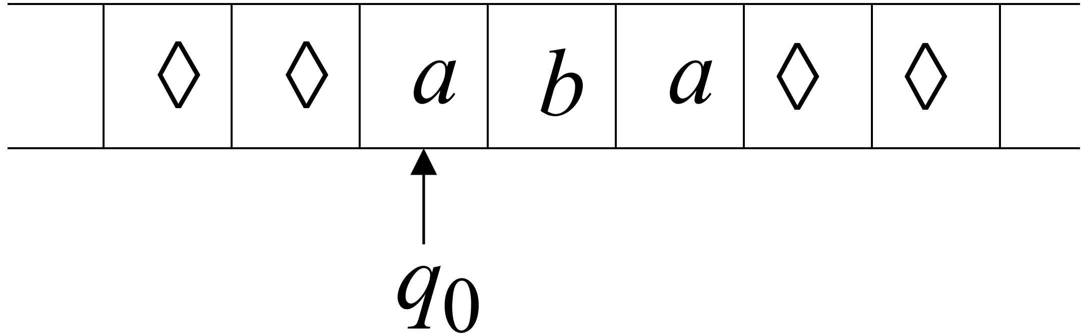
esempio Infinito Loop

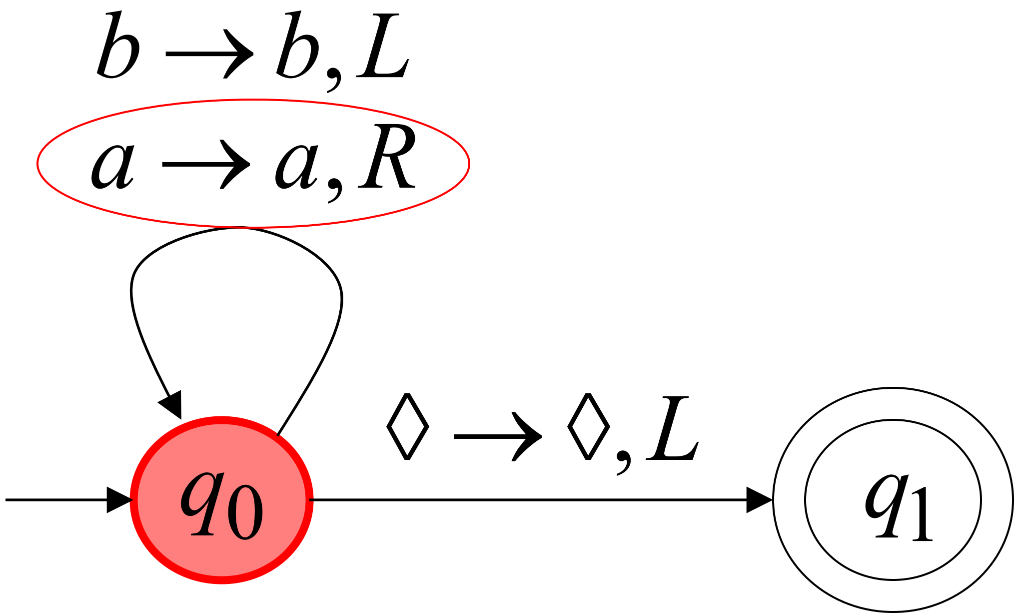
una macchina di Turing

Per il linguaggio $a^* + b(a+b)^*$

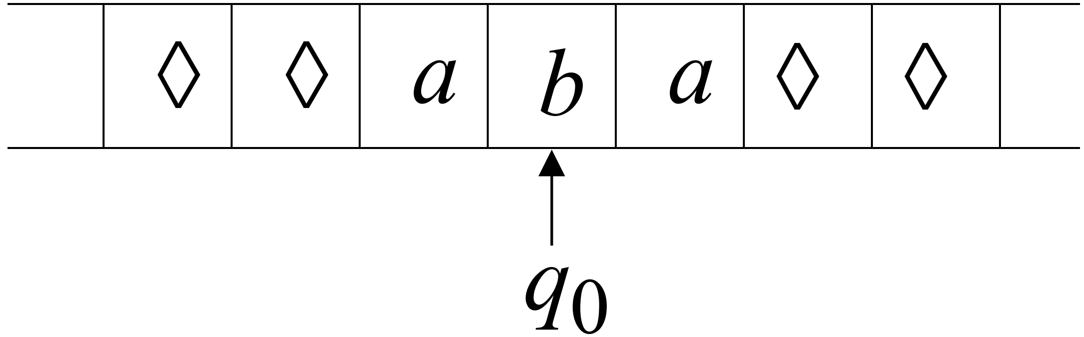


Time 0

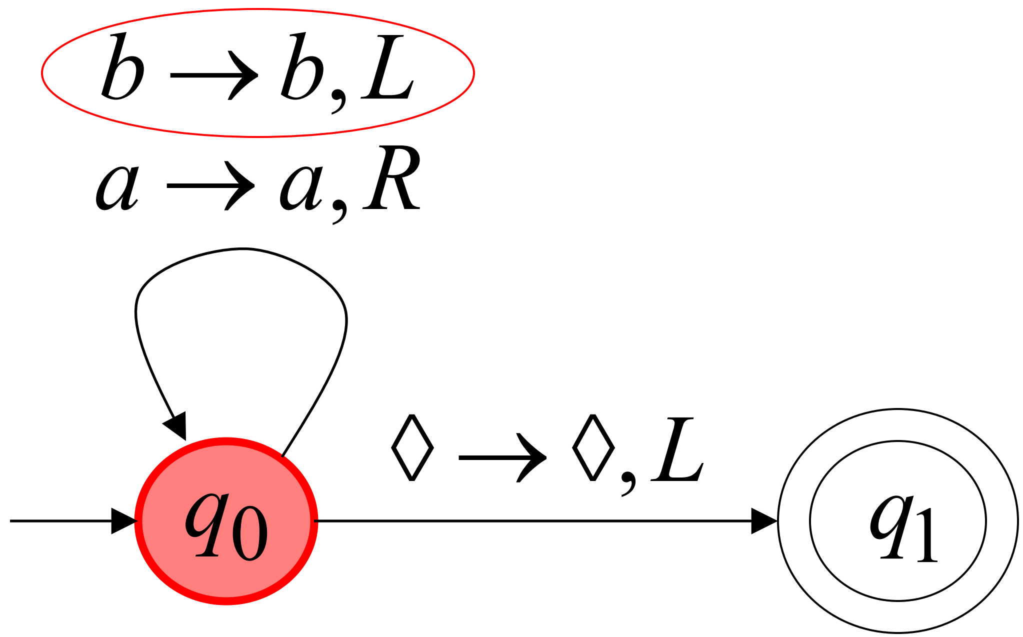

◊ ◊ a b a ◊ ◊
q_0


$b \rightarrow b, L$
$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

Time 1

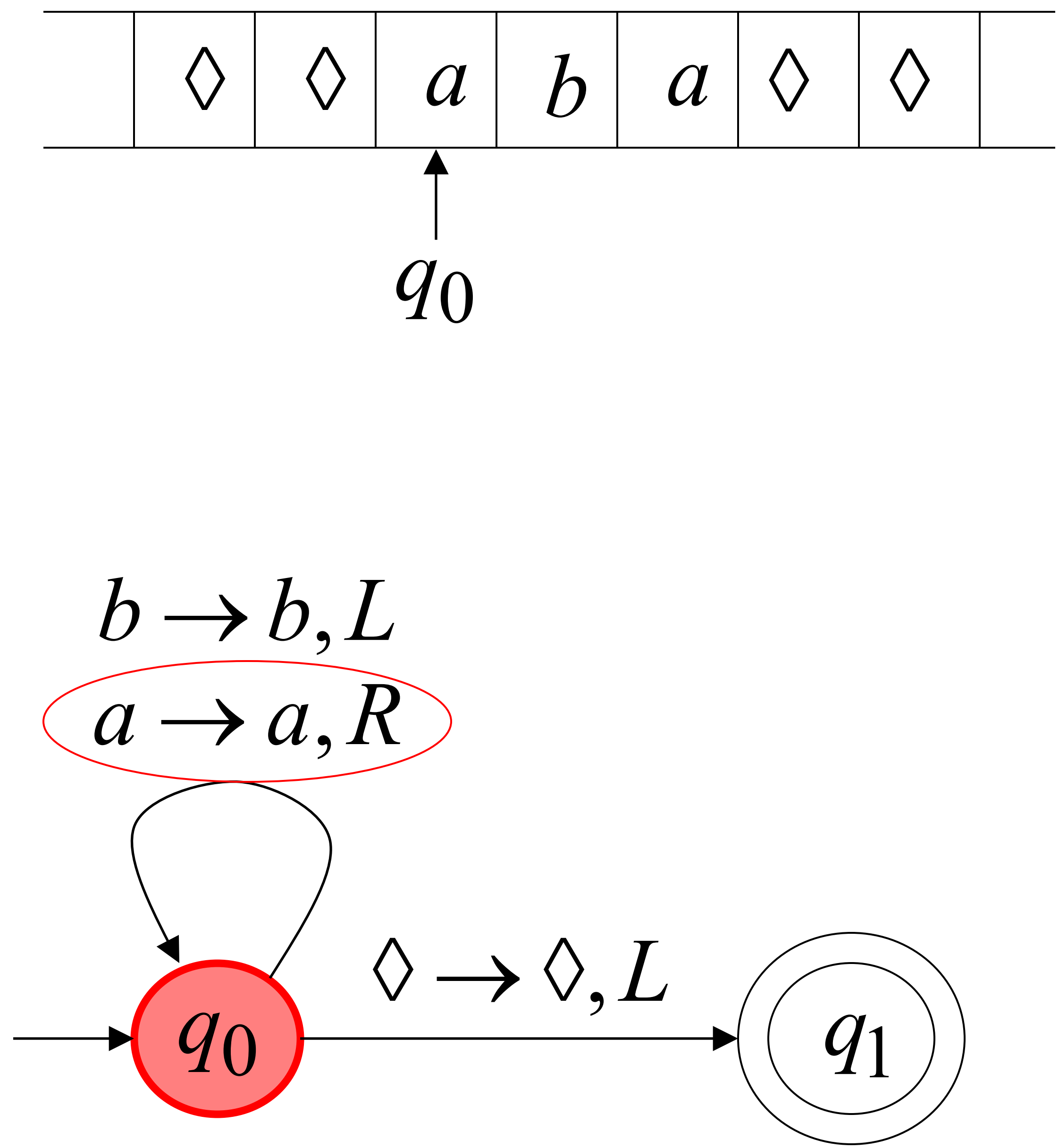


◊ ◊ a b a ◊ ◊
q_0

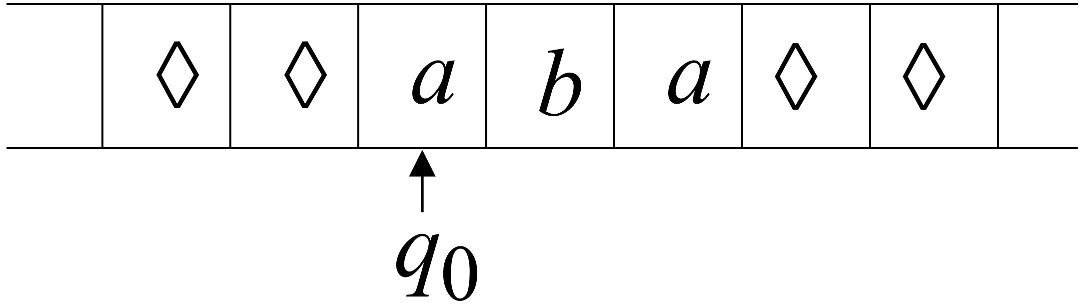


$b \rightarrow b, L$
$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

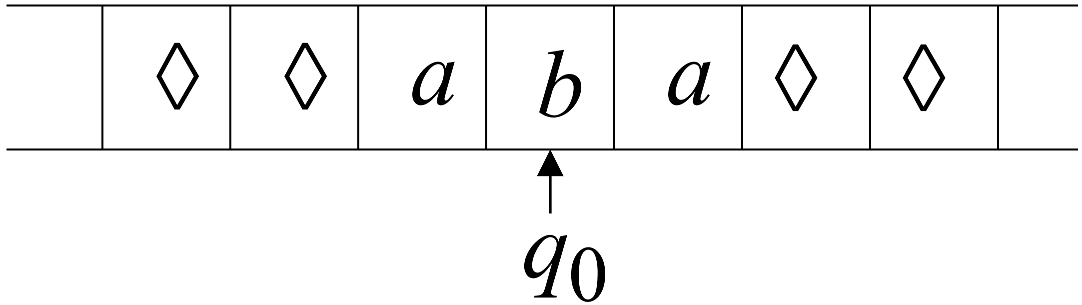
Time 2


◊ ◊ a b a ◊ ◊
q_0
$b \rightarrow b, L$
$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

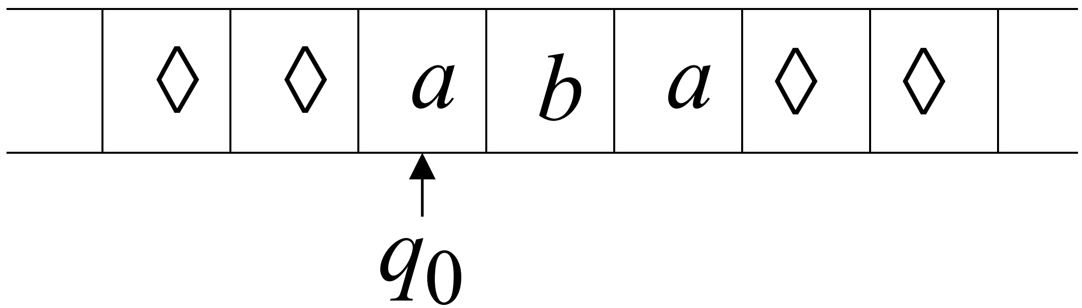
Time 2



Time 3

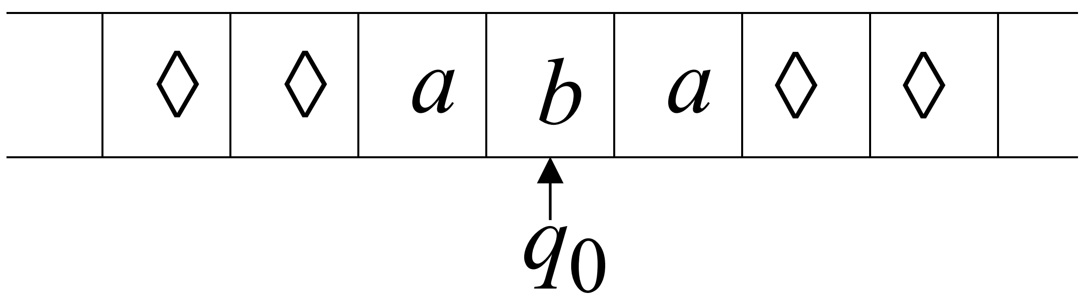


Time 4



Infinite loop

Time 5



A causa dell' infinite loop:

- lo stato accettazione non può
- essere raggiunto

- La macchina non si ferma

- La stringa di input è "rejected-rigettata"

Basic Idea:

$\{a^n b^n\}$

Match a con le b:

Repeat:

La a piu a sinistra cambiala in x

trova la b piu a sinistra cambiala con y

Until non vi sono più a o b

Se rimane una a o una b reject

--- xxayyb

q_0 a ->X q_1

q_1 devo scavalcare tutte le a tutte le Y u b

b->Y vai stato q_2

q_2 vai a L scavalcando tutte le Y e tutte le a fino a raggiungere una X

Spostare R q_0

aaaabbbb

Xaaabbbb

XaaaYbbb

XXaaYbbb

XXaaYYbb

XXXaYYbb

XXXaYYYb

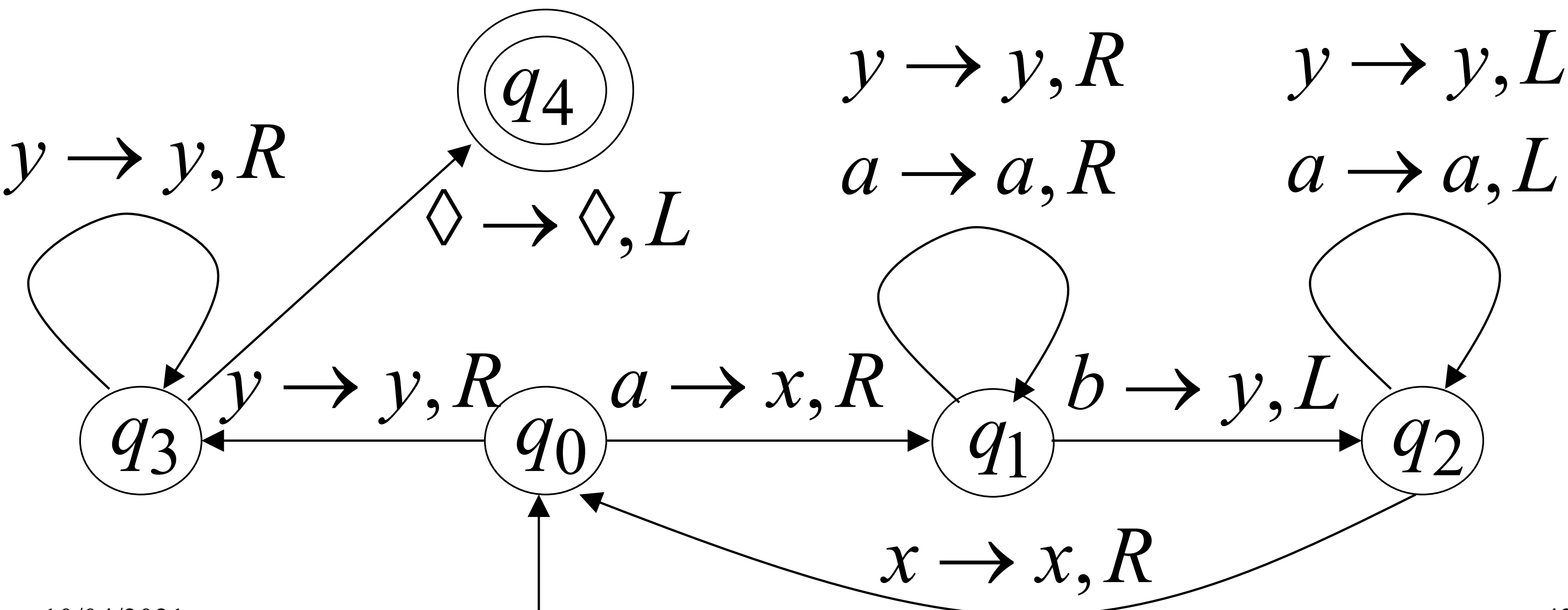
XXXXYYYb

XXXXYYYYb

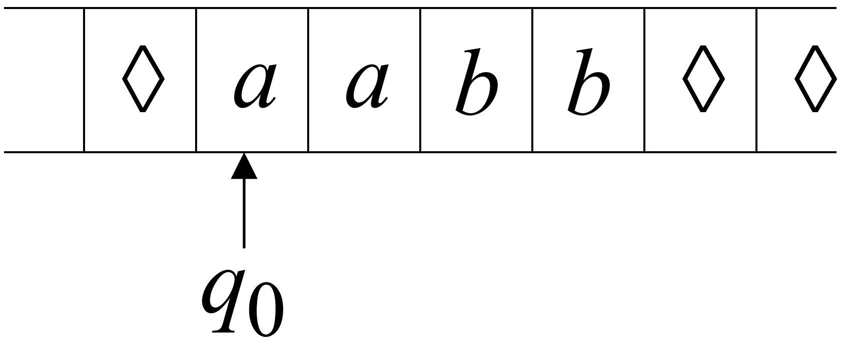
Turing macchina esempio

Machina di Turing per il linguaggio $\{a^n b^n\}$

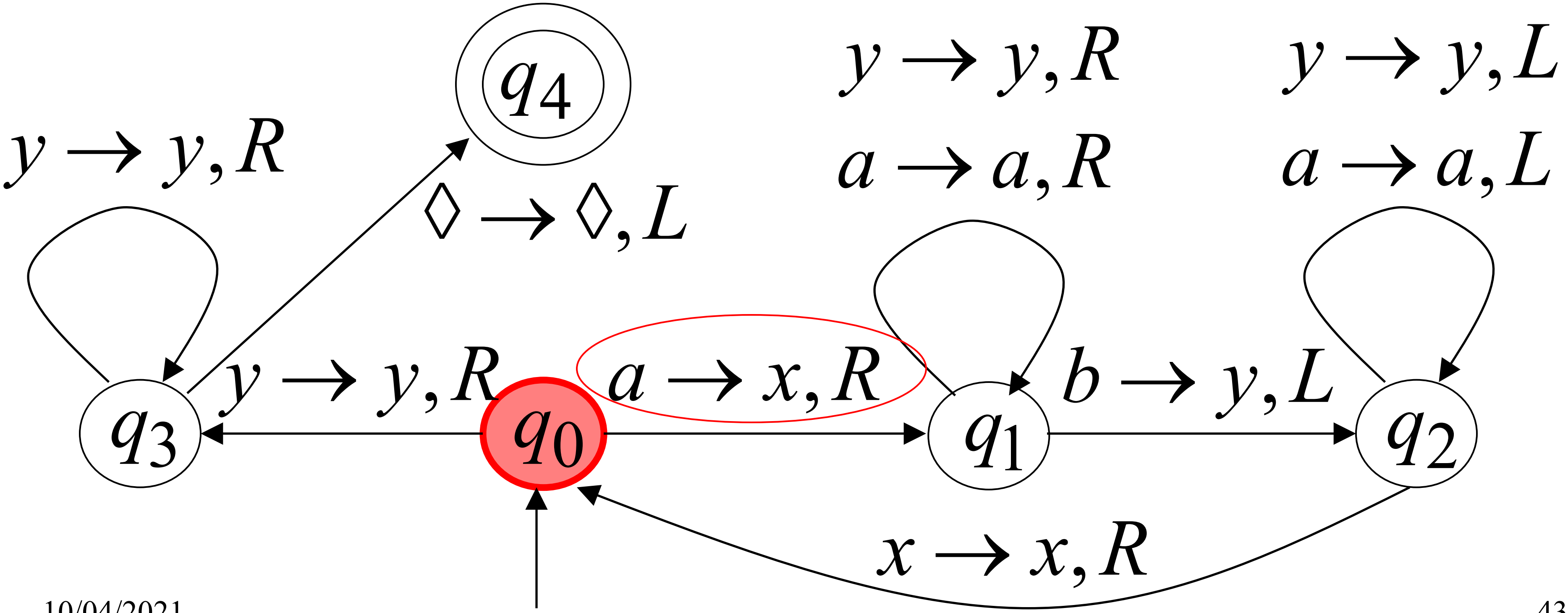
$n \geq 1$



Time 0

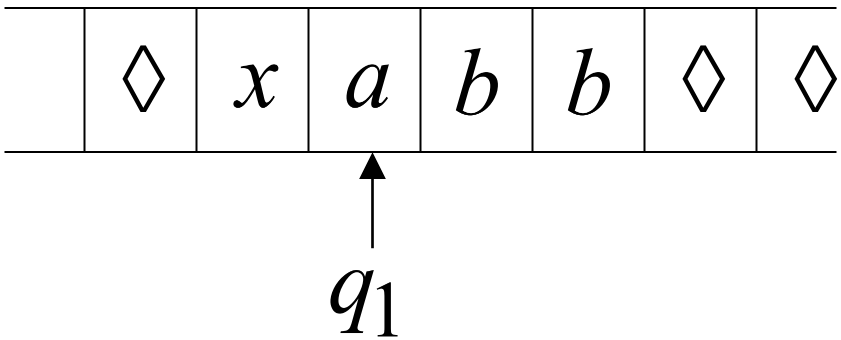


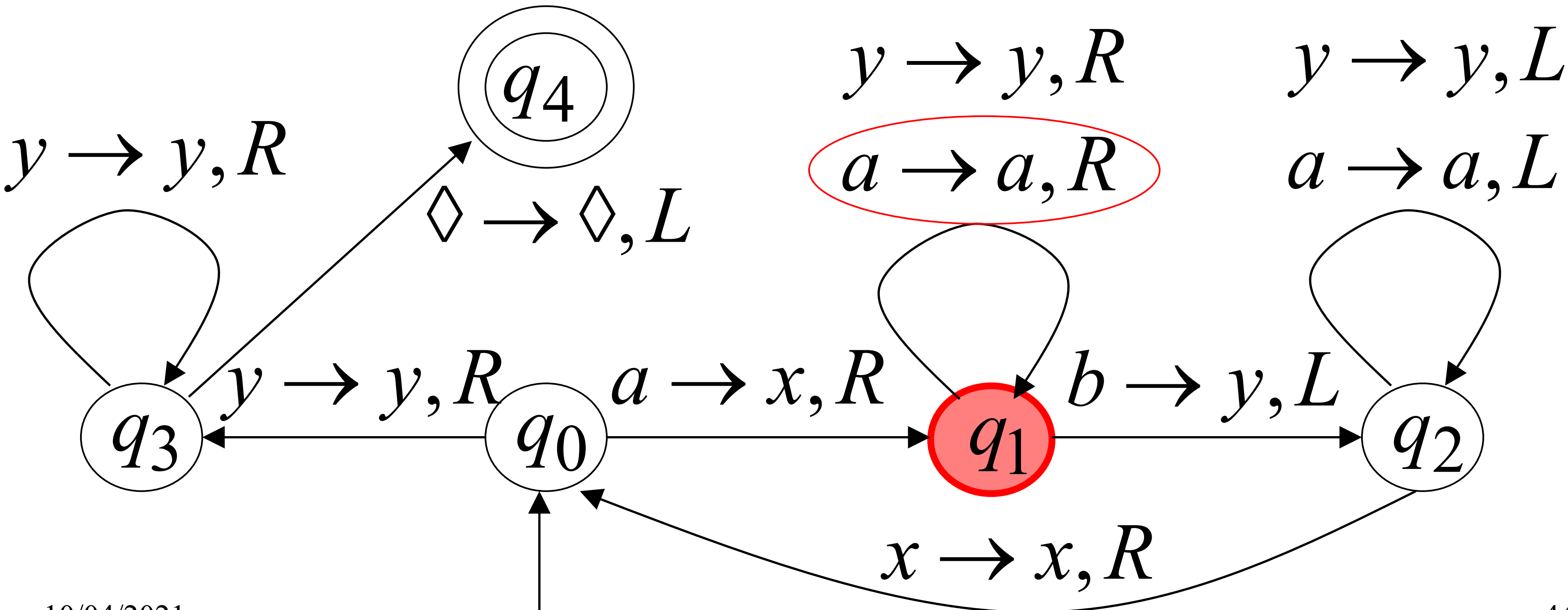
◊
a
a
b
b
◊
◊
q_0



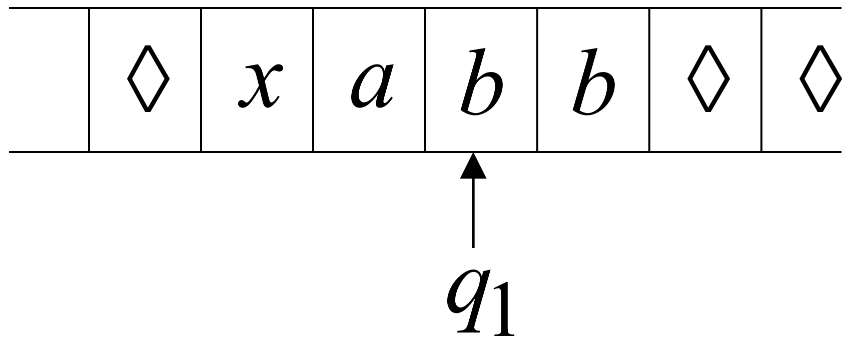
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

Time 1

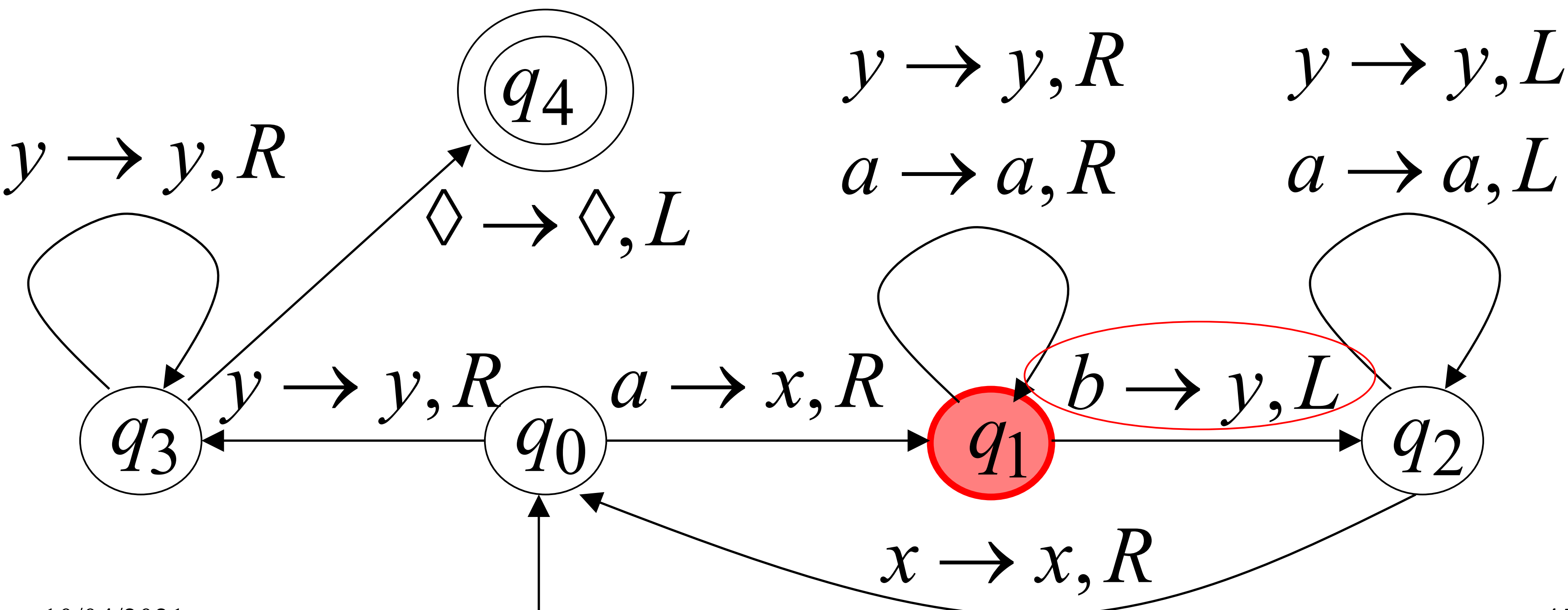

◊
x
a
b
b
◊
◊
q_1


q_4
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$

Time 2

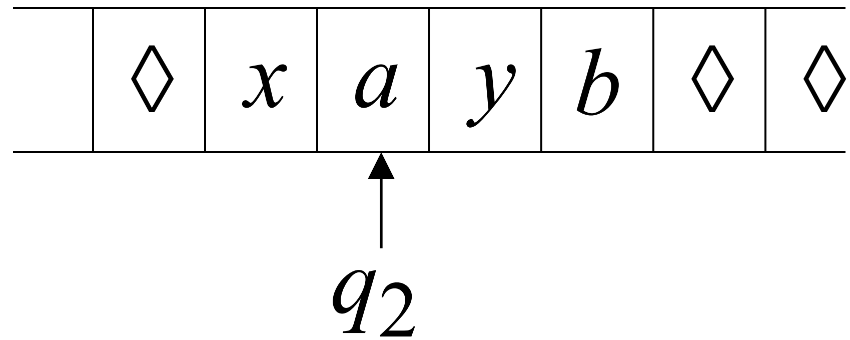


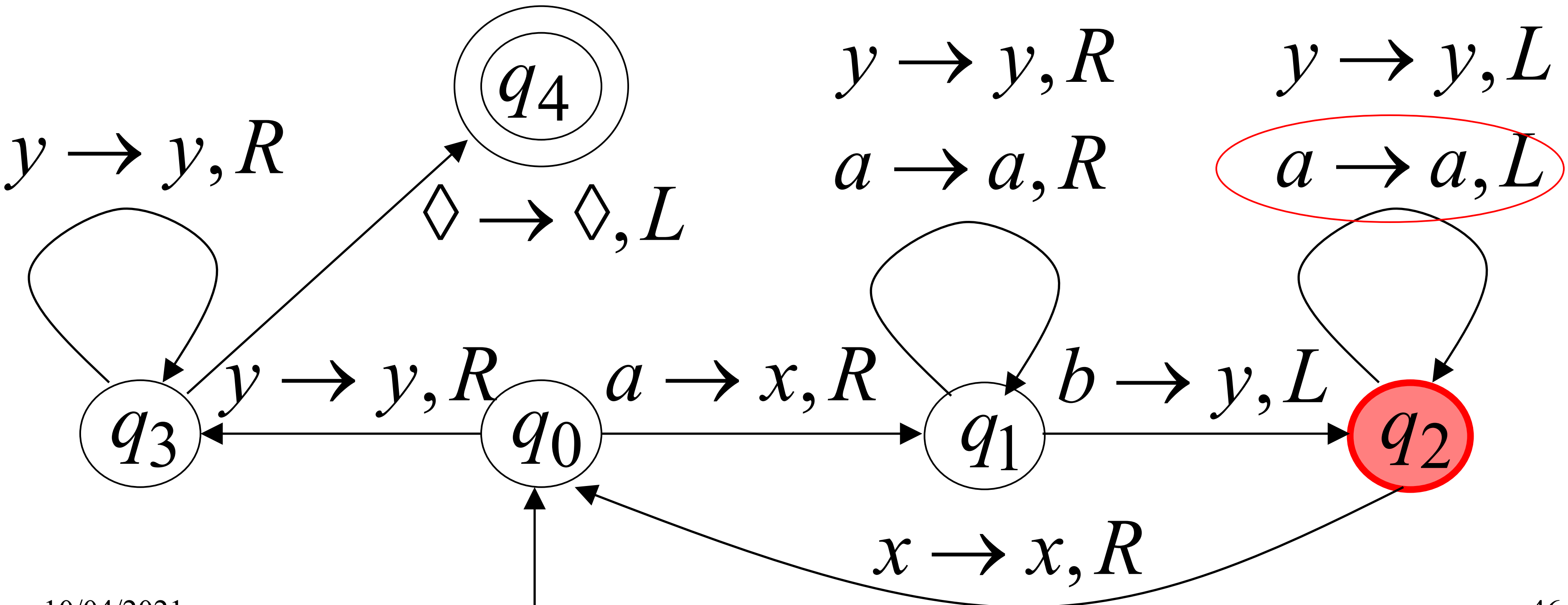
◊ x a b b ◊ ◊
q_1



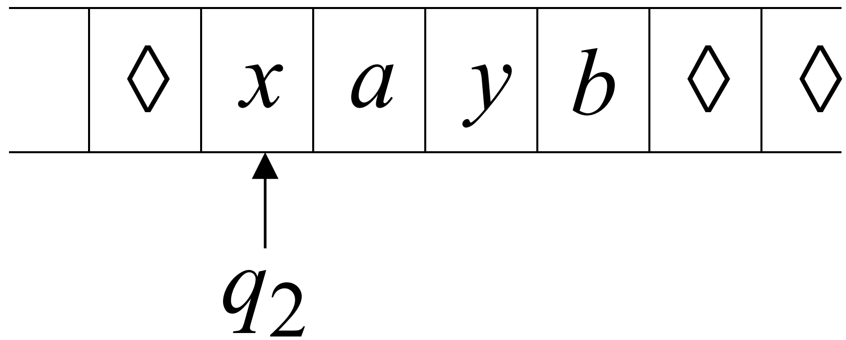
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

Time 3

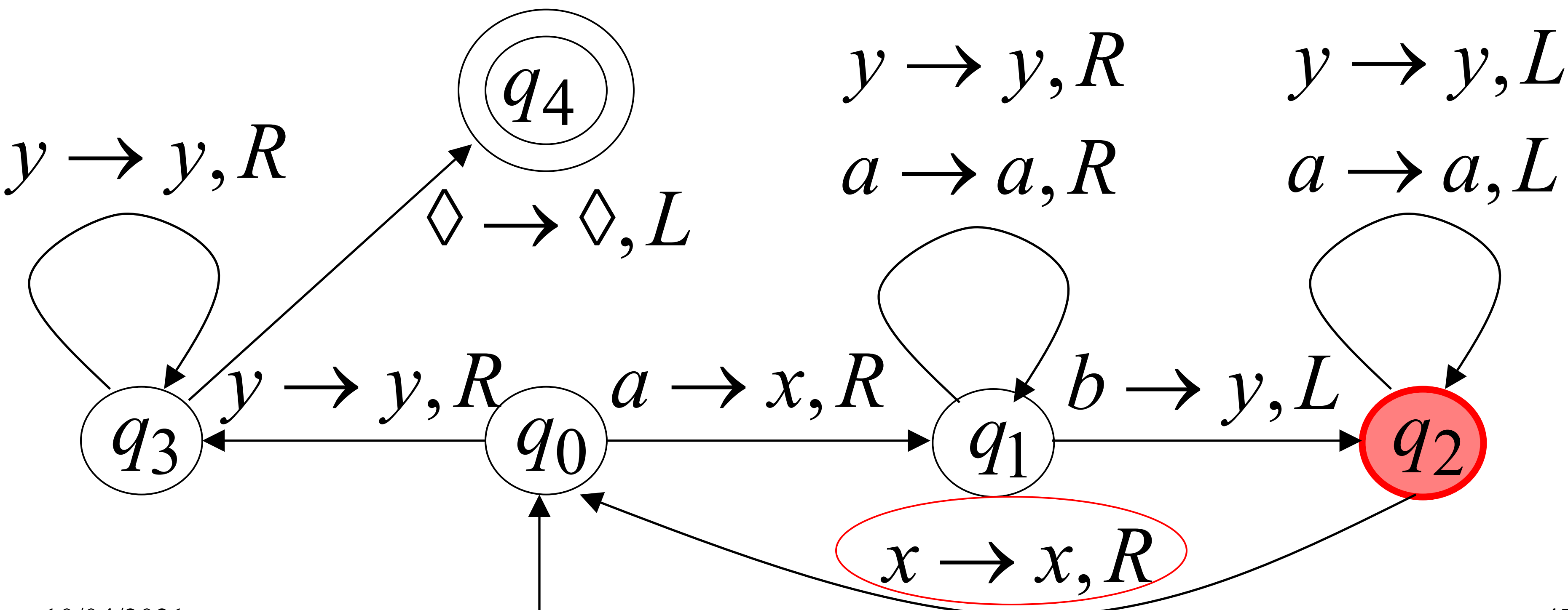

◊ x a y b ◊ ◊
q_2


q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$

Time 4

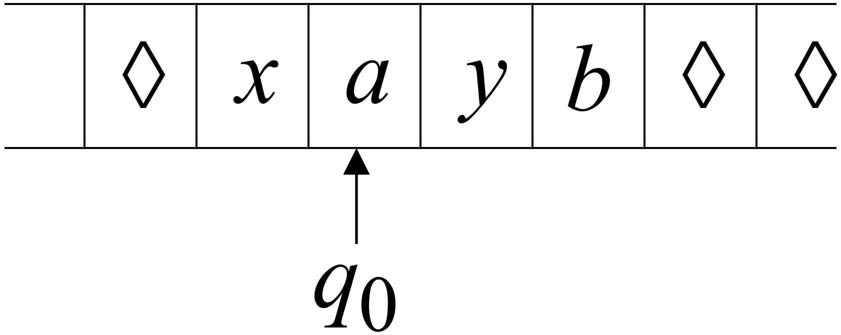


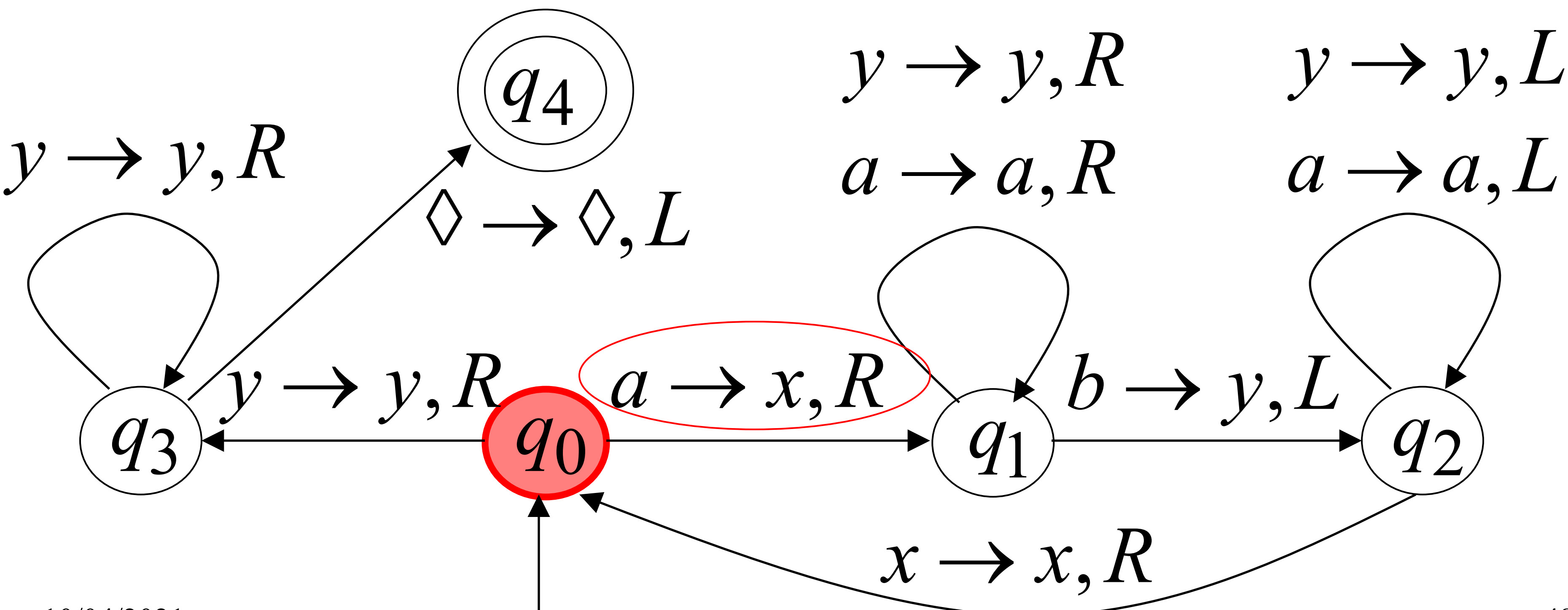
◊
x
a
y
b
◊
◊
q_2



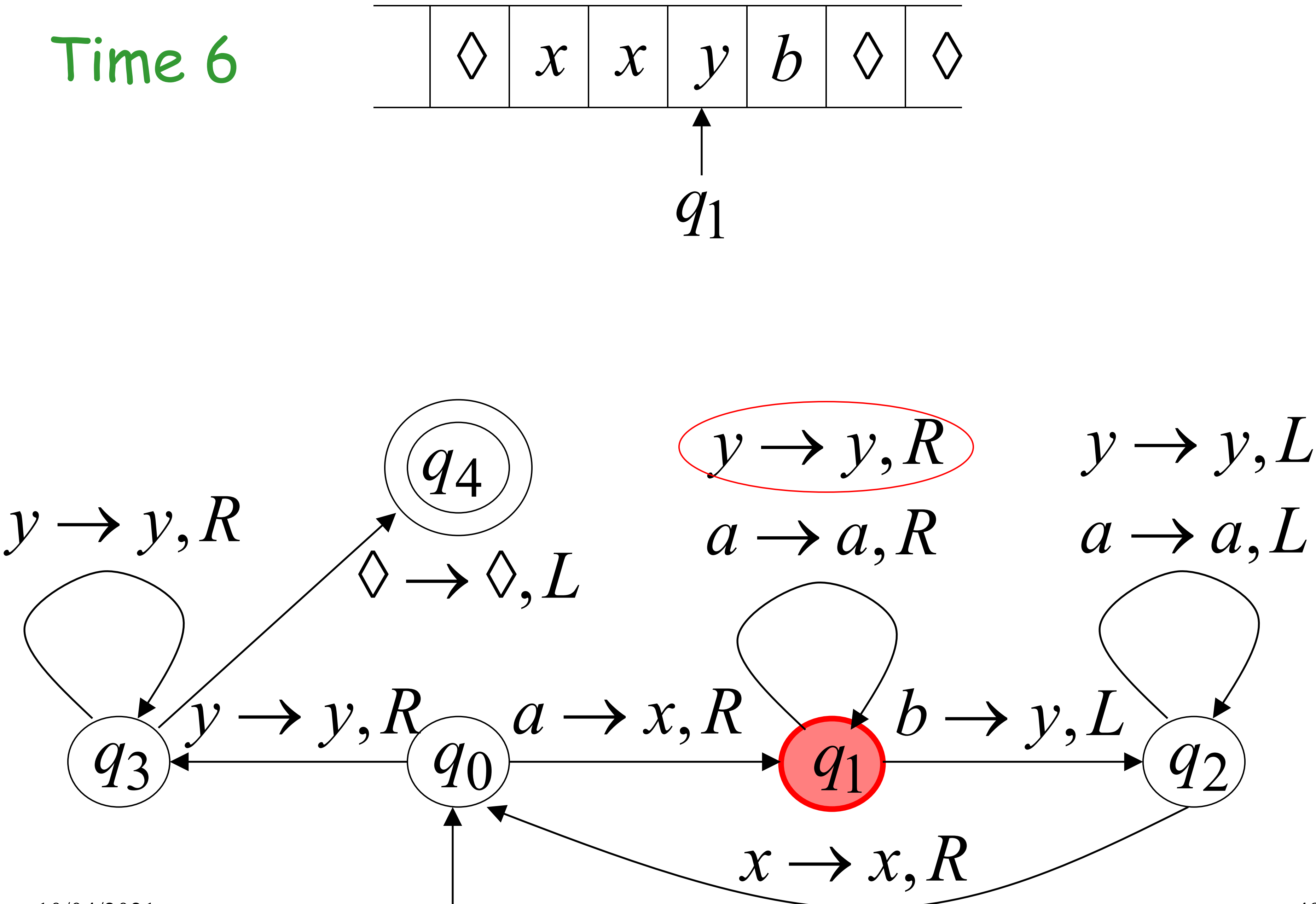
q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$

Time 5

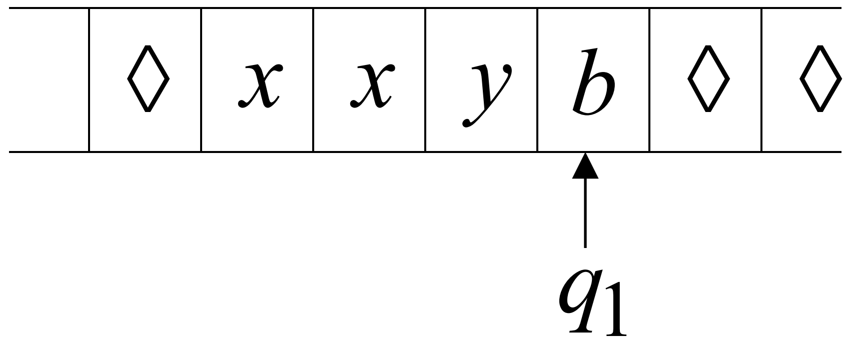

◊
x
a
y
b
◊
◊
q_0

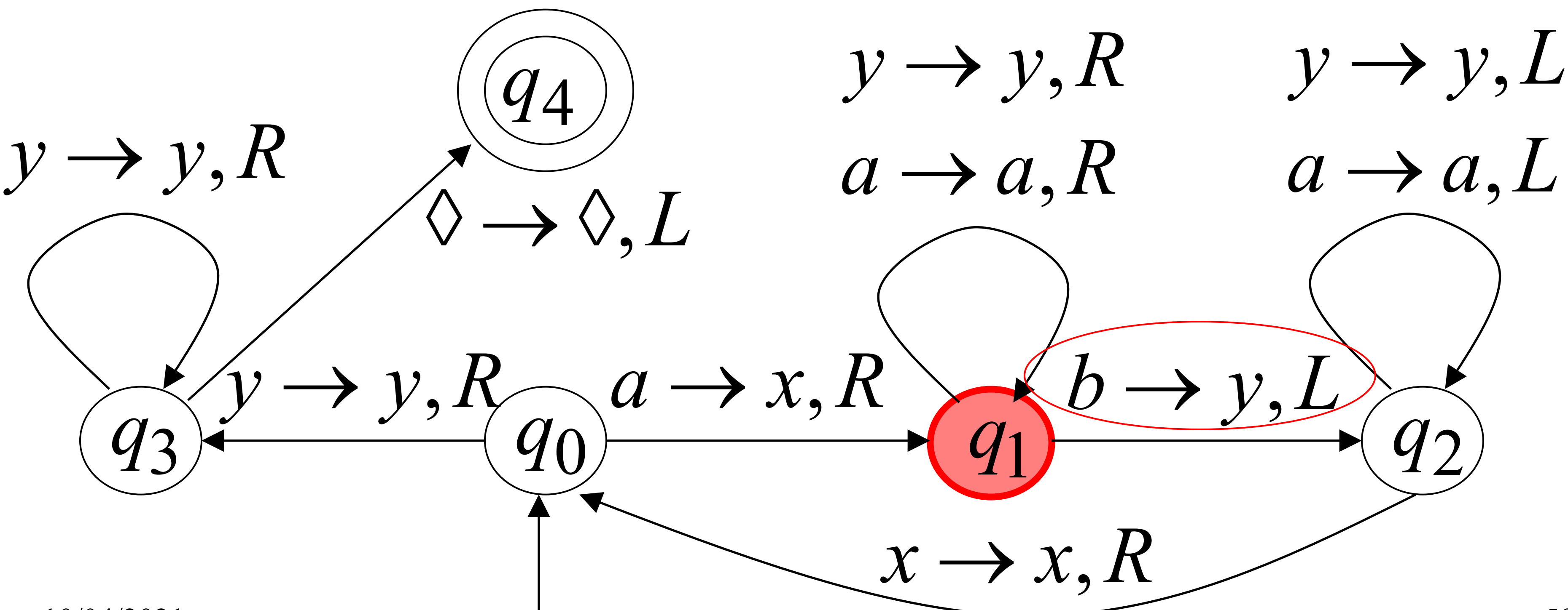

q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$

Time 6

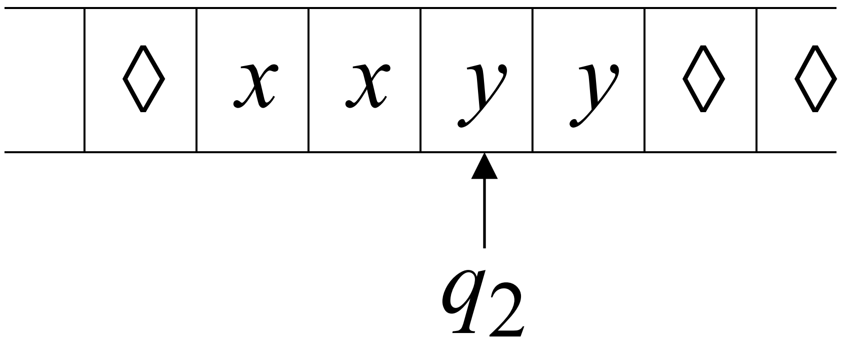


Time 7

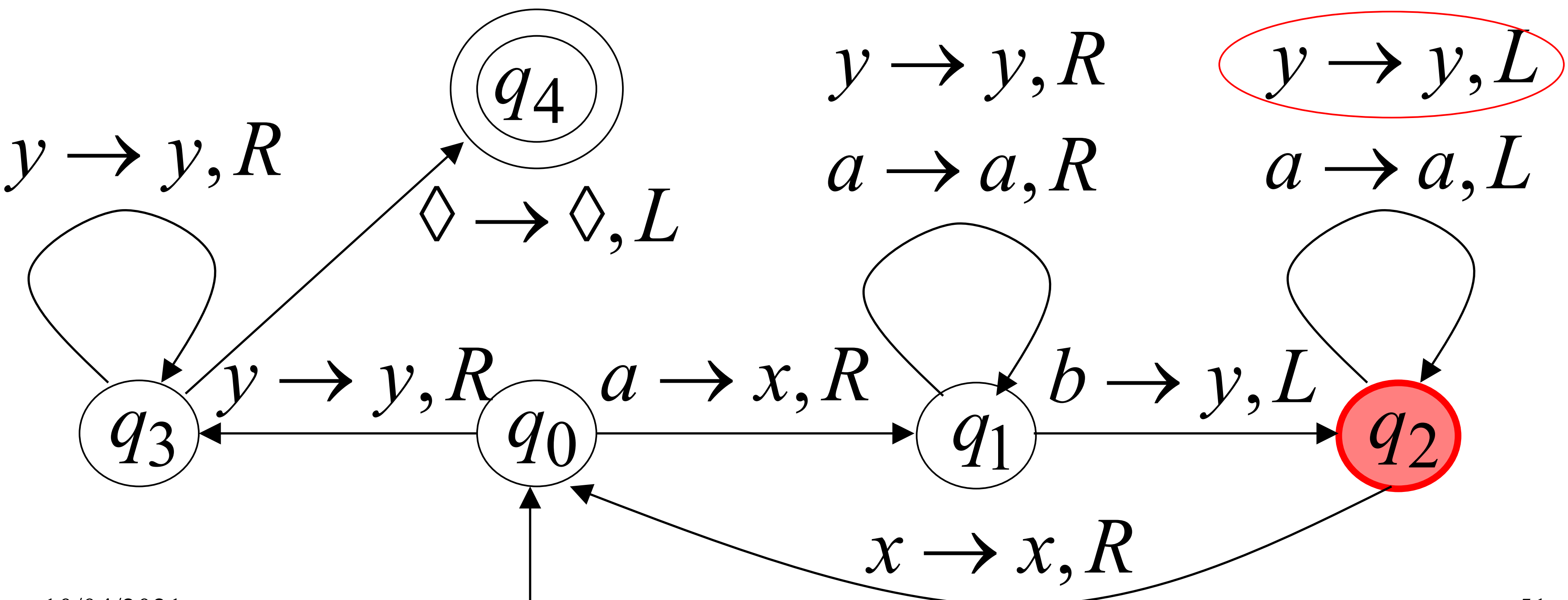




Time 8

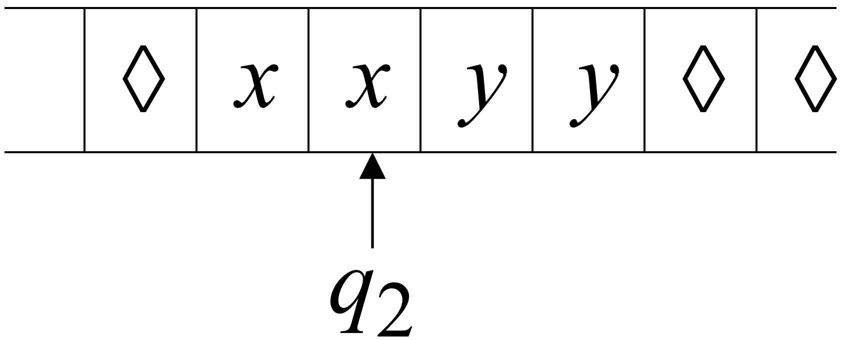


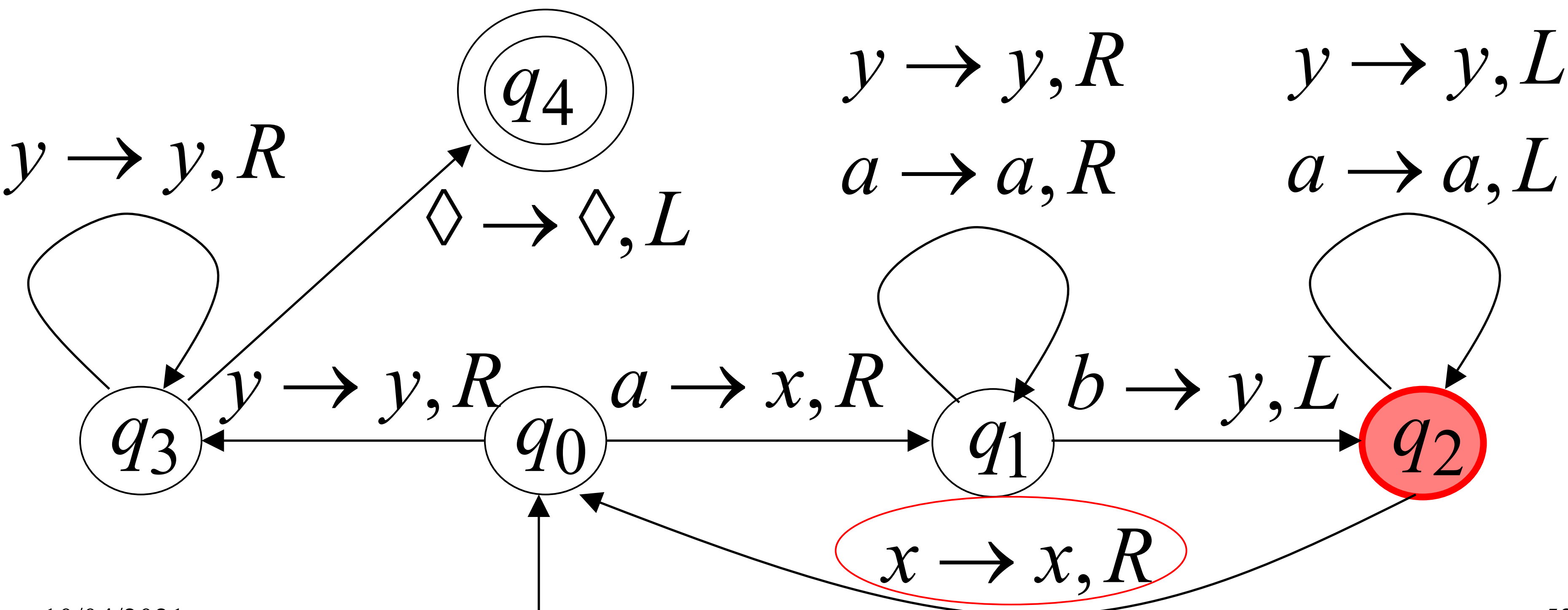
$\Diamond$
x
x
y
y
$\Diamond$
$\Diamond$
q_2



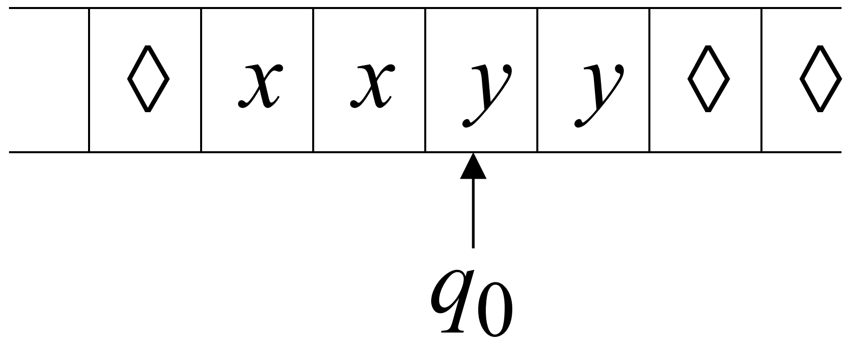
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$

Time 9

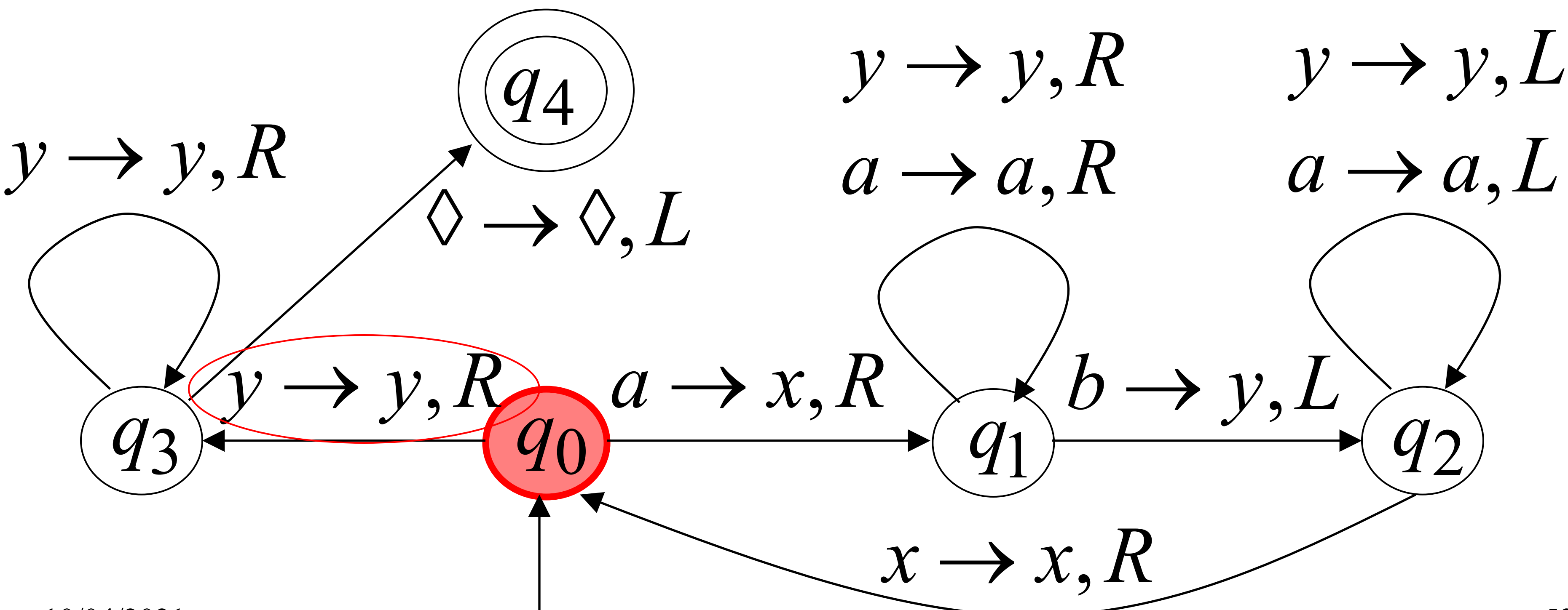




Time 10

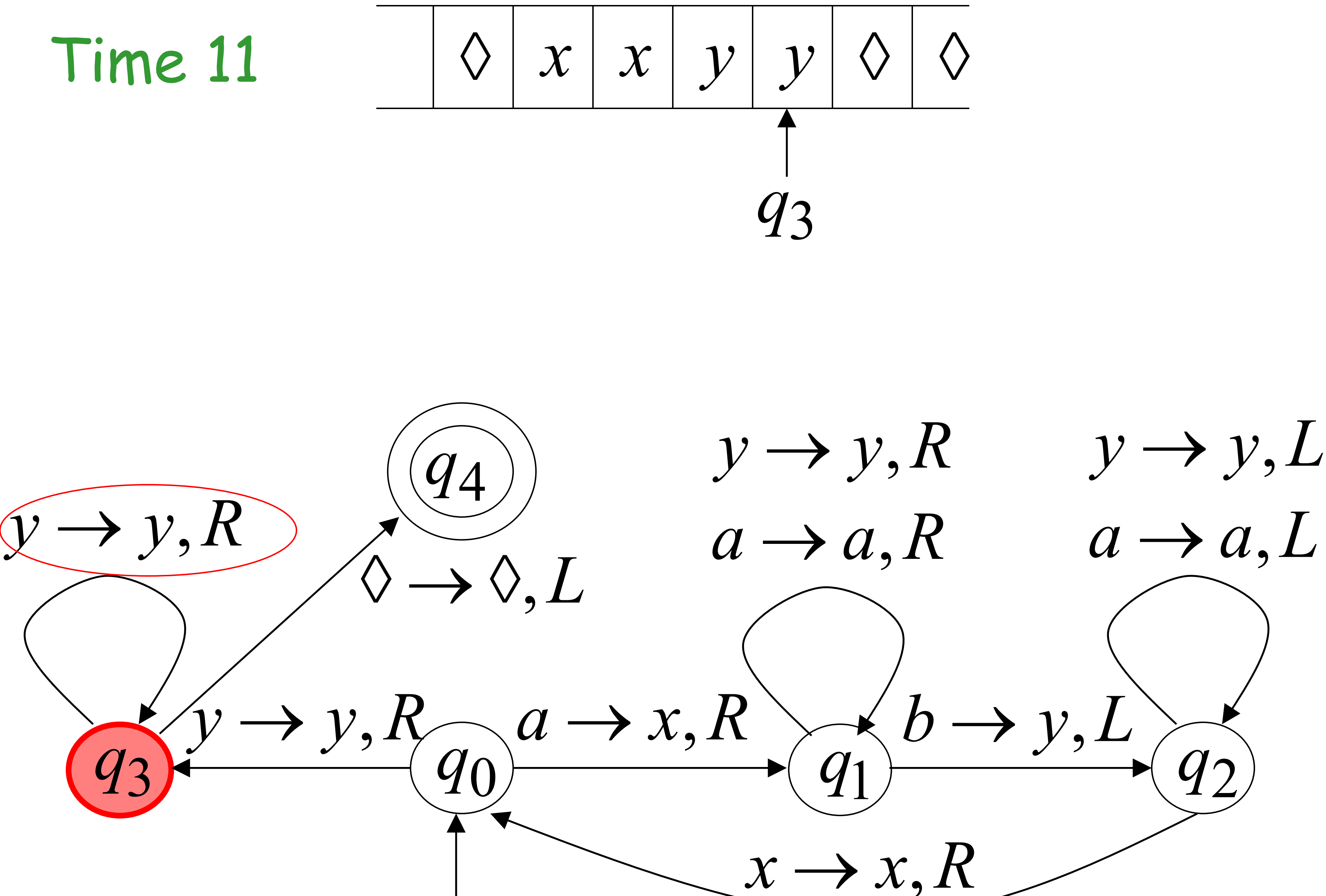


◊
x
x
y
y
◊
◊
q_0

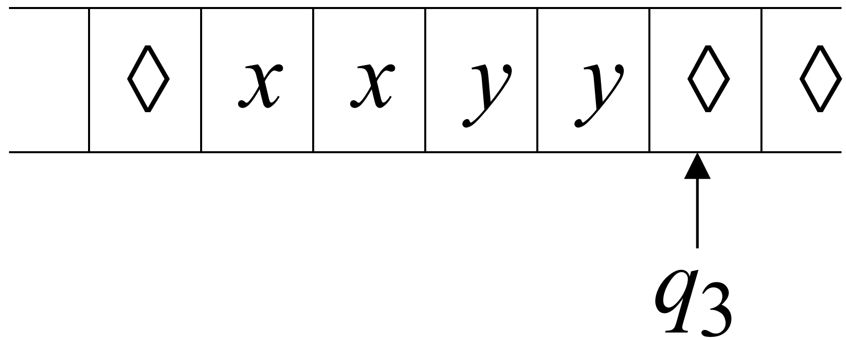


q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$

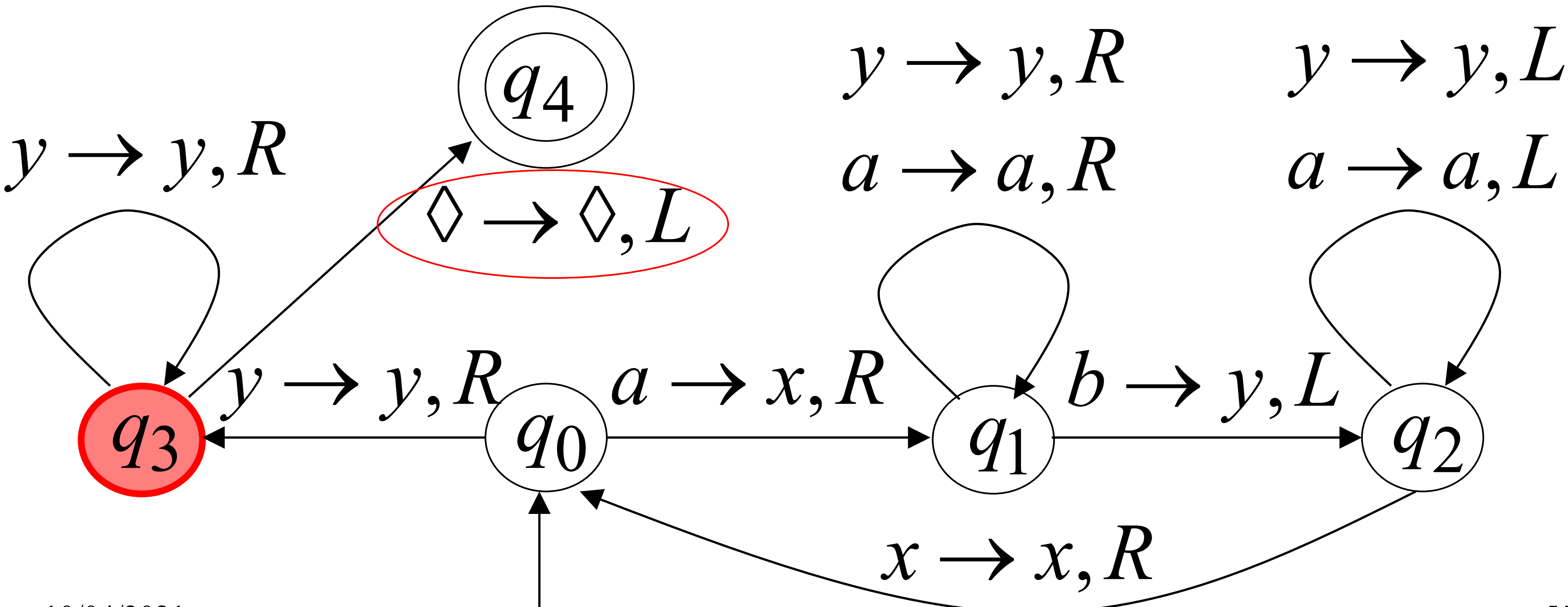
Time 11


◊
x
x
y
y
◊
◊
q_3
$y \rightarrow y, R$
q_4
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$

Time 12

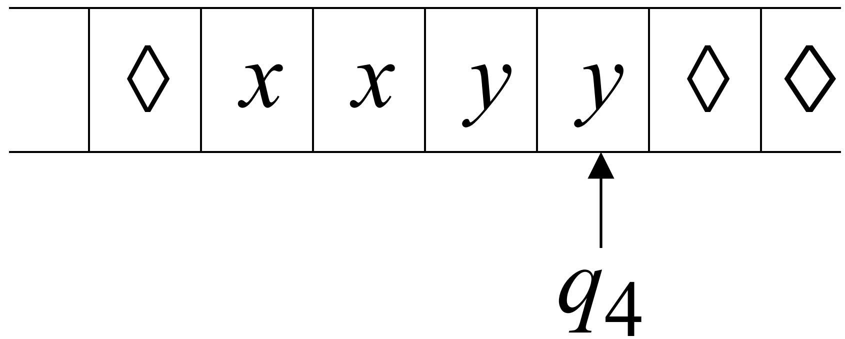


◊
x
x
y
y
◊
◊
q_3

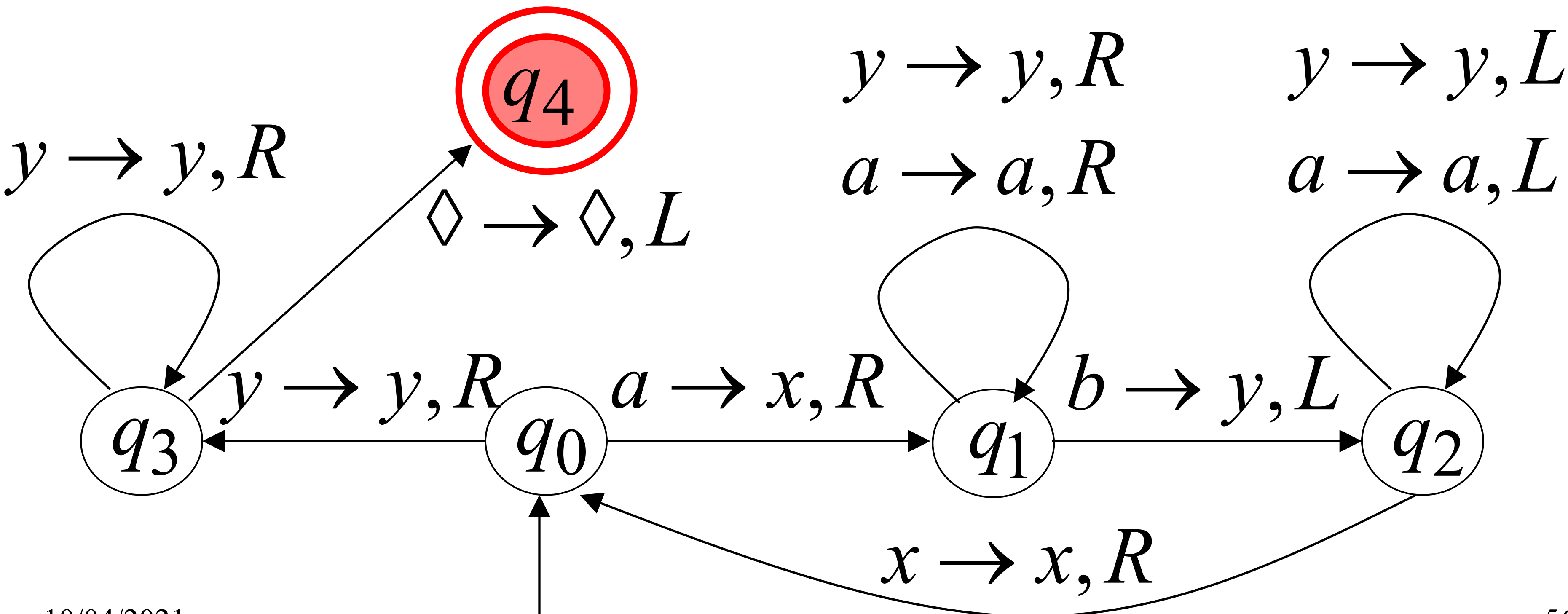


$y \to y, R$
q_4
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_0
q_1
q_2
$x \to x, R$

Time 13



Halt & accettazione



Osservazione:

Se modifichiamo
La macchina per il linguaggio $\{a^n b^n\}$

Facilmente possiamo costruire
Una macchina per il linguaggio $\{a^n b^n c^n\}$

Definizione formale
di
macchina di turing

Funzione Transizione

$$q_1 \xrightarrow{a \rightarrow b, R} q_2$$

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

Funzione Transizione

$$q_1 \xrightarrow{c \rightarrow d, L} q_2$$

$$\delta(q_1, c) = (q_2, d, L)$$

Turing macchina:

stati

Input alfabeto

Tape alfabeto

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

Transition function

iniziale stato

blank

accettazi stati

Tipo della delta

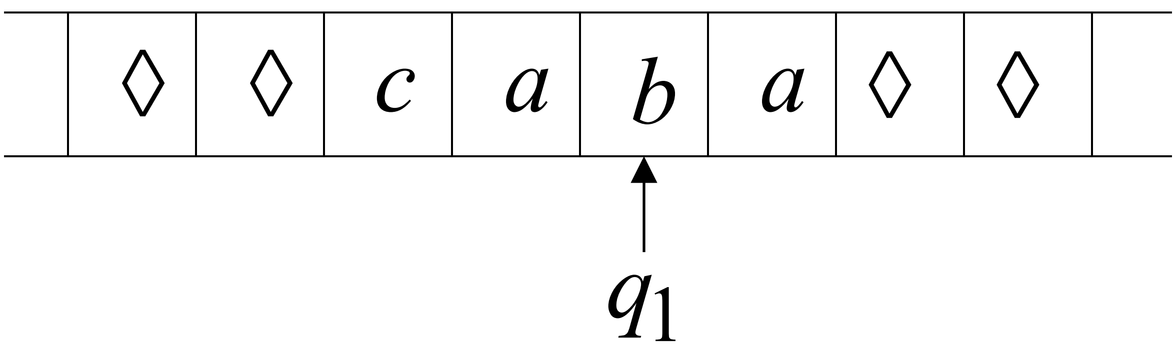
Input

Stati xAI U AN->

Stati x AI U AN x Op

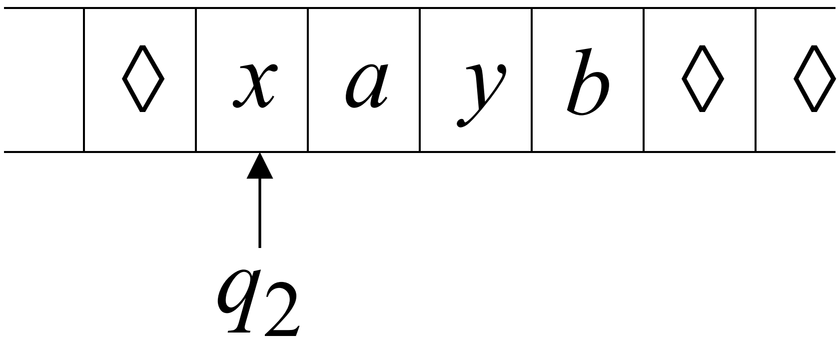
Op =L. R

Configurazione

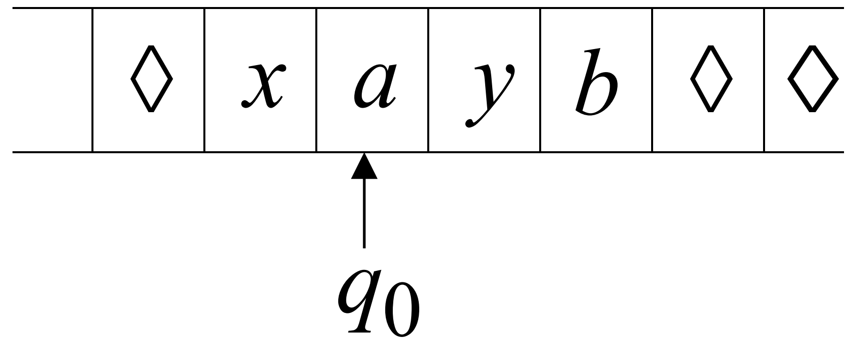


descrizione istantanea: $ca\ q_1\ ba$

Time 4



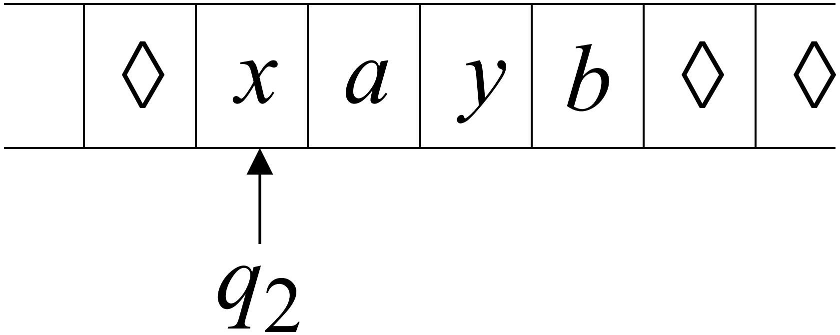
Time 5



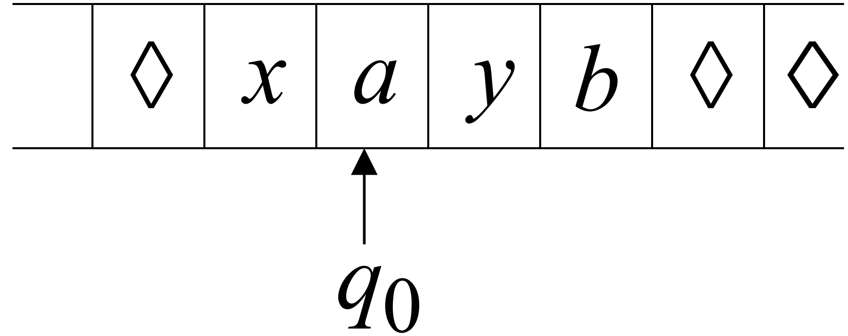
Una mossa $$q_2\ xayb \succ x\ q_0\ ayb$$

(dà)

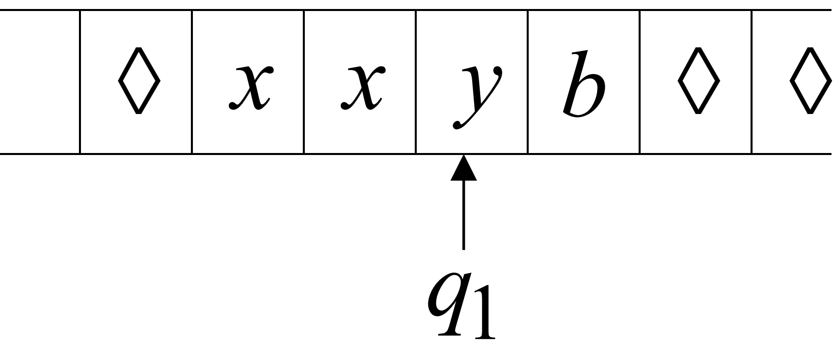
Time 4



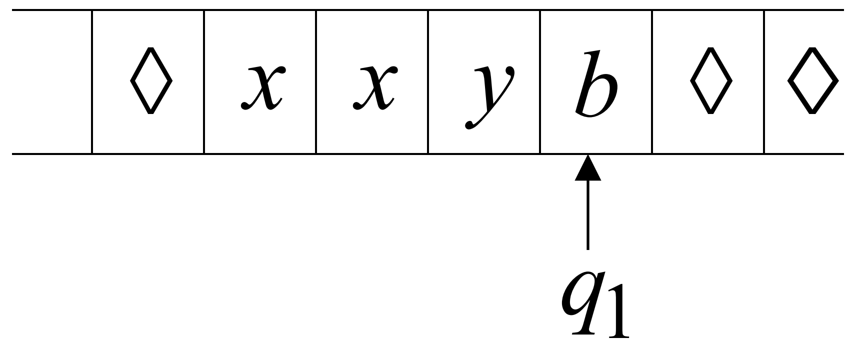
Time 5



Time 6



Time 7



Un calcolo

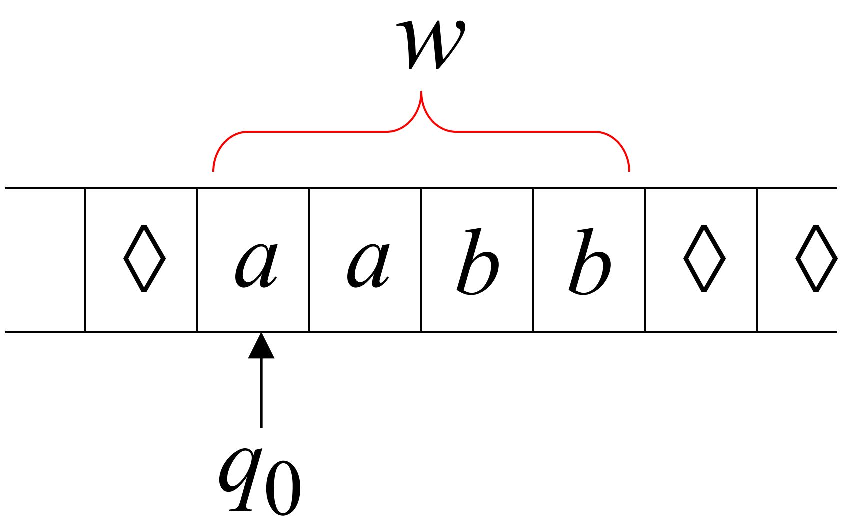
$$q_2\ xayb \succ x\ q_0\ ayb \succ xx\ q_1\ yb \succ xxy\ q_1\ b$$

$$q_2\ xayb \succ x\ q_0\ ayb \succ xx\ q_1\ yb \succ xxy\ q_1\ b$$

Notazione equivalente: $q_2\ xayb \overset{*}{\succ} xxy\ q_1\ b$

configurazione Iniziale : $q_0\ w$

stringa di input



il linguaggio accettato

Per ogni macchina Turing M

$$L(M) = \{w : \quad q_0\, w \overset{*}{\succ} x_1\, q_f\, x_2\}$$

Accettato in forma standard

q_o w da* w q_f

stato iniziale

accettazione stato

Se un linguaggio L è accettato da
Una macchina di Turing M
A noi diciamo che L è:

- Turing Riconoscibile

Alfabeto

Altri nomi usati:

- Turing accettati
- Recursivamente Enumerabili

Alfabeto L definito a partire da quell'alfabeto

L turing riconoscibile

w elemento A* M(w) raggiunge lo stato finale se w appartiene ad L

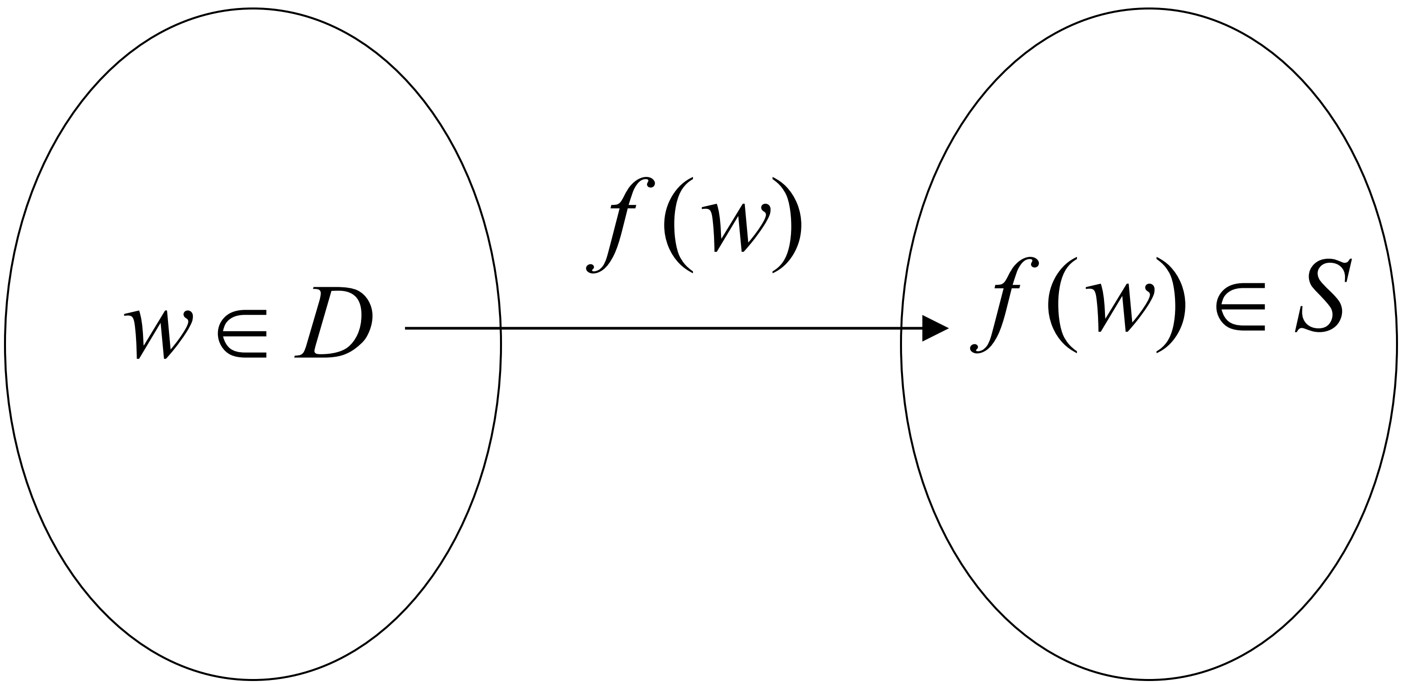
non lo raggiunge ? Altrimenti

Non raggiunge uno stato finale. Ma questo non vuol dire che la macchina si ferma

Calcolare funzioni con macchine di Turing

Una funzione $f(w)$ ha:

Dominio: D Regione dei risultati: S



Una funzione può avere molti parametri:

esempio: funzione Addizione

$$f(x, y) = x + y$$

dominio degli interi

Decimali: 5

Binari: 101

Unario: 11111 ⇨ 0->1

1->11

n-> n+1 1

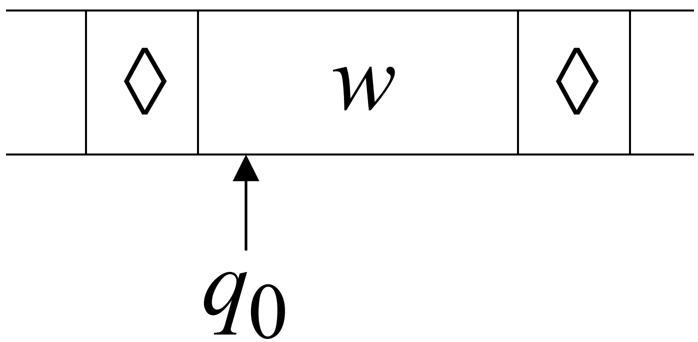
Useremo rapresentazione **unaria** :

Più facile da usare con le macchine di Turing

Definizione:

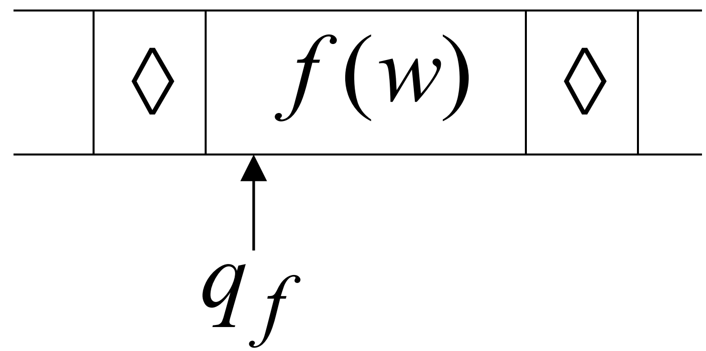
Una funzione f è calcabile se
vi è una macchina di Turing M tale che:

Configurazione iniziale



stato iniziale

Configurazione Finale



Stato di accettazione

Per tutti $w \in D$ dominio

Calcolare in modo standard

Inltre parole :

A Una funzione f è calcabile se
Vi è una macchina di Turing M tale che:

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

configurazione iniziale

configurazione Finale

Per tutte $w \in D$ dominio

esempio

la funzione $f(x, y) = x + y$ è calcolabile

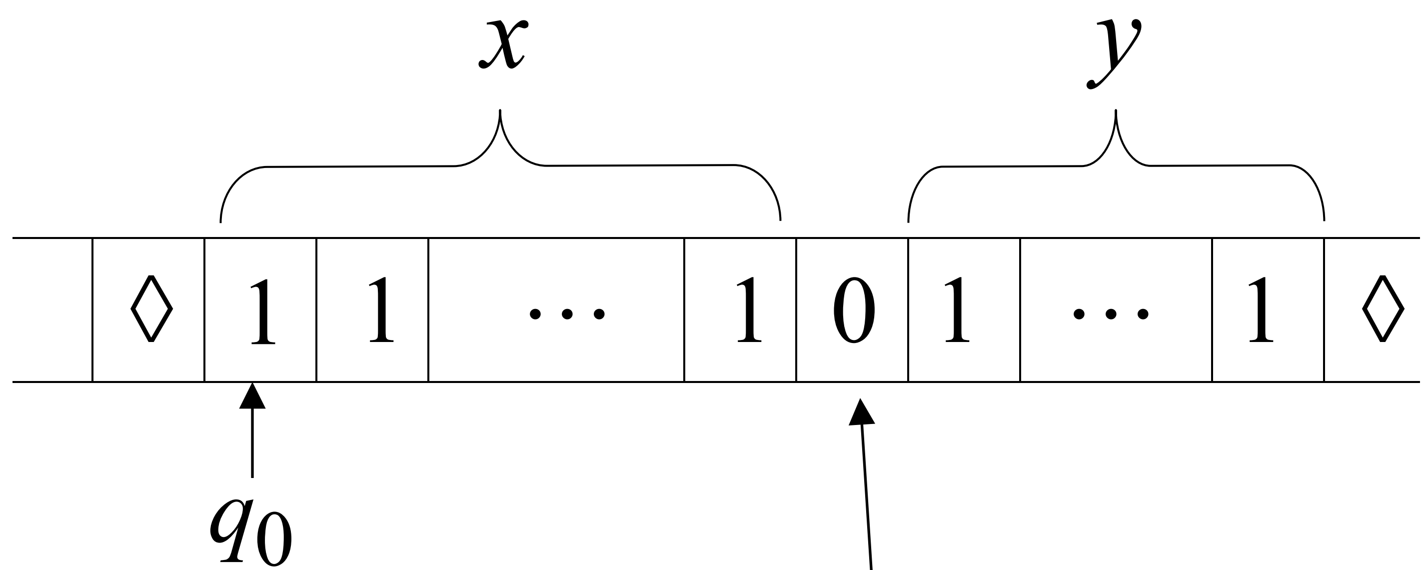
x, y Sono interi

macchina Turing :

stringa di input: $x0y$ unario

Output stringa: $xy0$ unario

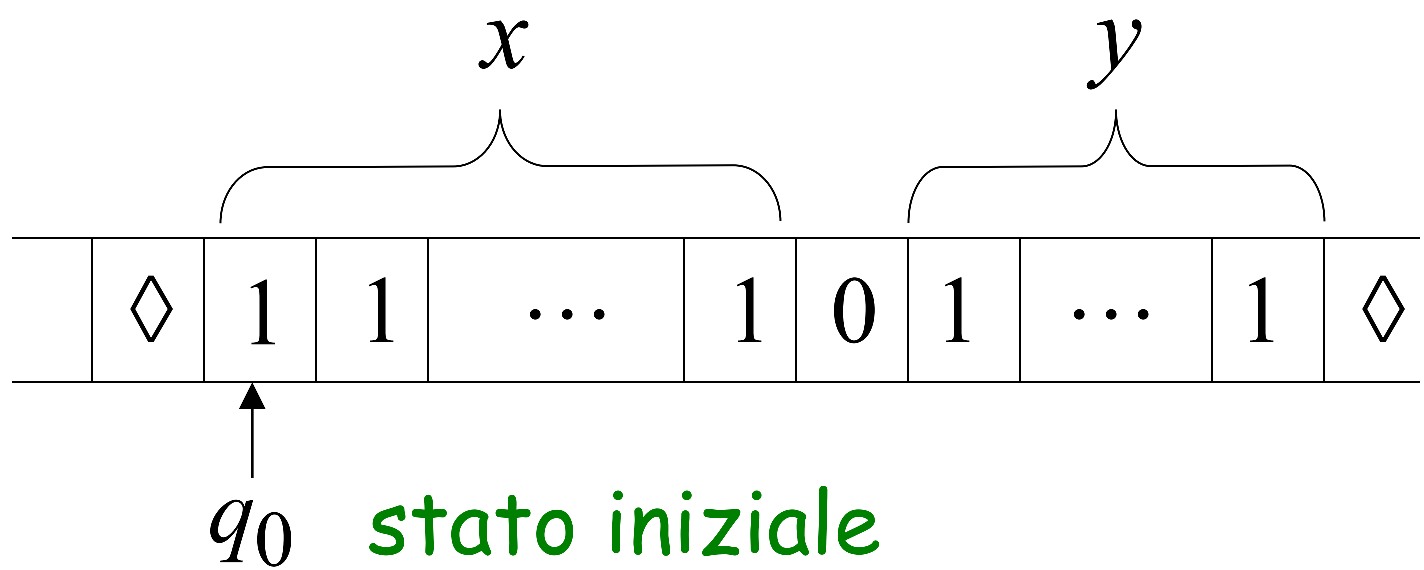
Start



iniziale stato

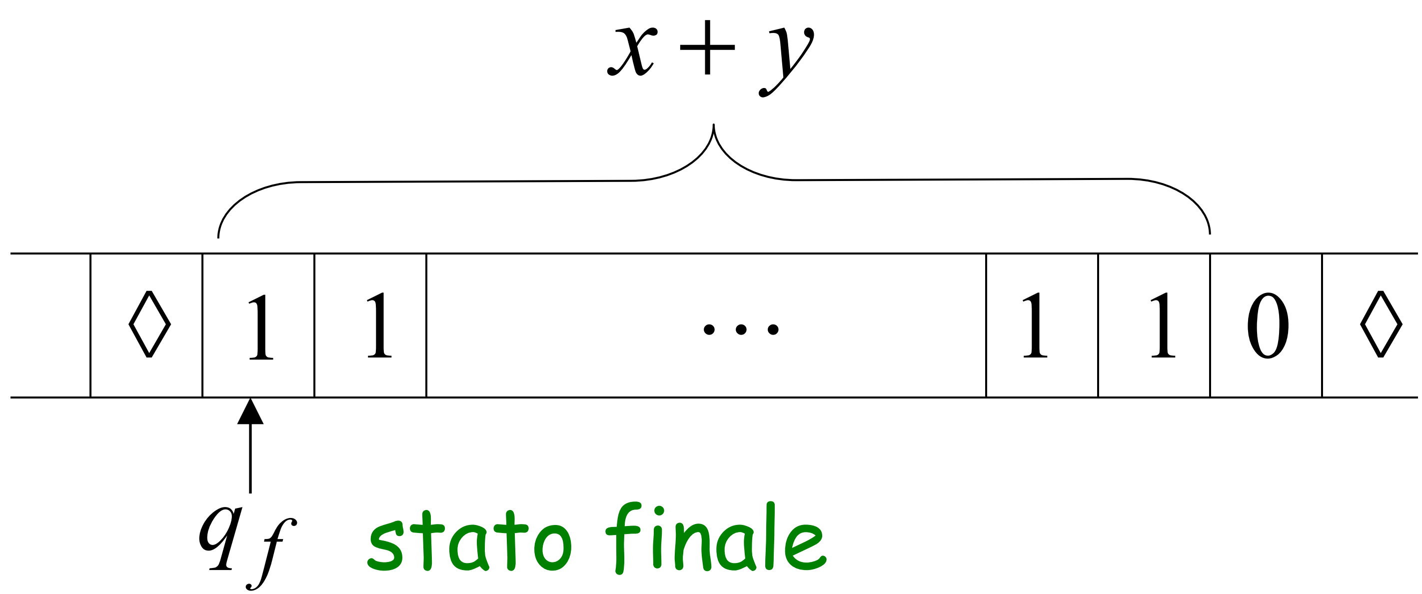
il 0 è il delimitatore che
Separa I due numeri

Start



stato iniziale

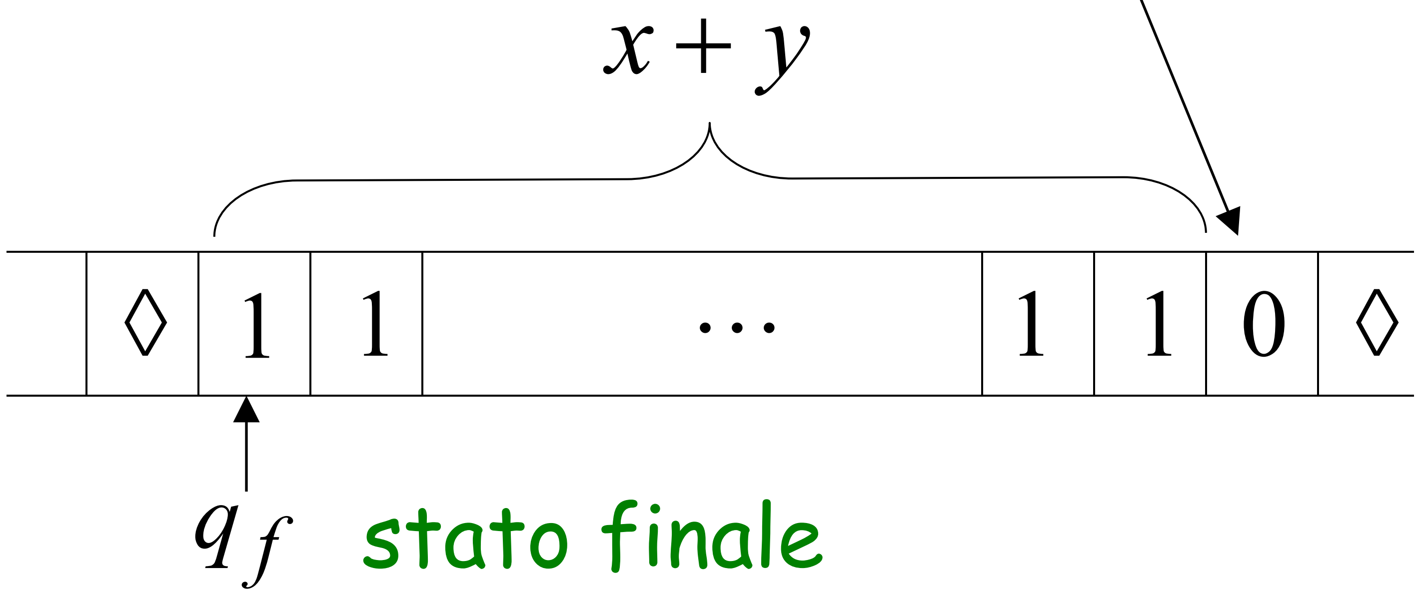
Fine



stato finale

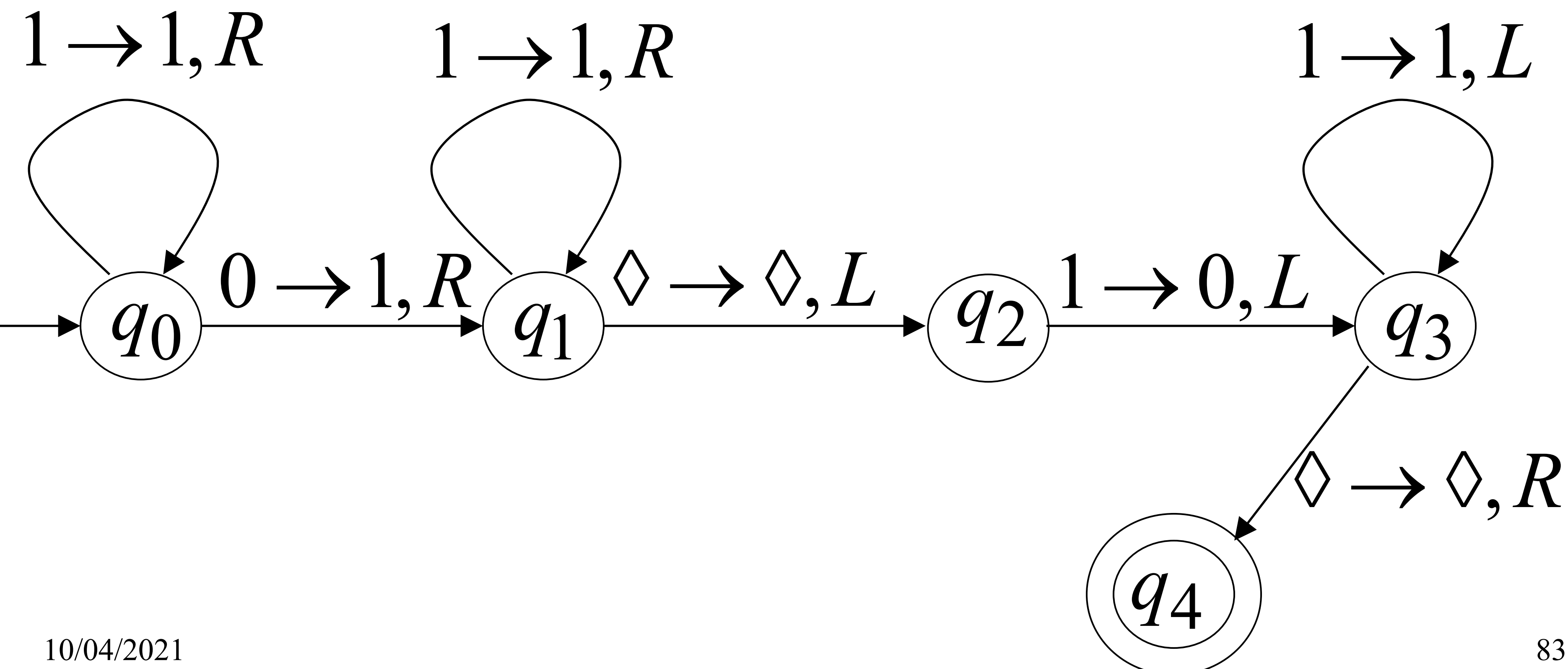
lo 0 ci può aiutare se usiamo
il risultato per un altra operazione

Fine



macchina Turing per la funzione $f(x, y) = x + y$

Ricordarsi di eliminare due 1 alla fine



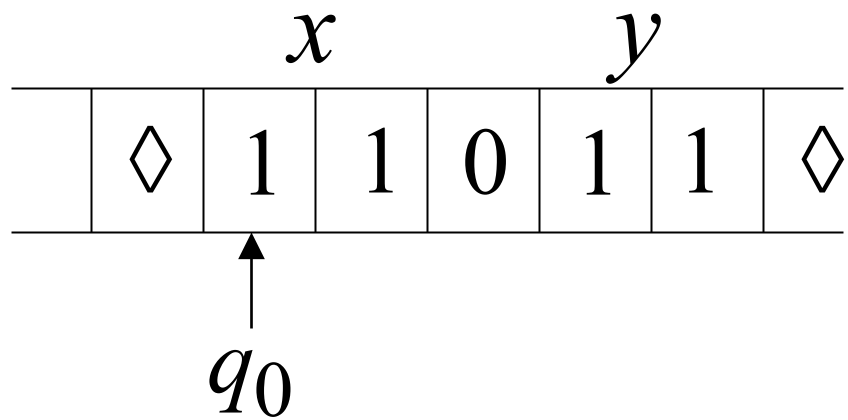
Consideriamo i numeri naturali
senza lo zero
Quindi basta avere n= 1alla n

esempio di esecuzione:

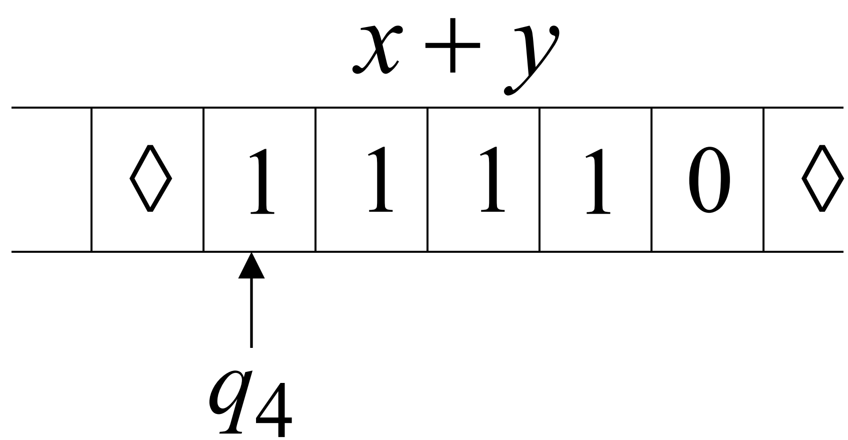
$x = 11$ (=2)

$y = 11$ (=2)

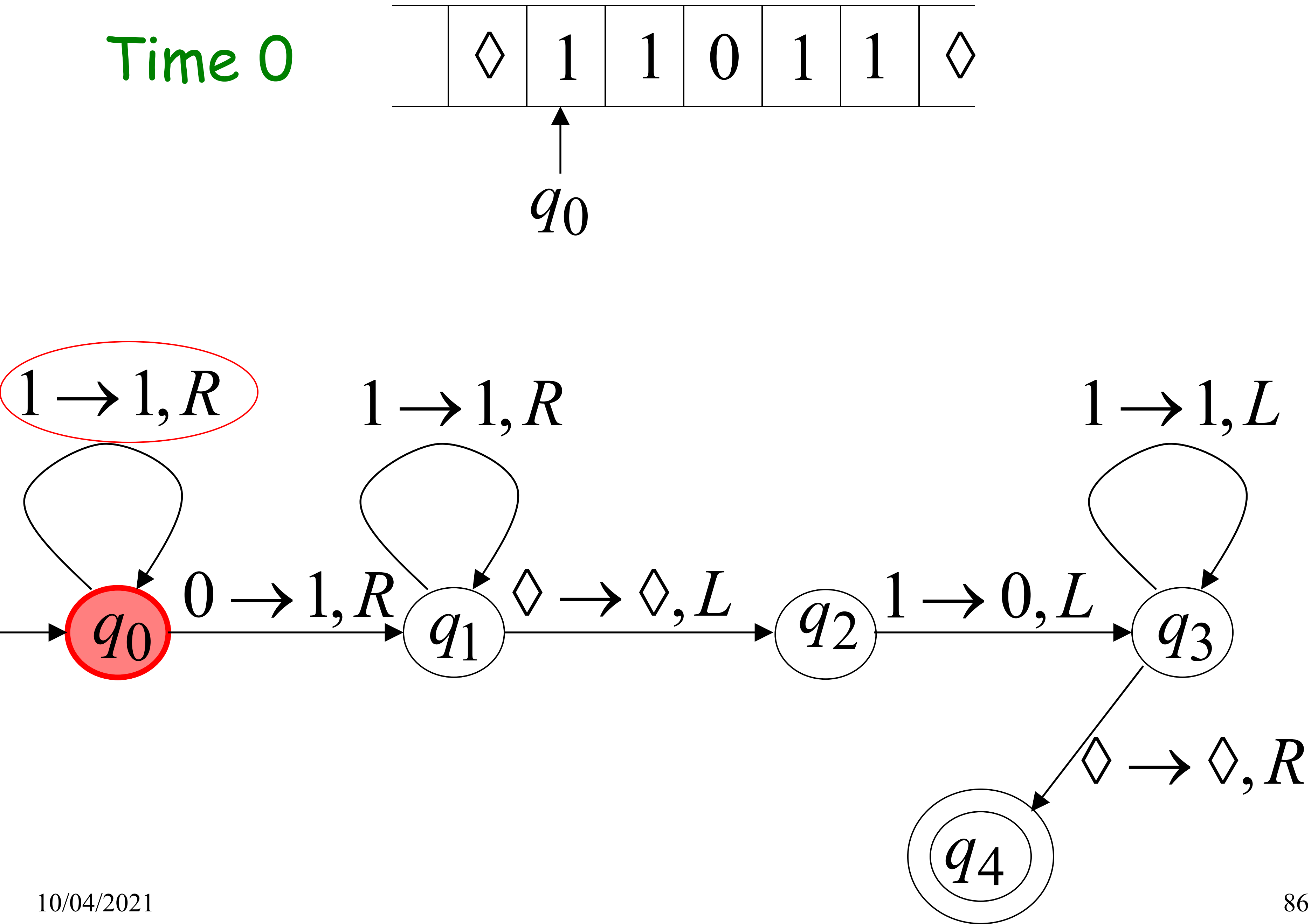
Time 0



Final Result

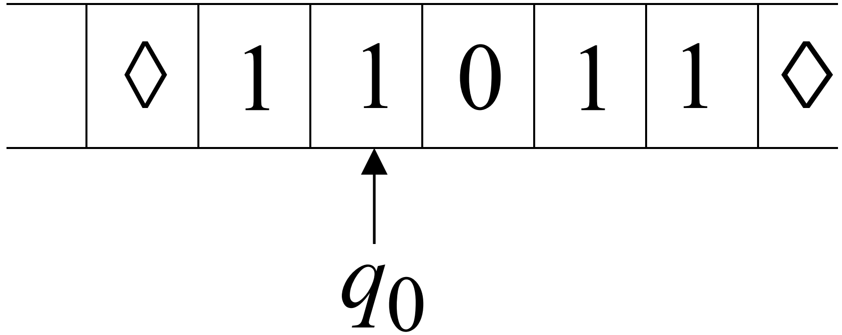


Time 0

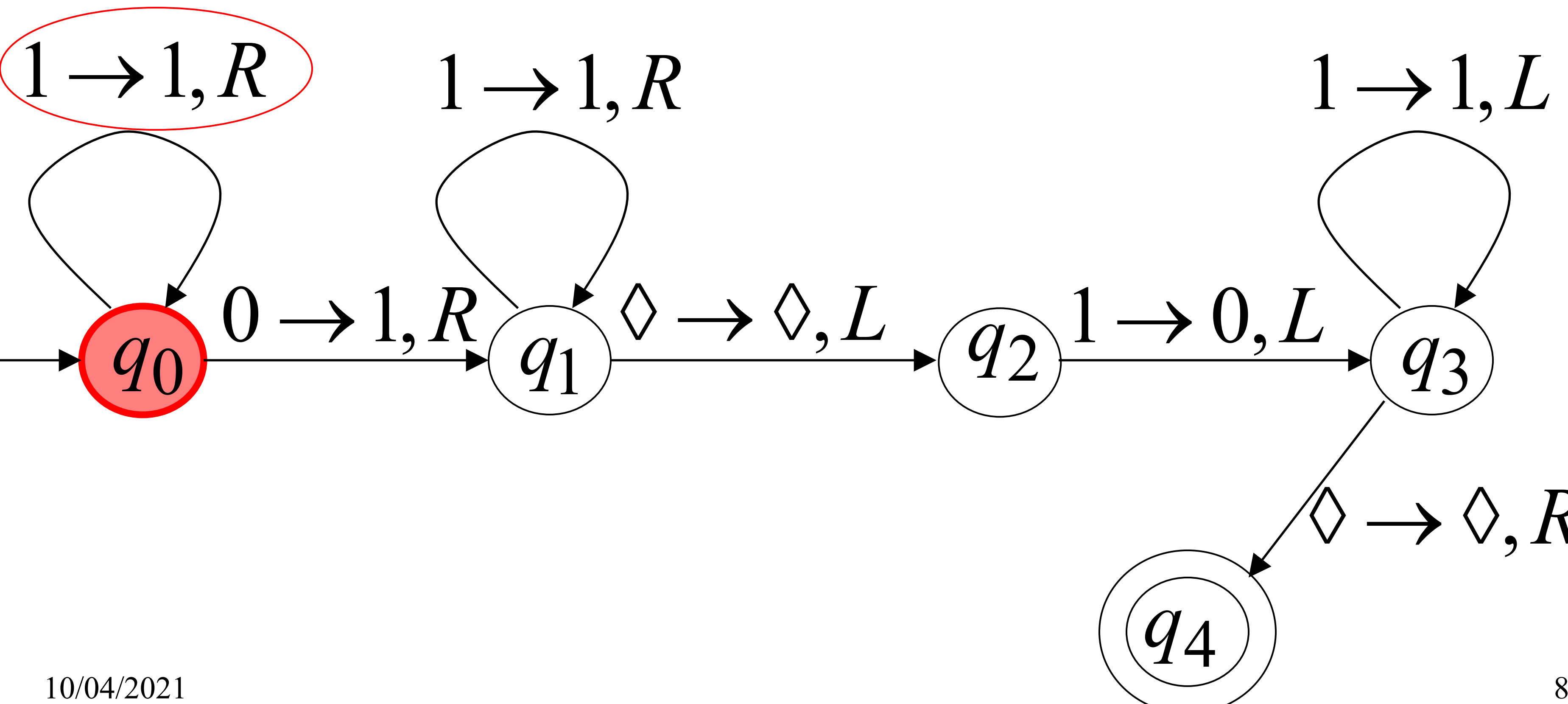


◊
1
1
0
1
1
◊
q_0
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4

Time 1

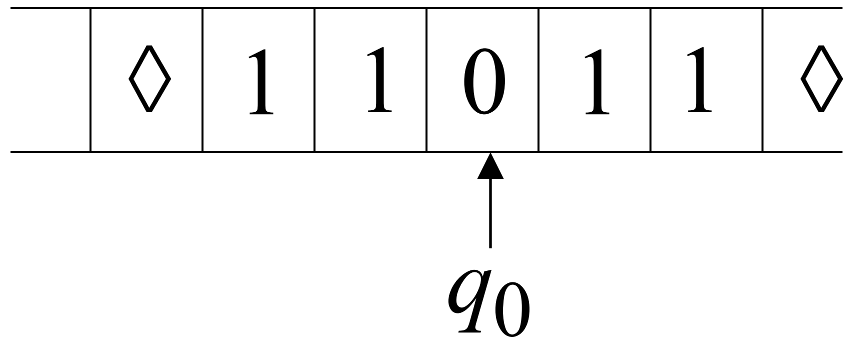


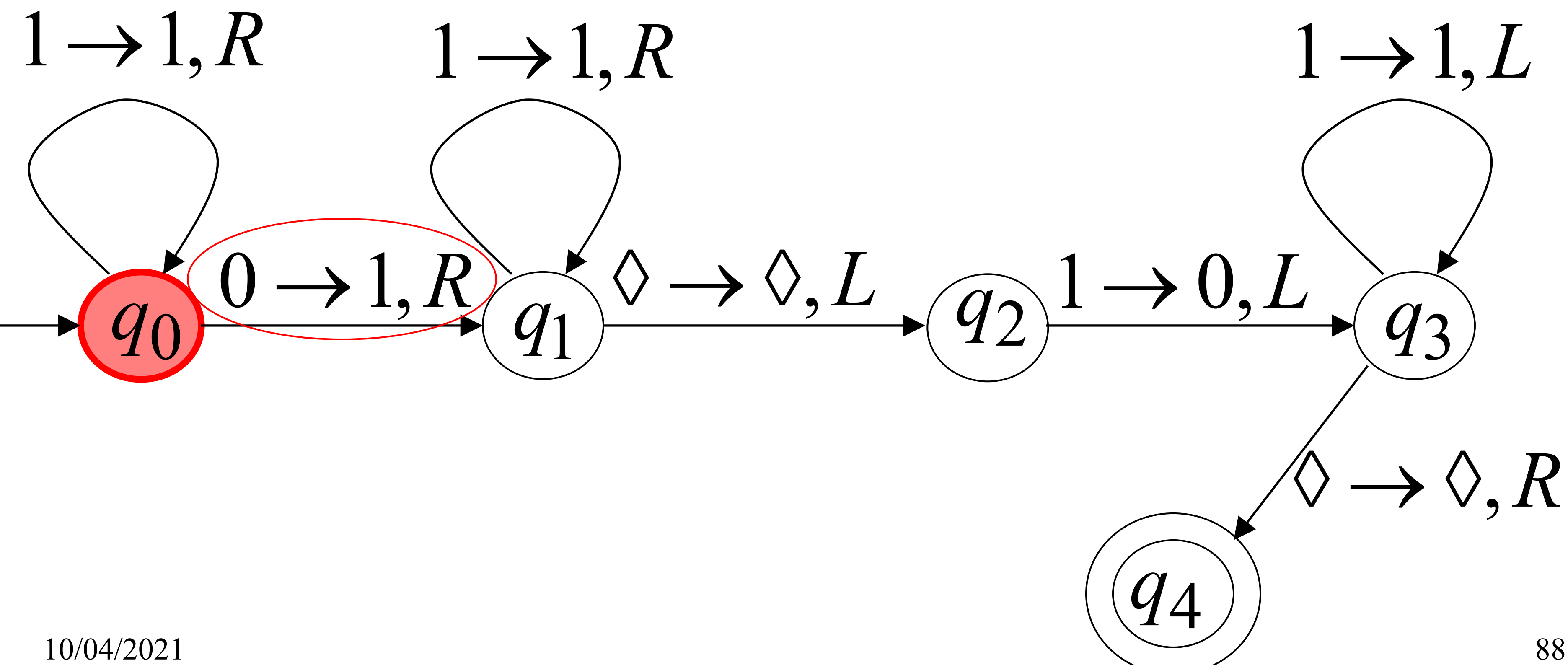
◊ 1 1 0 1 1 ◊
q_0



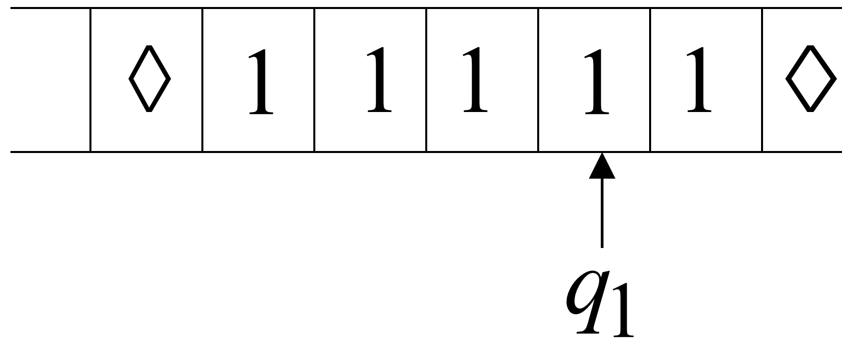
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\Diamond \rightarrow \Diamond, L$
q_2
$1 \rightarrow 0, L$
q_3
$\Diamond \rightarrow \Diamond, R$
q_4

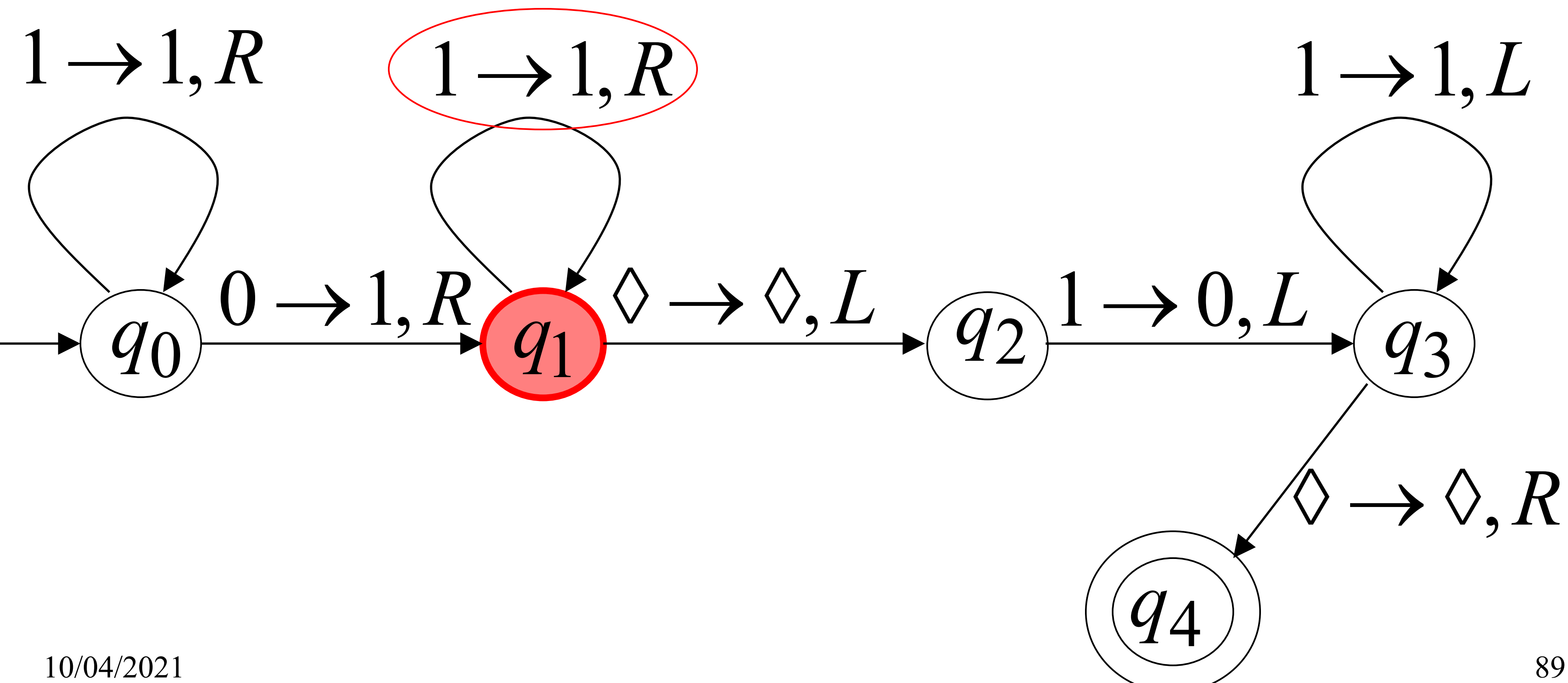
Time 2


◊ 1 1 0 1 1 ◊
q_0

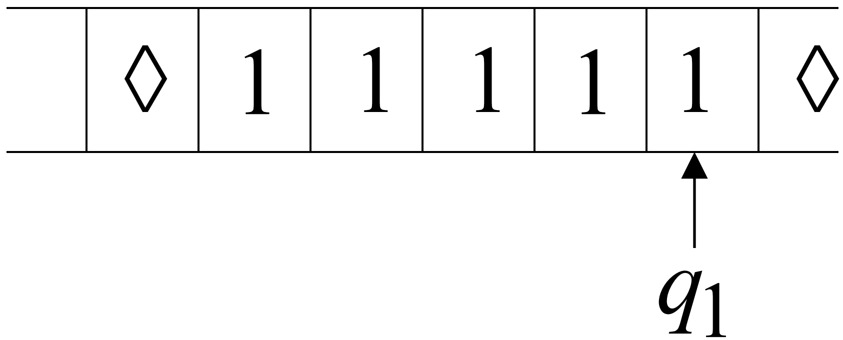

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\Diamond \rightarrow \Diamond, L$
q_2
$1 \rightarrow 0, L$
q_3
$\Diamond \rightarrow \Diamond, R$
q_4

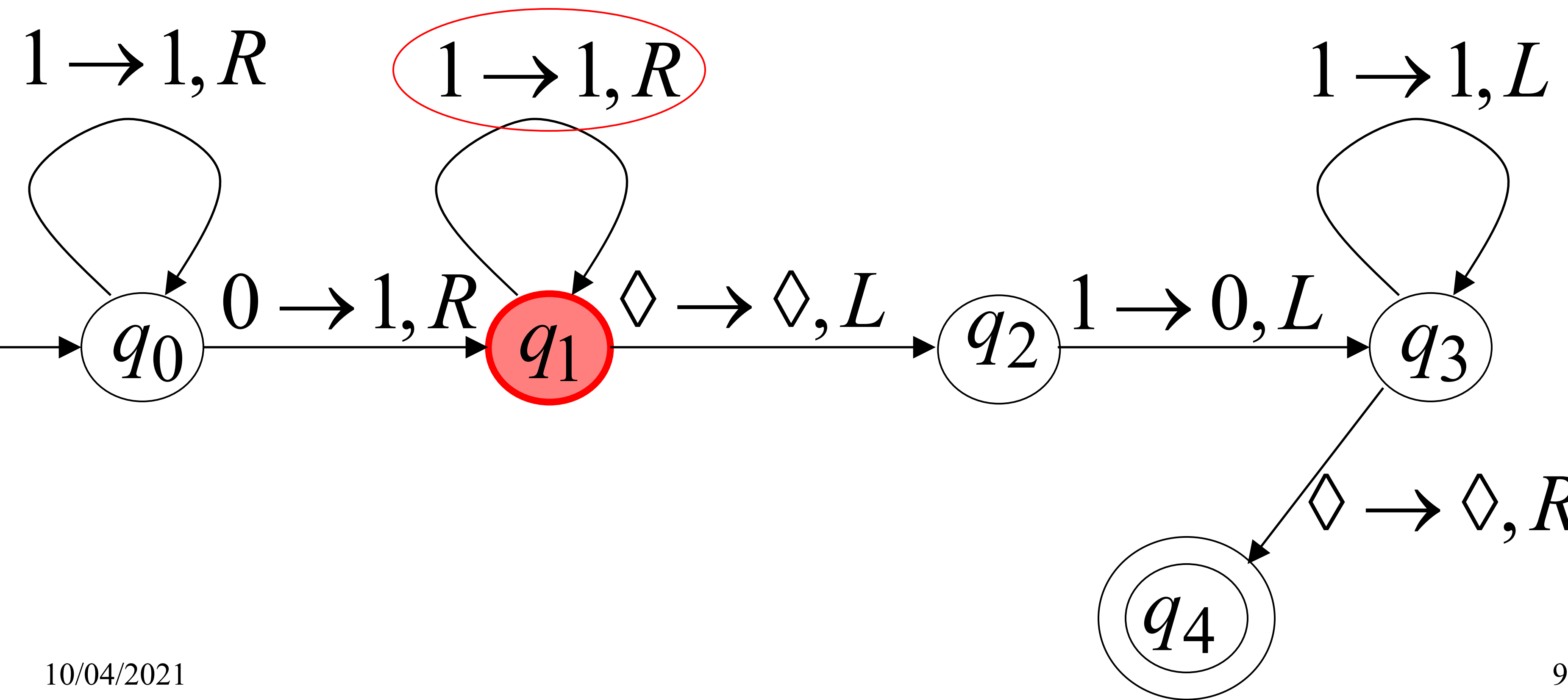
Time 3



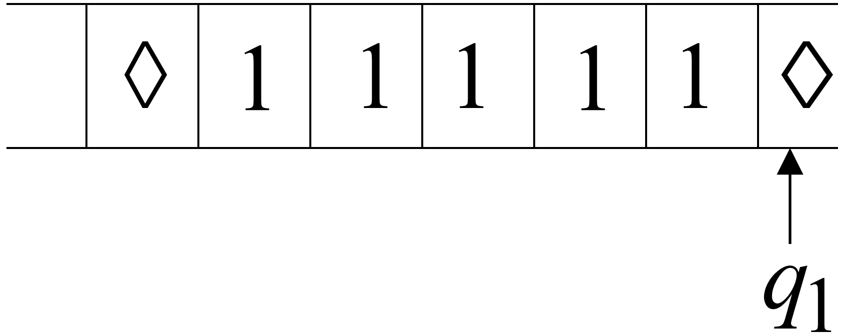


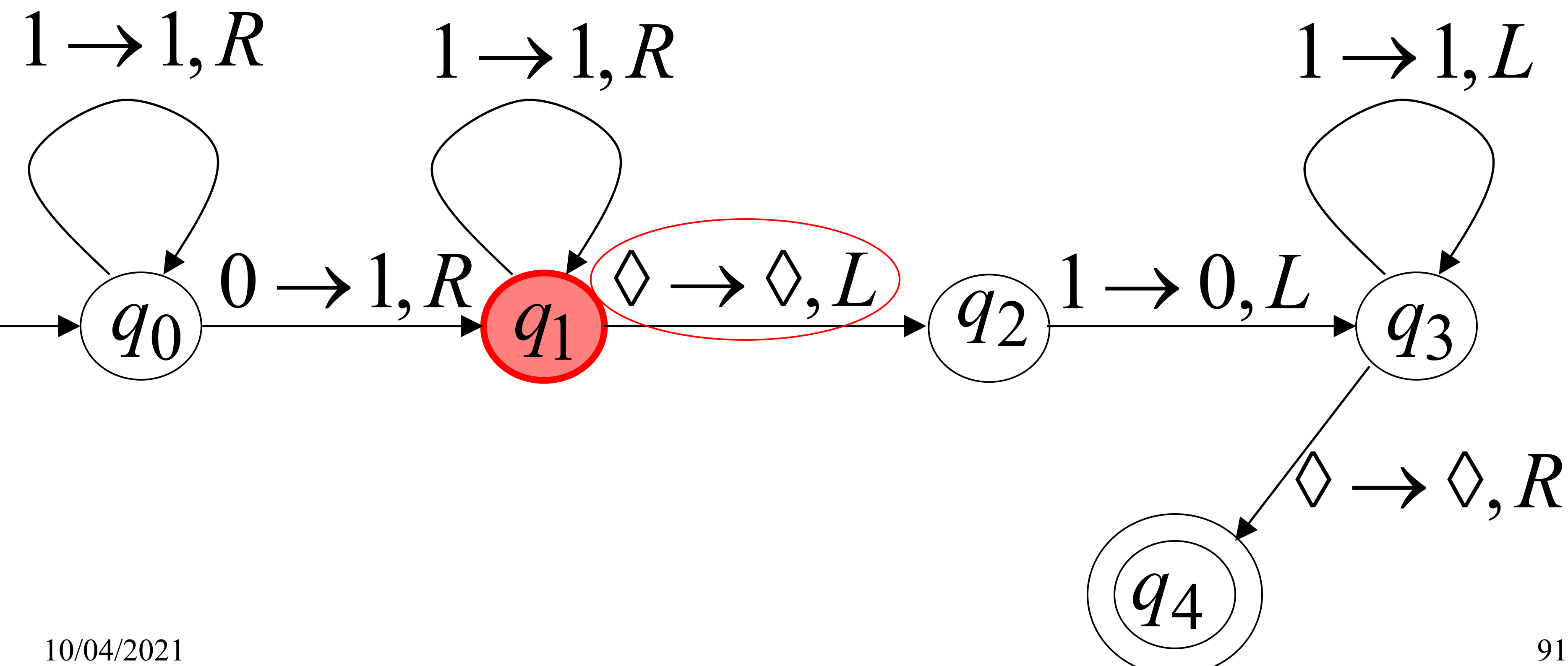
Time 4


◊
1
1
1
1
1
◊
q_1

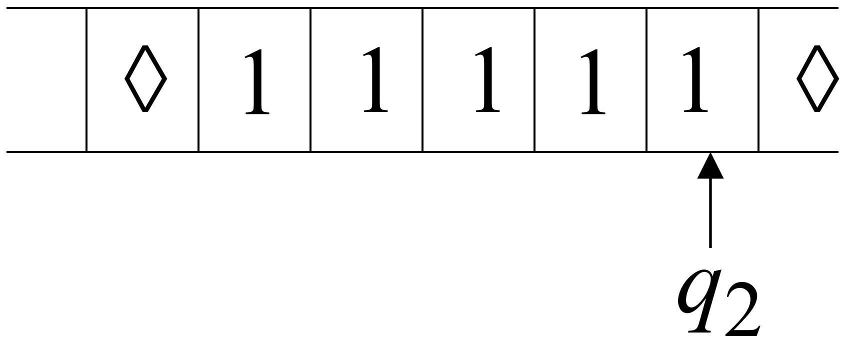

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\lozenge \rightarrow \lozenge, L$
q_2
$1 \rightarrow 0, L$
q_3
$\lozenge \rightarrow \lozenge, R$
q_4

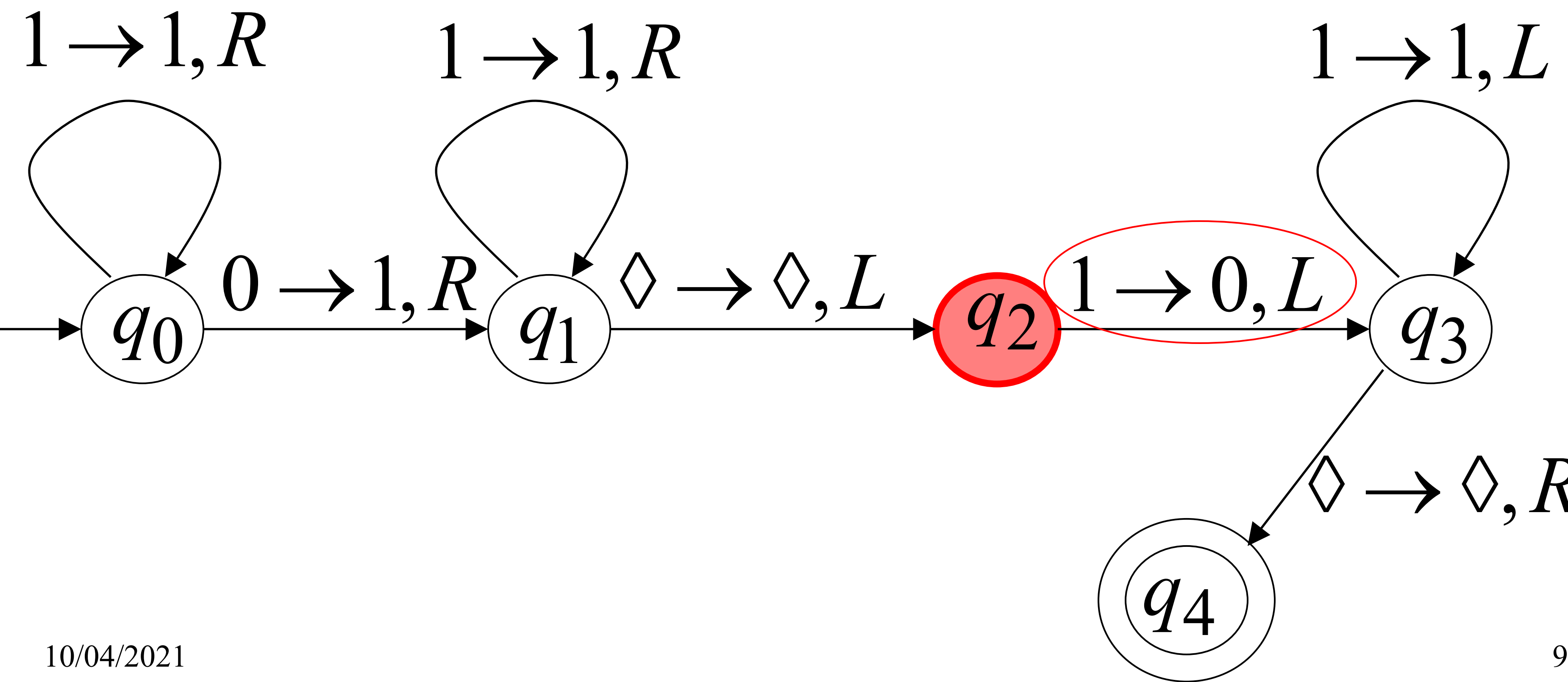
Time 5


◊ 1 1 1 1 1 ◊
q_1

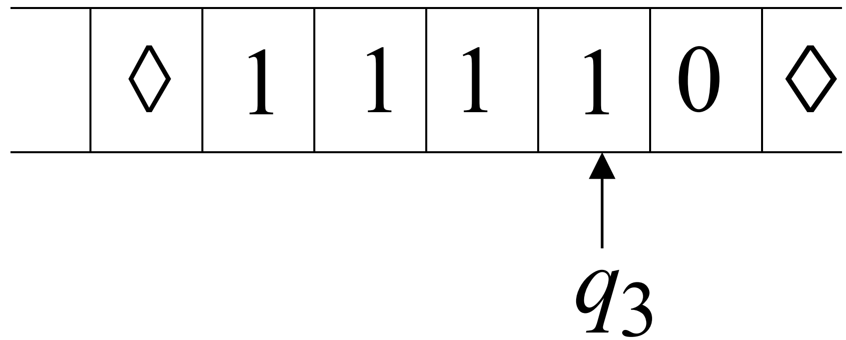

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\Diamond \rightarrow \Diamond, L$
q_2
$1 \rightarrow 0, L$
q_3
$\Diamond \rightarrow \Diamond, R$
q_4

Time 6

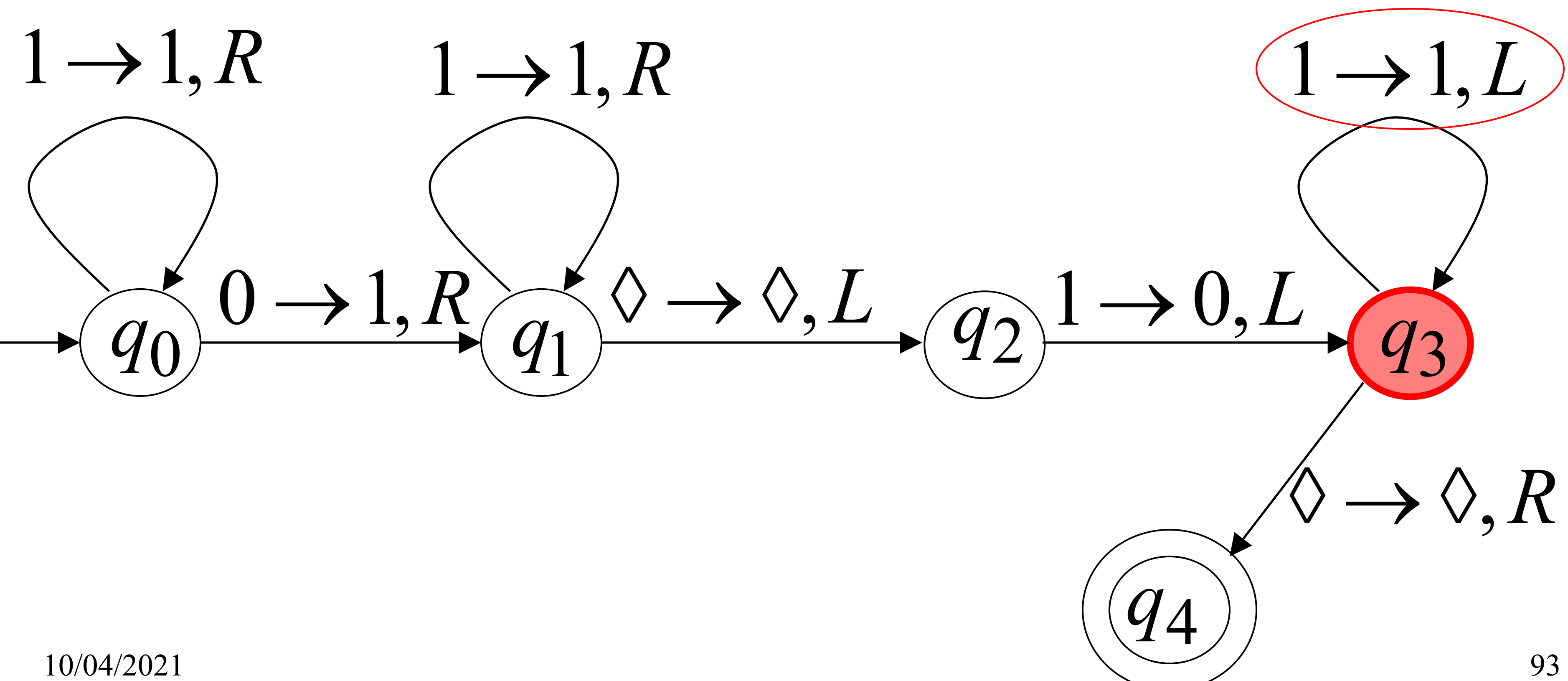

◊
1
1
1
1
1
◊
q_2


$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\lozenge \rightarrow \lozenge, L$
q_2
$1 \rightarrow 0, L$
q_3
$\lozenge \rightarrow \lozenge, R$
q_4

Time 7

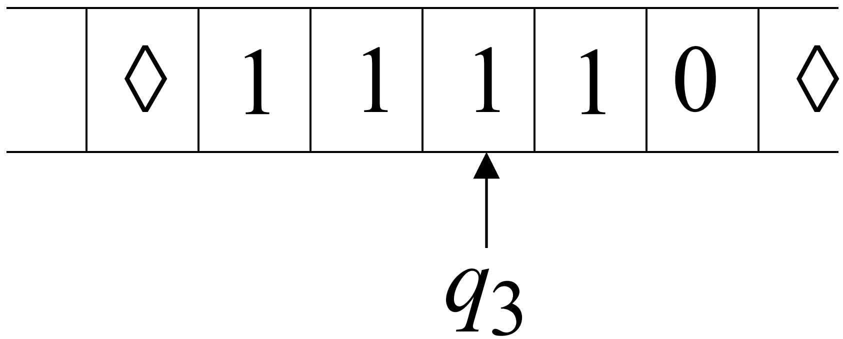


◊
1
1
1
1
0
◊
q_3

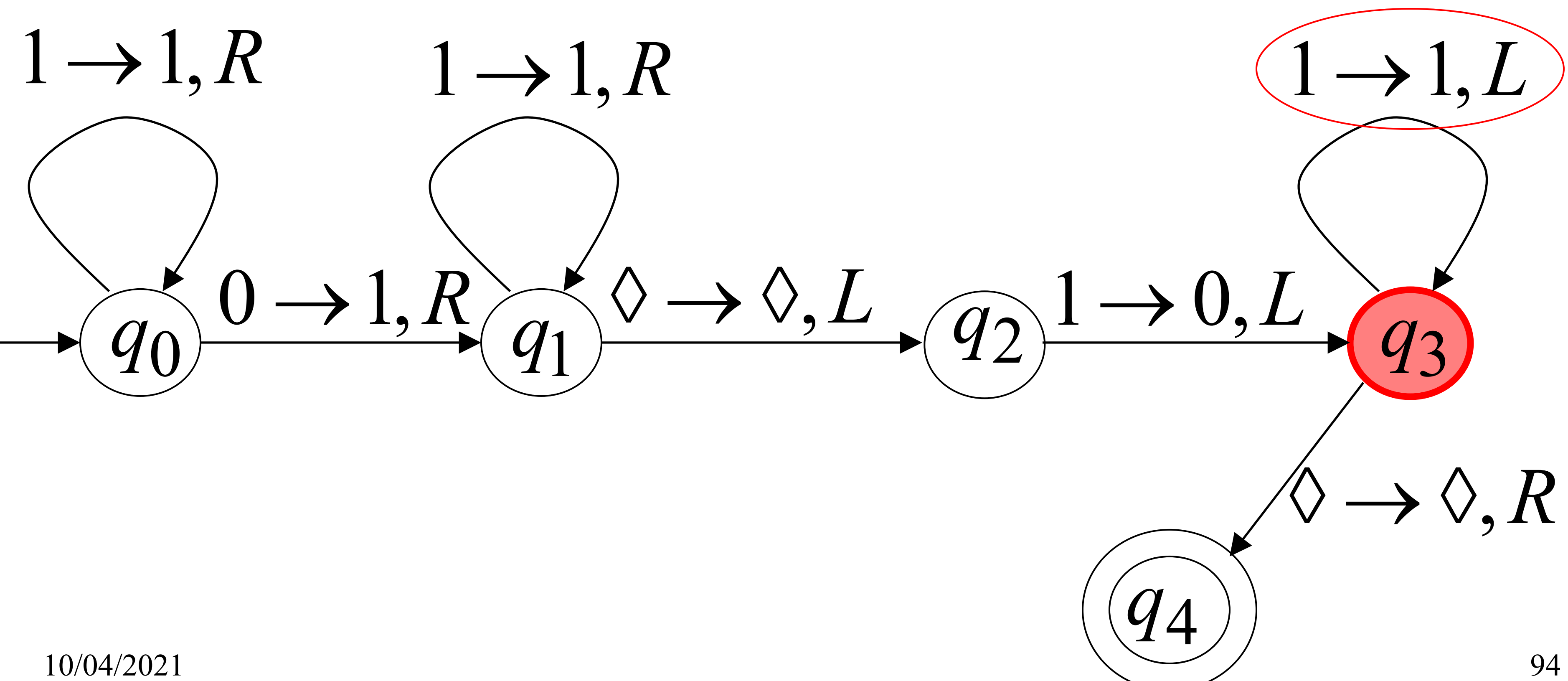


$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\Diamond \rightarrow \Diamond, L$
q_2
$1 \rightarrow 0, L$
q_3
$\Diamond \rightarrow \Diamond, R$
q_4

Time 8

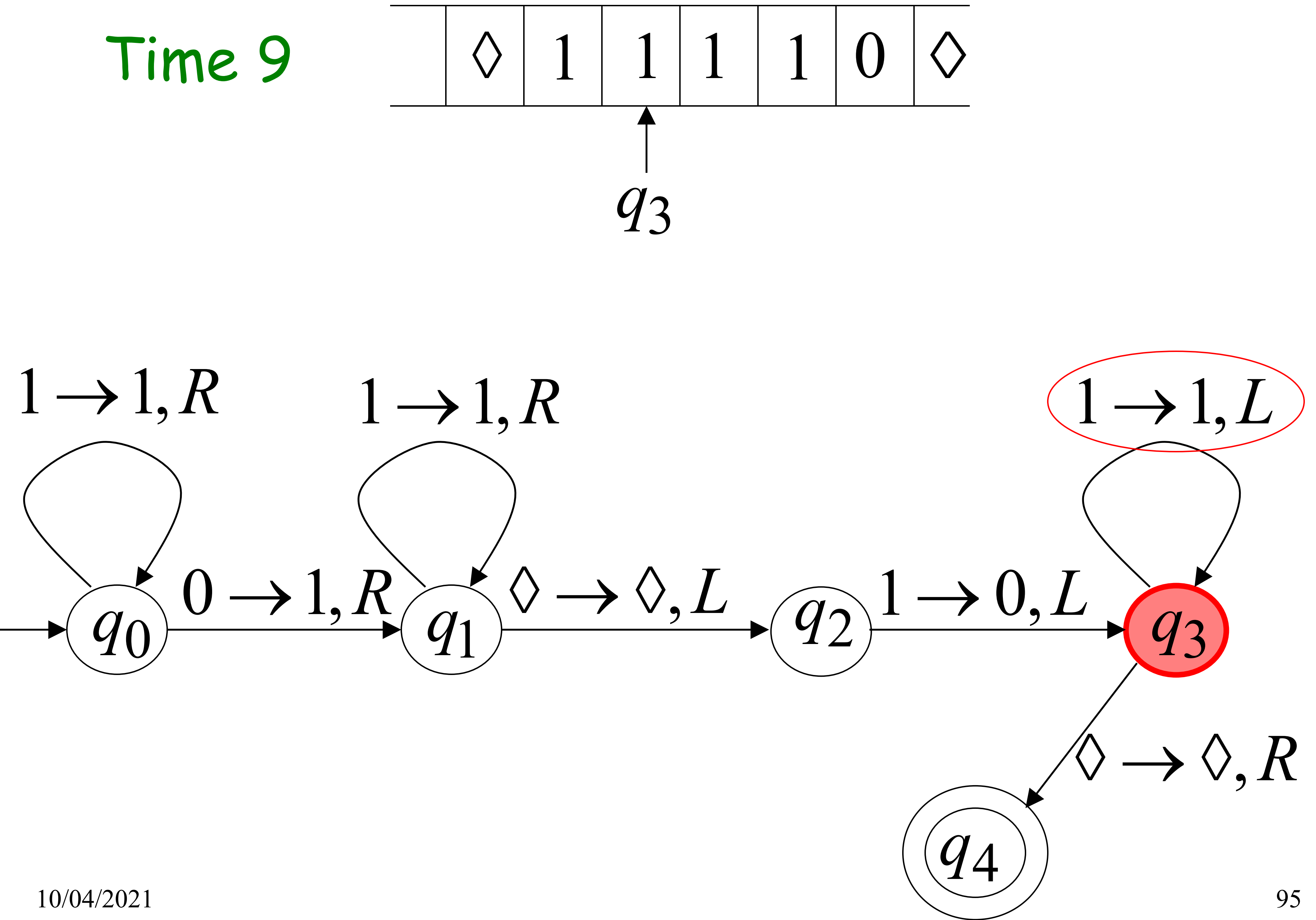


◊
1
1
1
1
0
◊
q_3



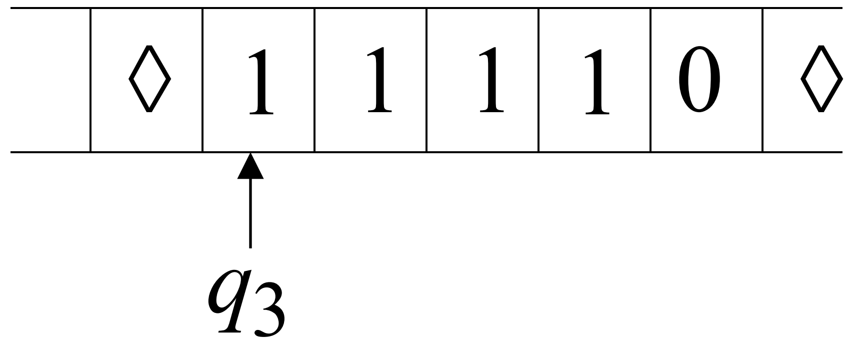
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4

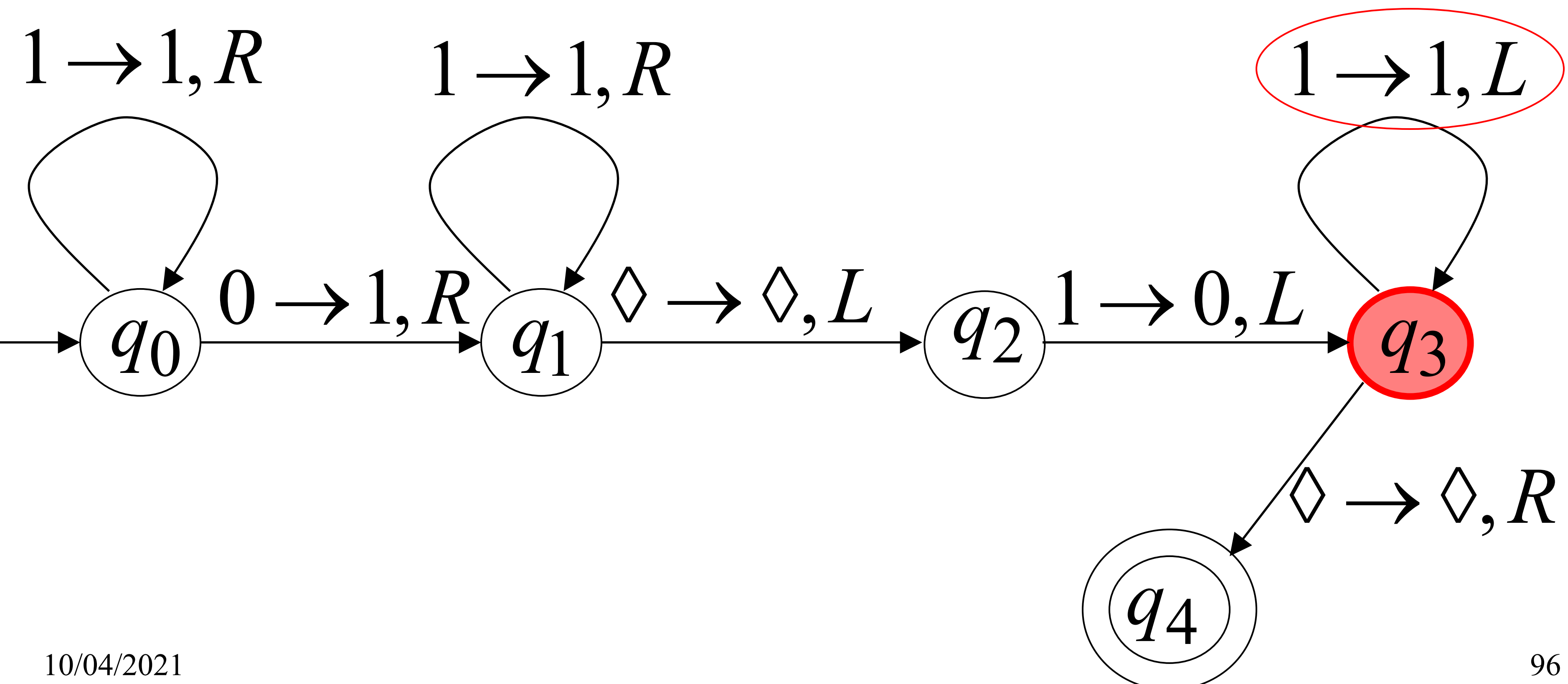
Time 9



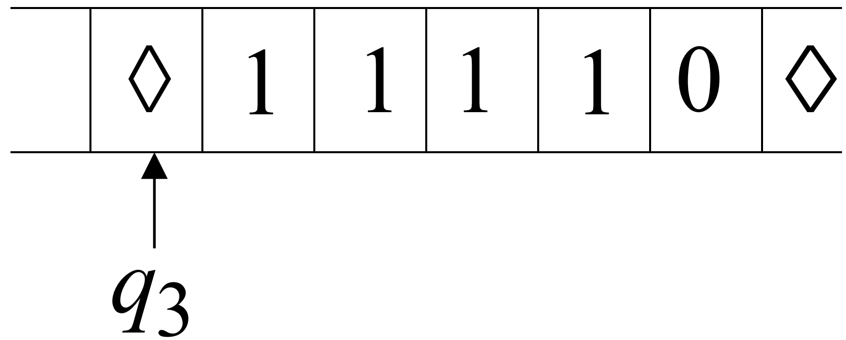
◊
1
1
1
1
0
◊
q_3
$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
q_0
$0 \to 1, R$
q_1
$\Diamond \to \Diamond, L$
q_2
$1 \to 0, L$
q_3
$\Diamond \to \Diamond, R$
q_4

Time 10

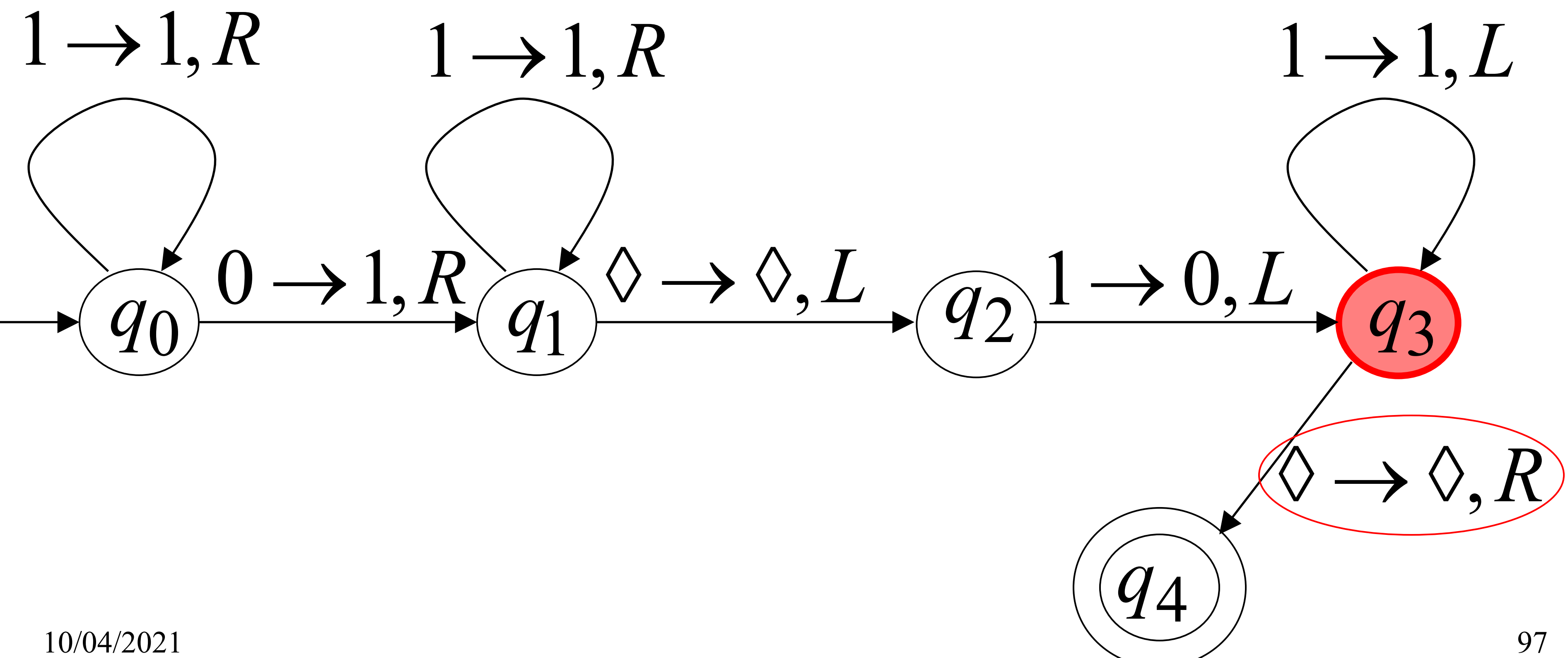

◊
1
1
1
1
0
◊
q_3


$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\Diamond \rightarrow \Diamond, L$
q_2
$1 \rightarrow 0, L$
q_3
$\Diamond \rightarrow \Diamond, R$
q_4

Time 11

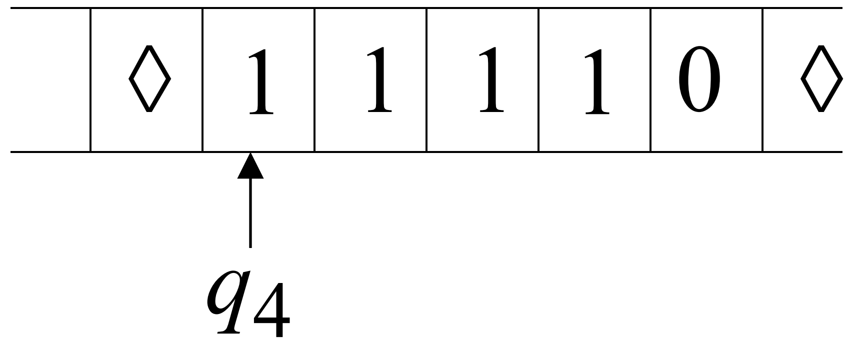


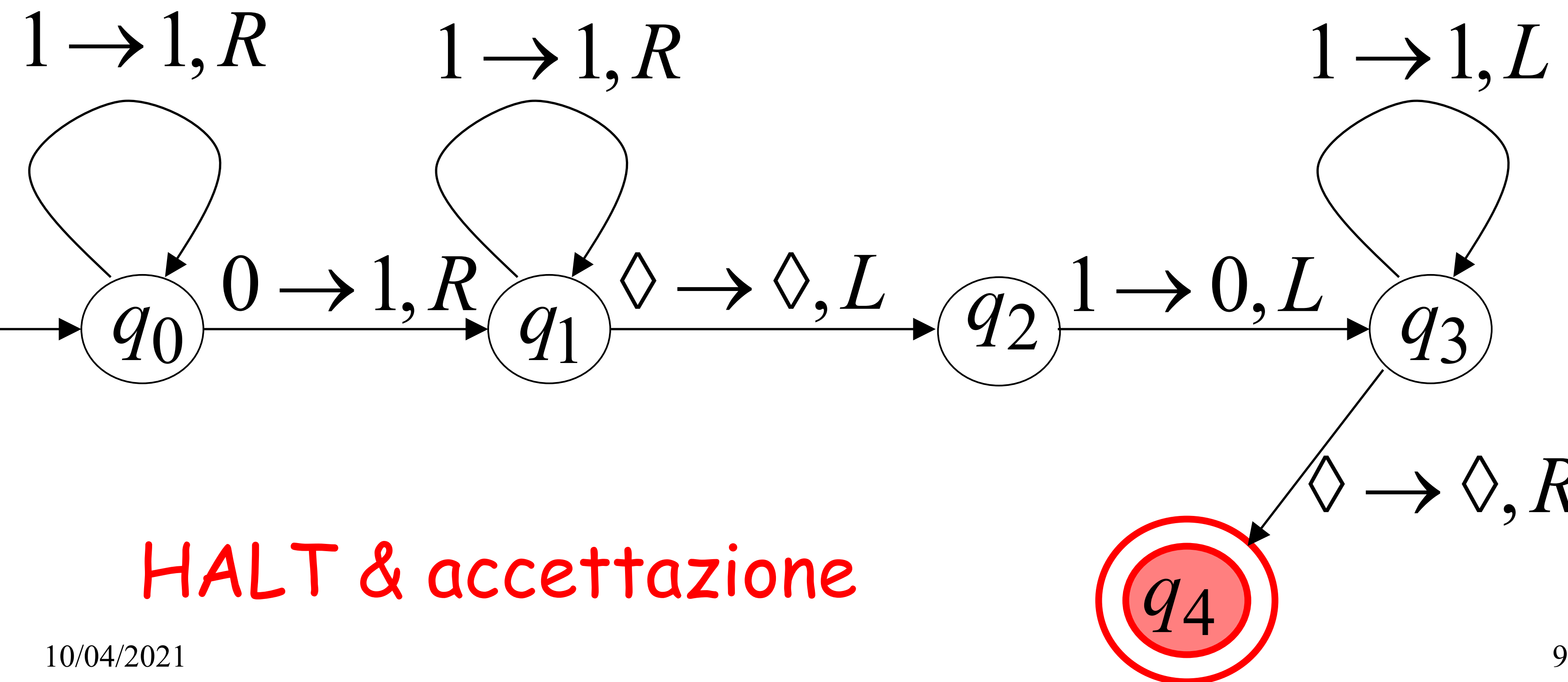
◊
1
1
1
1
0
◊
q_3



$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4

Time 12





HALT & accettazione

Un altro esempio

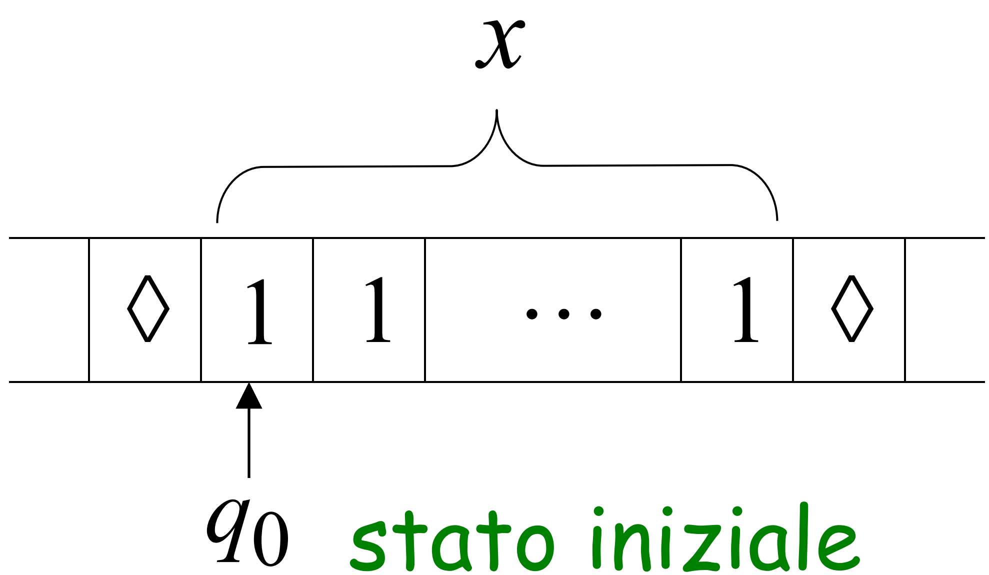
La funzione Che raddoppia il numero di 1 è calcolabile

Macchina di Turing :

stringa di input: x unario

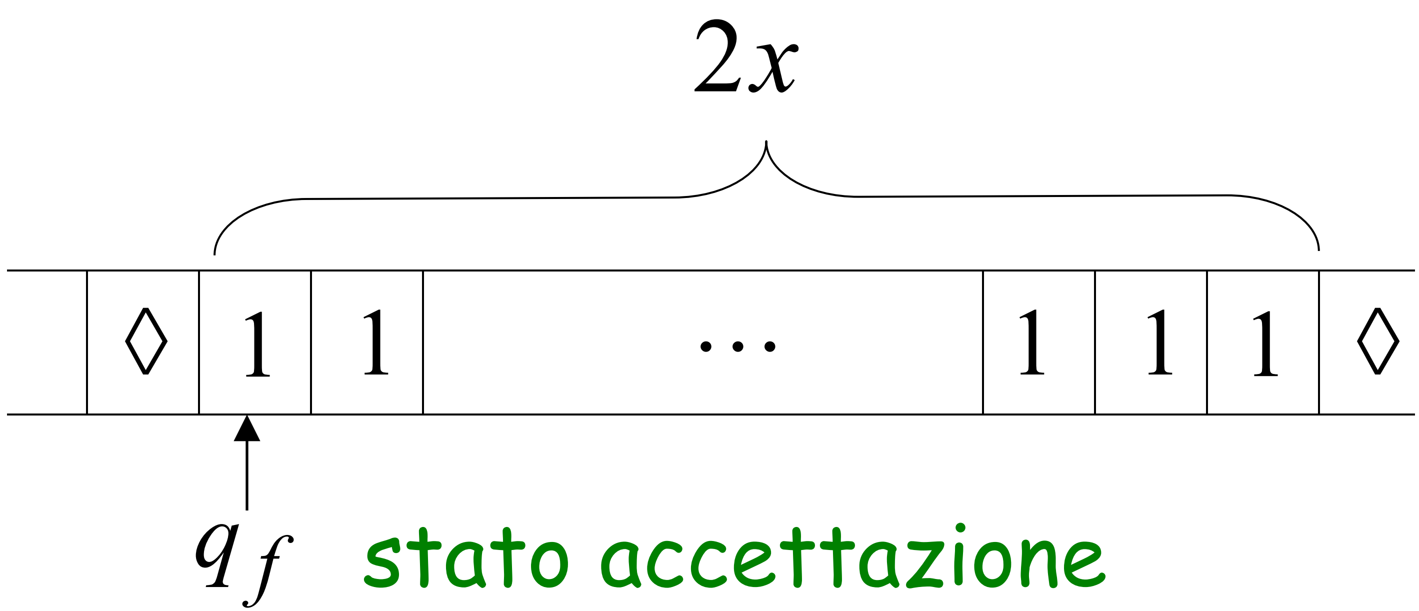
Output string: xx unario

Start



stato iniziale

Finish



stato accettazione

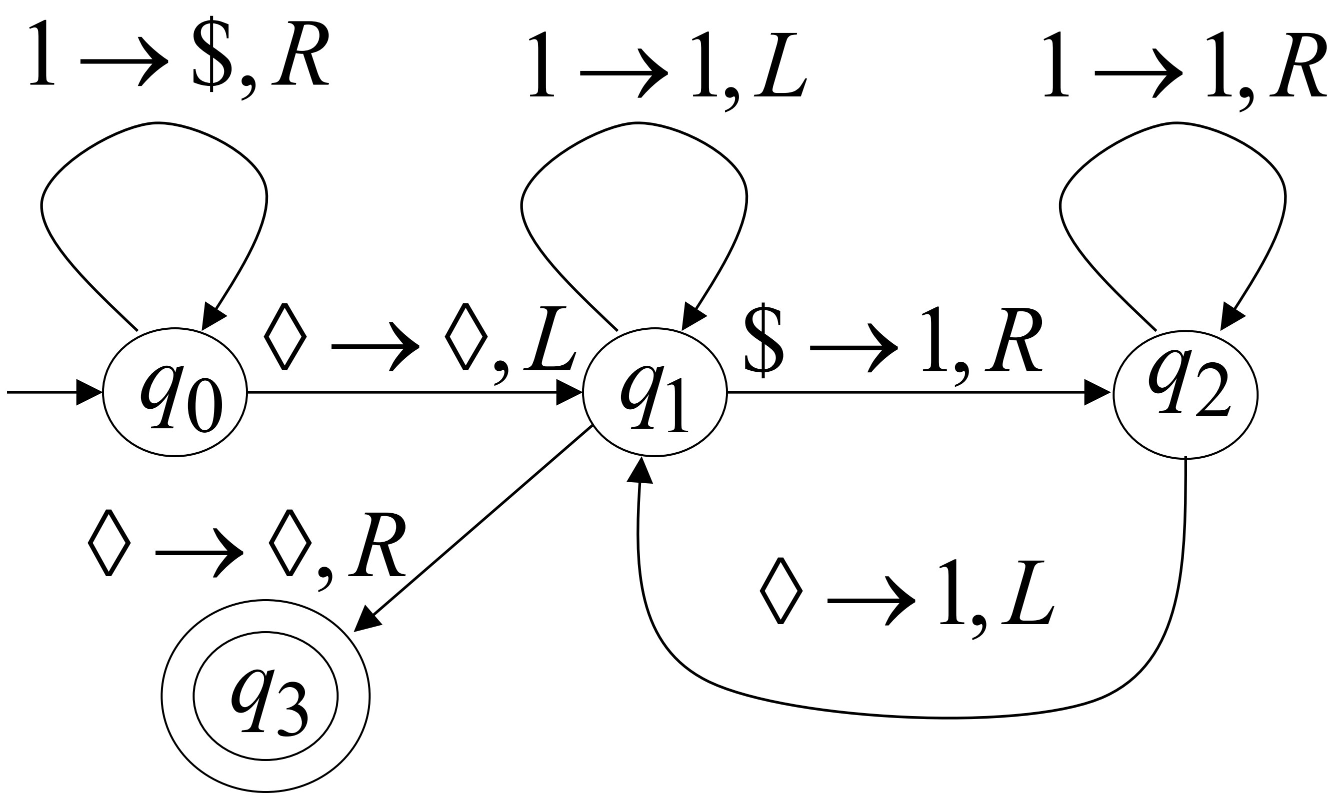
Pseudocodice per macchina Turing

$$f(x) = 2x$$

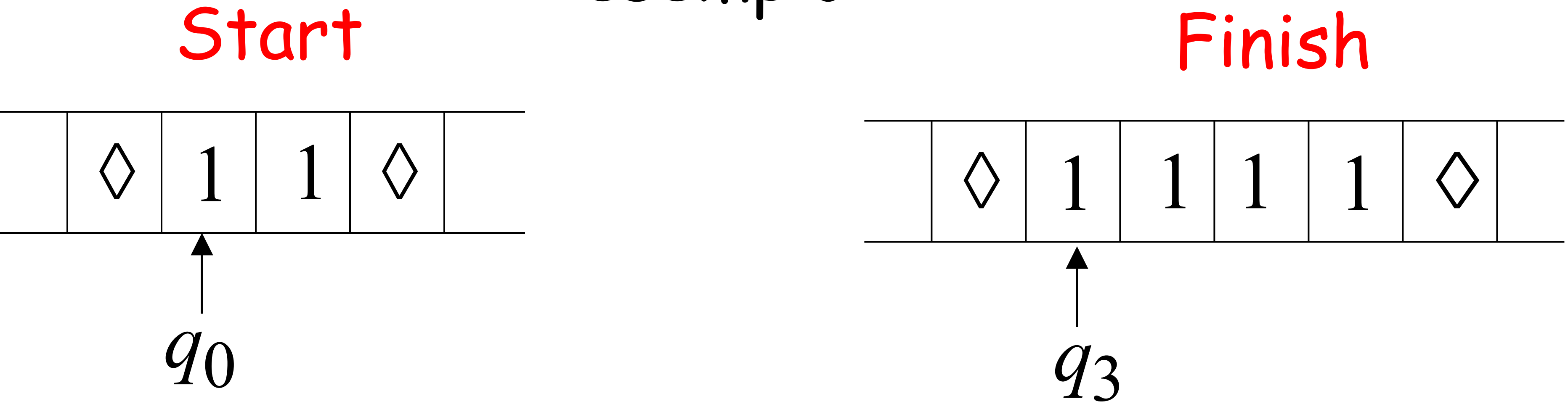
- ogni 1 diventa \$
- Repeat:
  - trova il \$ più a destra, cambia in 1
  - vai alla fine a destra, inserisci 1

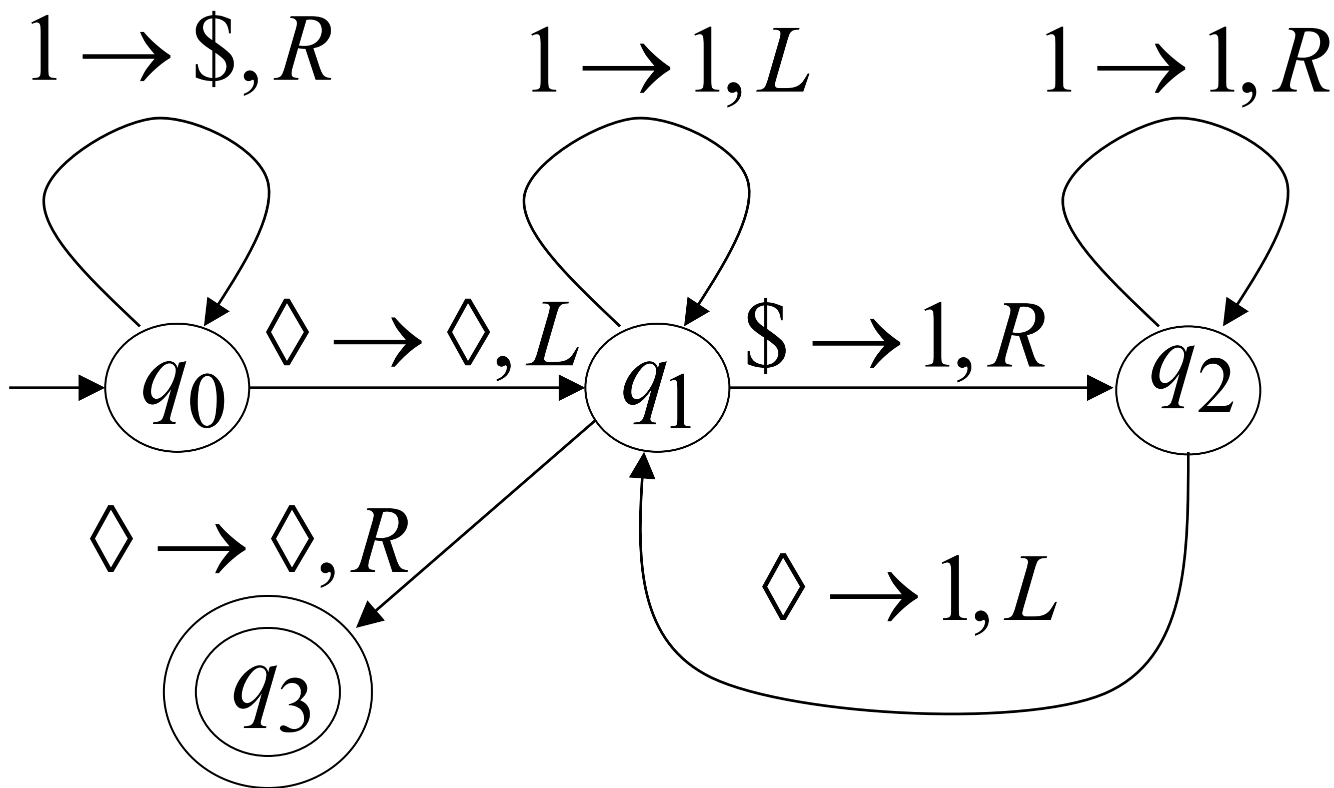
Until no \$ rimangono

Turing macchina per $f(x) = xx$



esempio


Start
◊
1
1
◊
q_0
Finish
◊
1
1
1
1
◊
q_3


$1 \to \$, R$
$1 \to 1, L$
$1 \to 1, R$
$\Diamond \to \Diamond, L$
$\$ \to 1, R$
q_0
q_1
q_2
$\Diamond \to \Diamond, R$
$\Diamond \to 1, L$
q_3

Copia a distanza di una stringa

Es ▸1111 dà 111101111

altro esempio

La funzione

È calcolabile

$$f(x,y) = \begin{cases} 1 & \text{if } x > y \\ 0 & \text{if } x \leq y \end{cases}$$

Input: $x0y$

Output: 1 or 0

macchina di Turing Pseudocodice:

- Repeat

verifica ogni 1 da x con 1 fda y

Until tutti gli 1 di x or y sono verificate

- If un 1 da x non è verificato

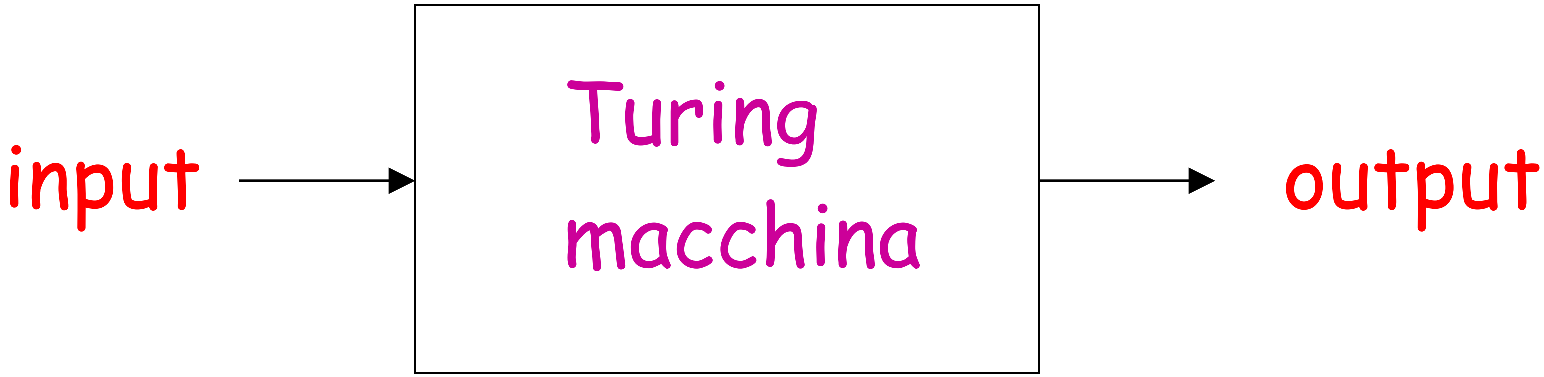
cancella tape, scrivi 1 $(x > y)$

else

cancella tape, scrivi 0 $(x \leq y)$

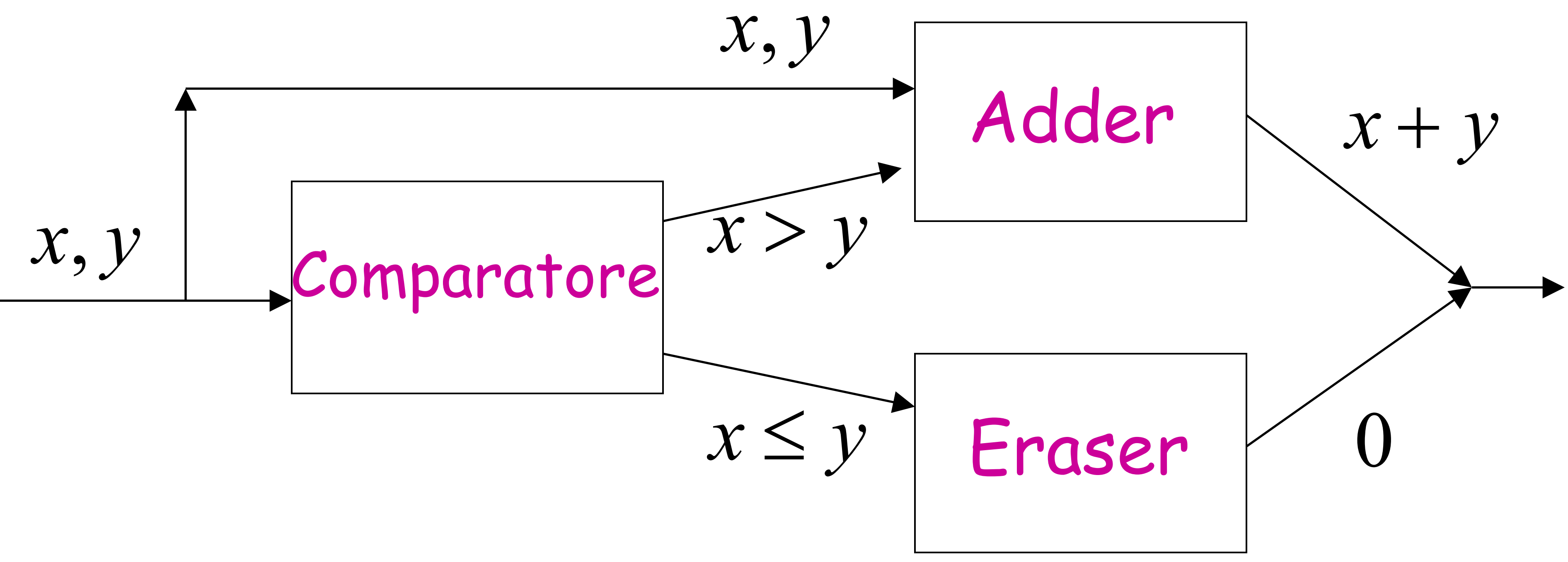
Mettere insieme macchine di turing

Block Diagram



esempio:

$$f(x,y)=\begin{cases} x+y & \text{if} \quad x>y \\ 0 & \text{if} \quad x\leq y \end{cases}$$



Ricordiamoci sempre

Calcolo standard salvando gli input.