CS116-Automata Theory and Formal Languages

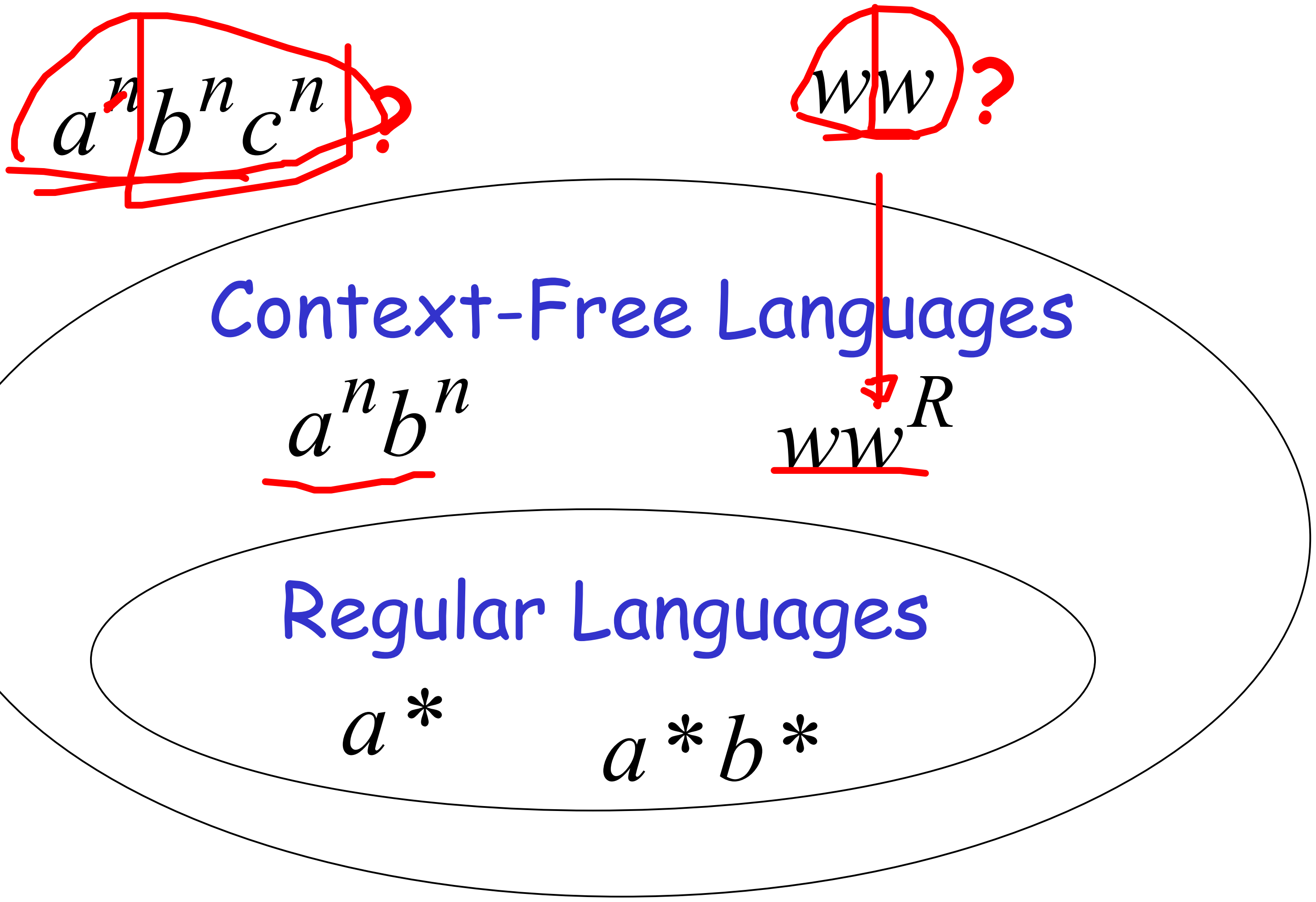
Lecture 11

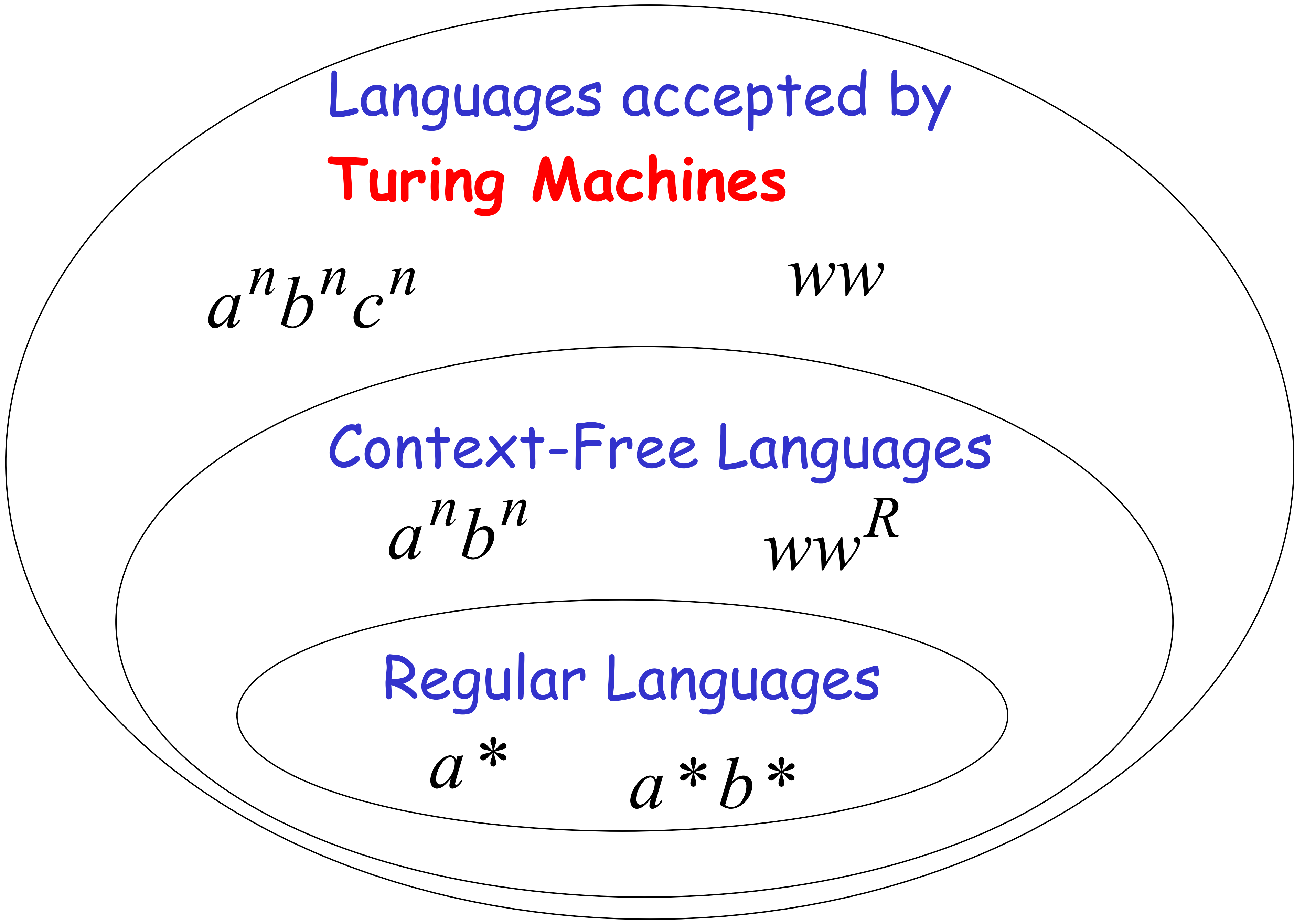
Turing Machines

Computer Science Department

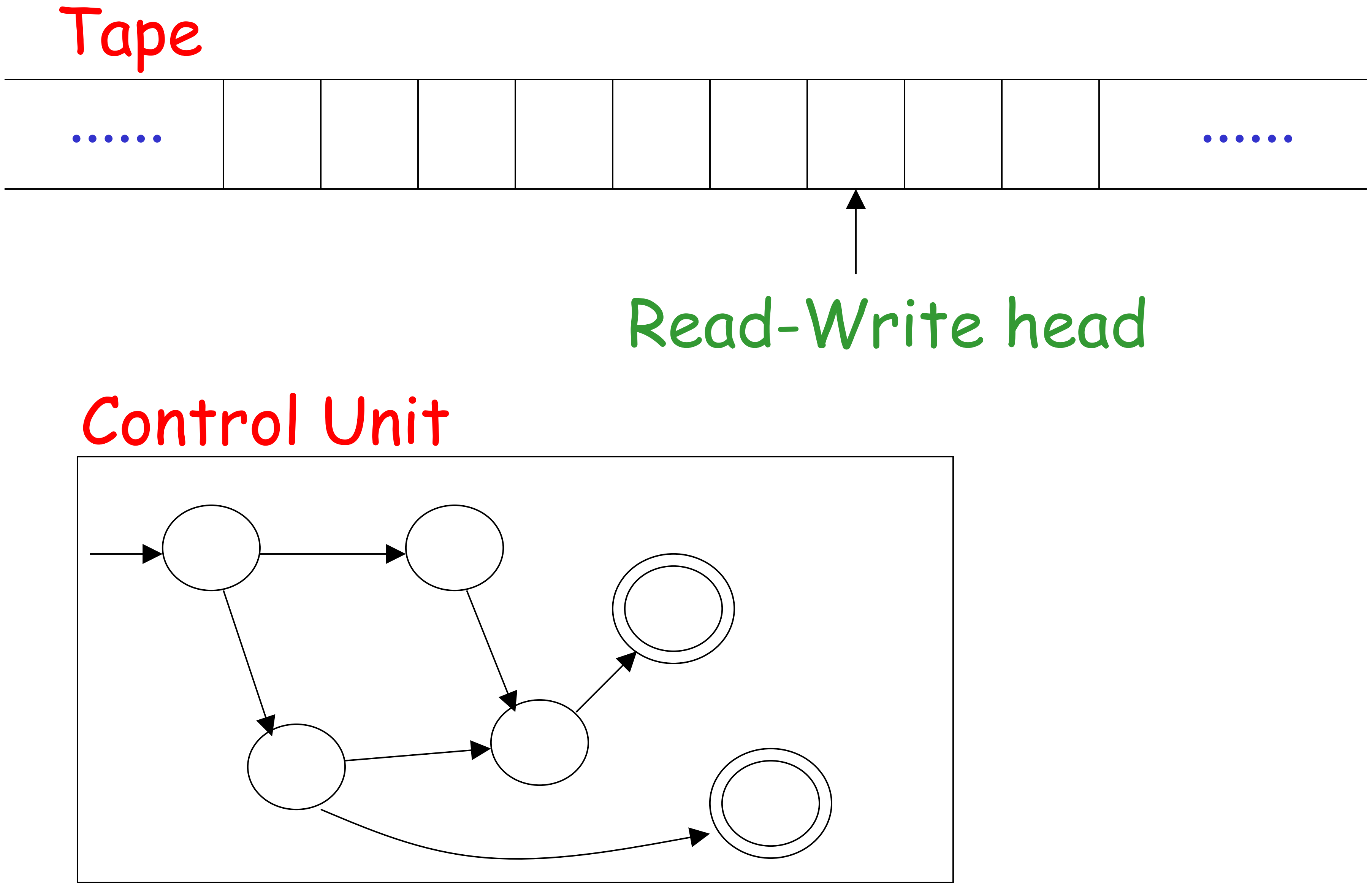
1st Semester 2025-2026

The Language Hierarchy



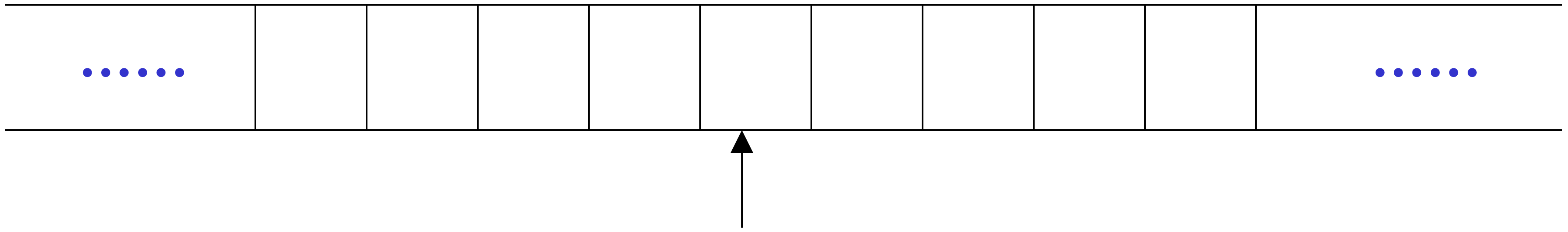

Languages accepted by
Turing Machines
$a^n b^n c^n$
ww
Context-Free Languages
$a^n b^n$
ww^R
Regular Languages
a^*
$a^* b^*$

A Turing Machine



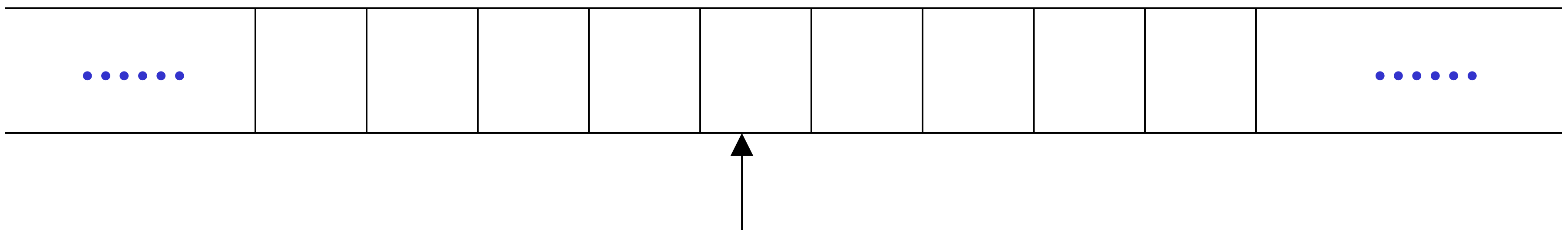
The Tape

No boundaries -- infinite length



Read-Write head

The head moves Left or Right

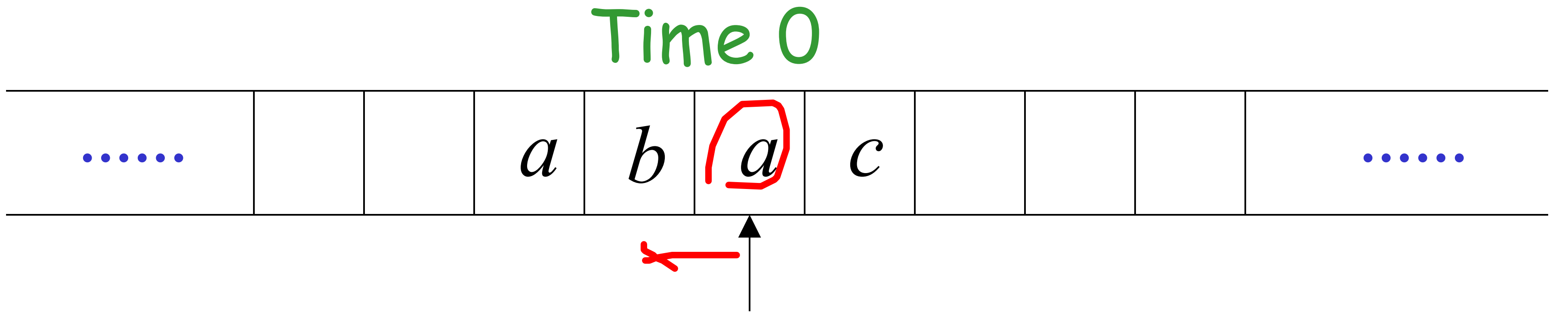


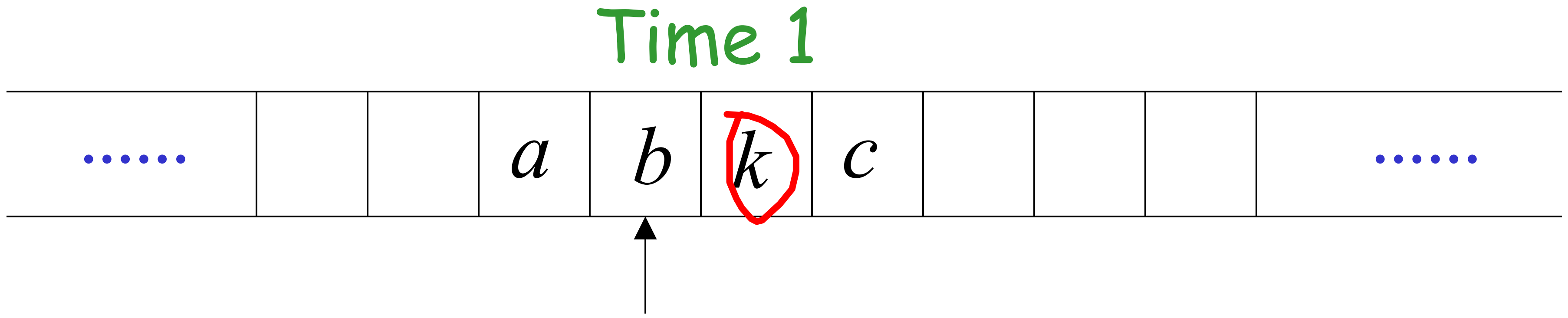
Read-Write head

The head at each transition (time step):

1. Reads a symbol
2. Writes a symbol
3. Moves Left or Right

Example:



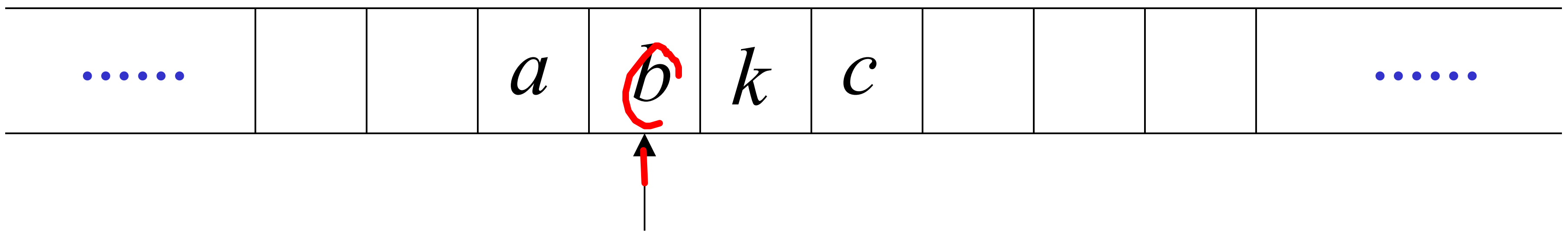


1. Reads a

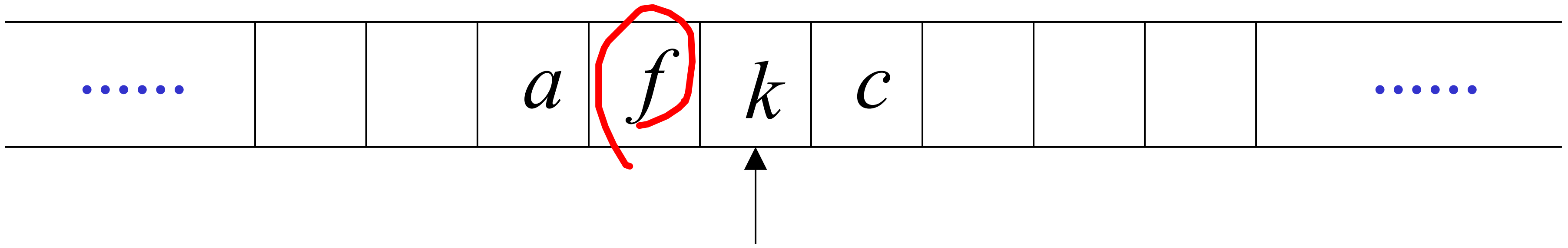
2. Writes k

3. Moves Left

Time 1

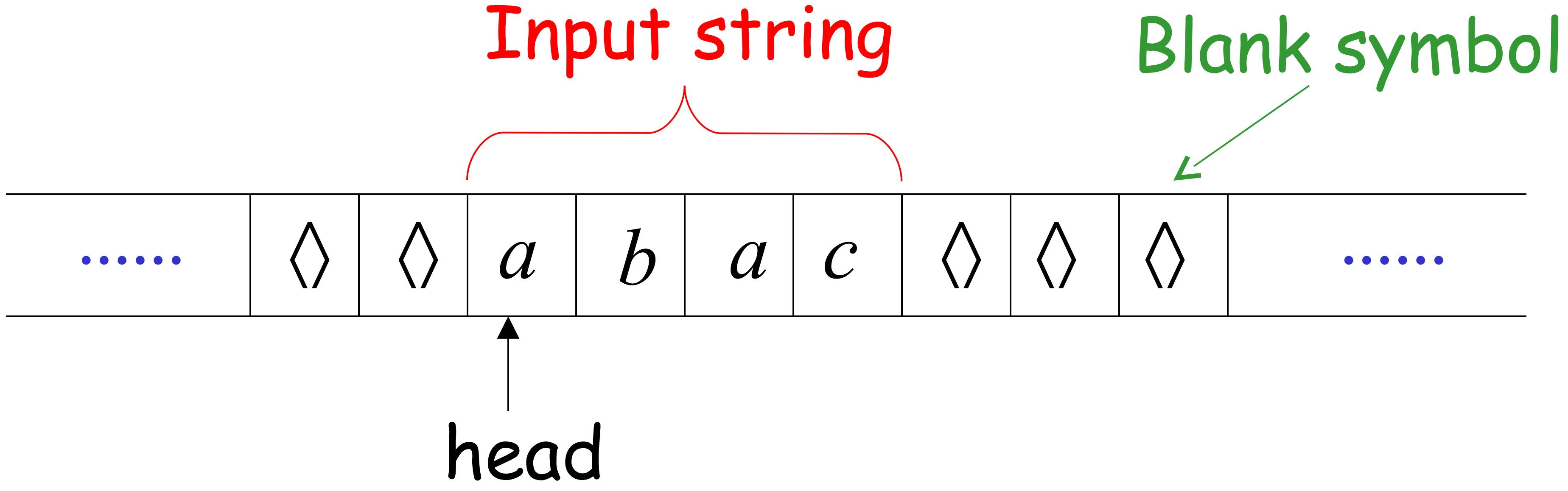


Time 2



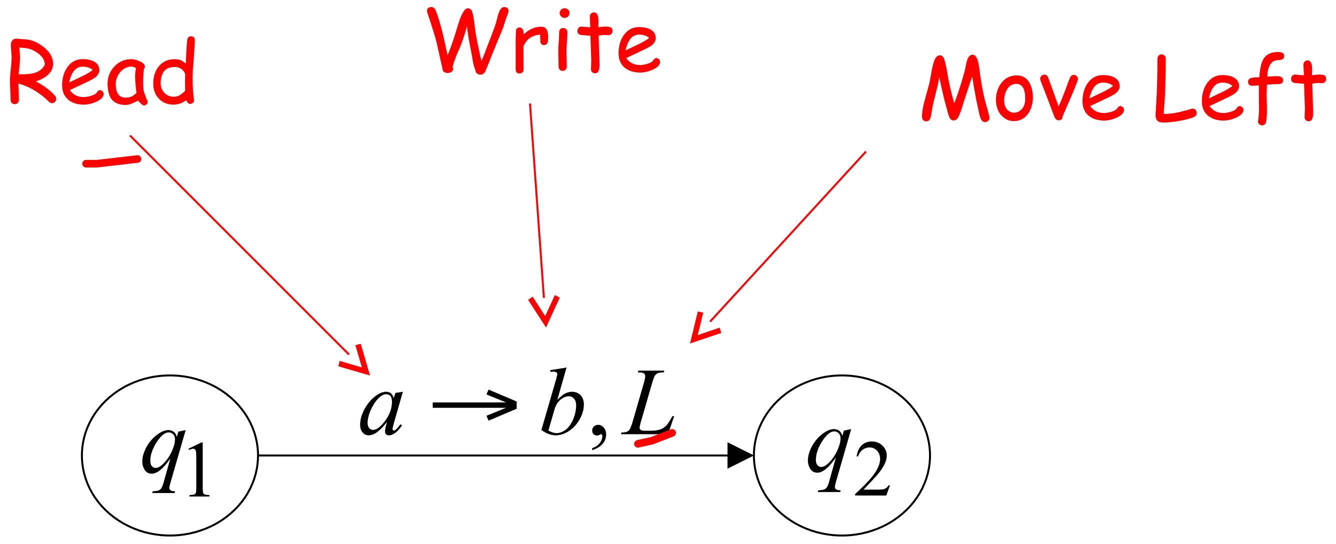
1. Reads b
2. Writes $\underline{\ }f$
3. Moves Right

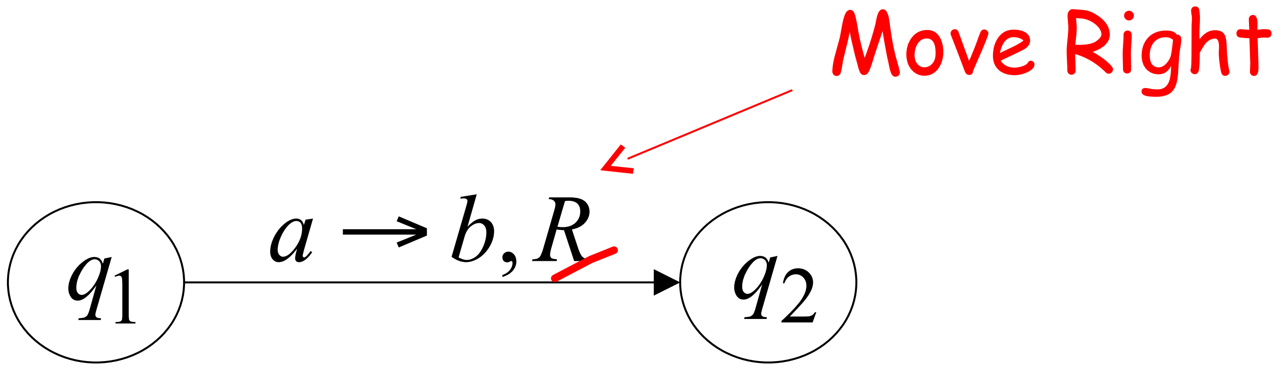
The Input String



Head starts at the leftmost position of the input string

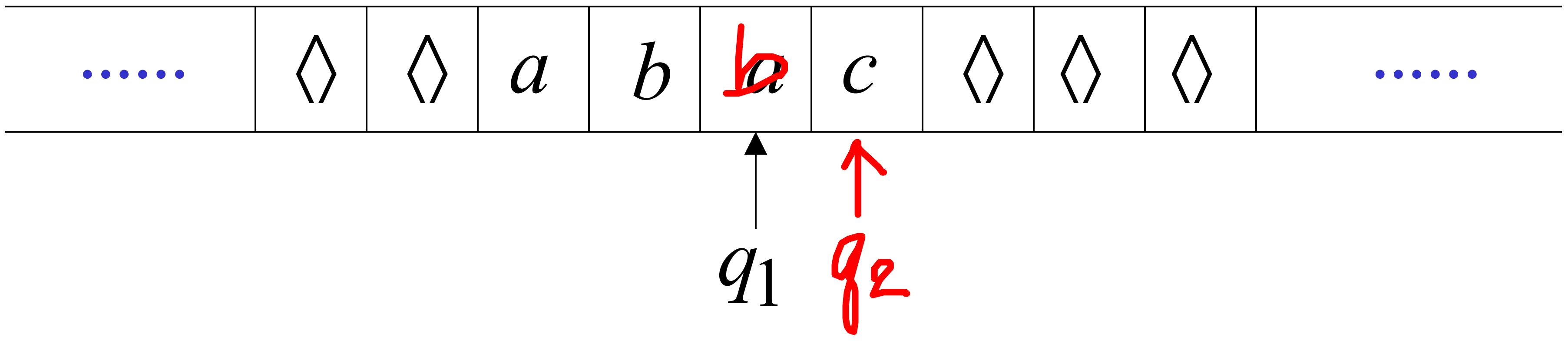
States & Transitions



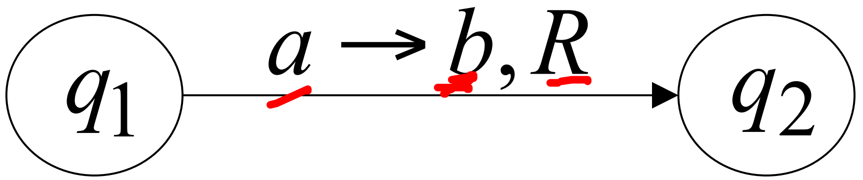


Example:

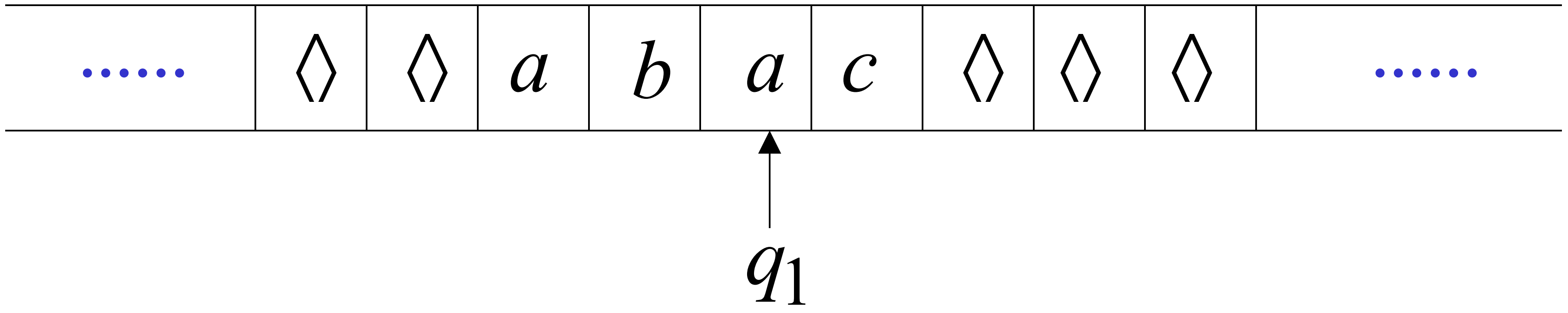
Time 1



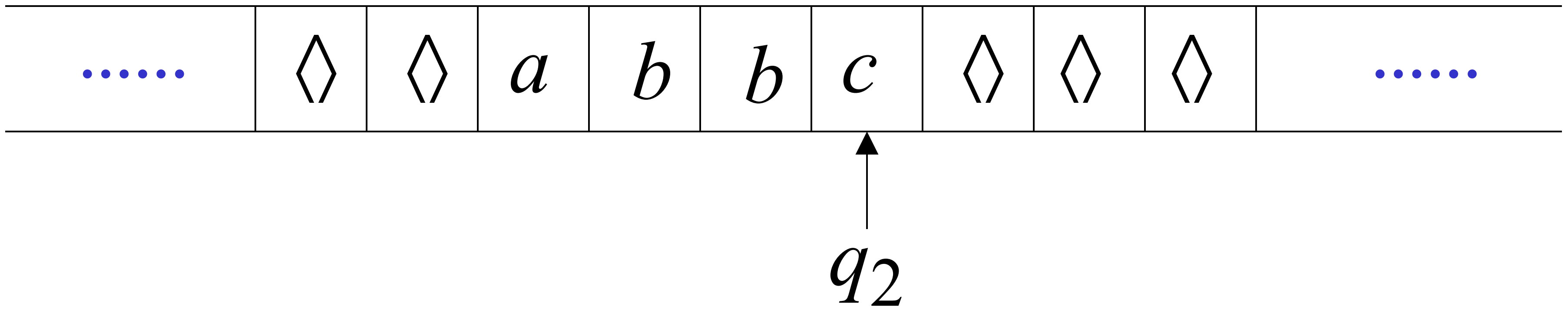
current state



Time 1

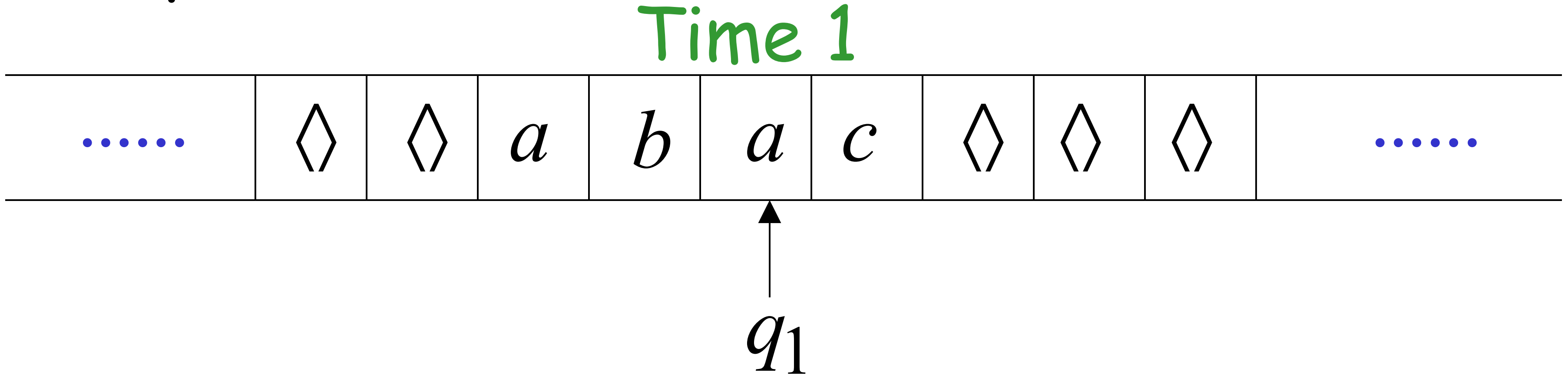


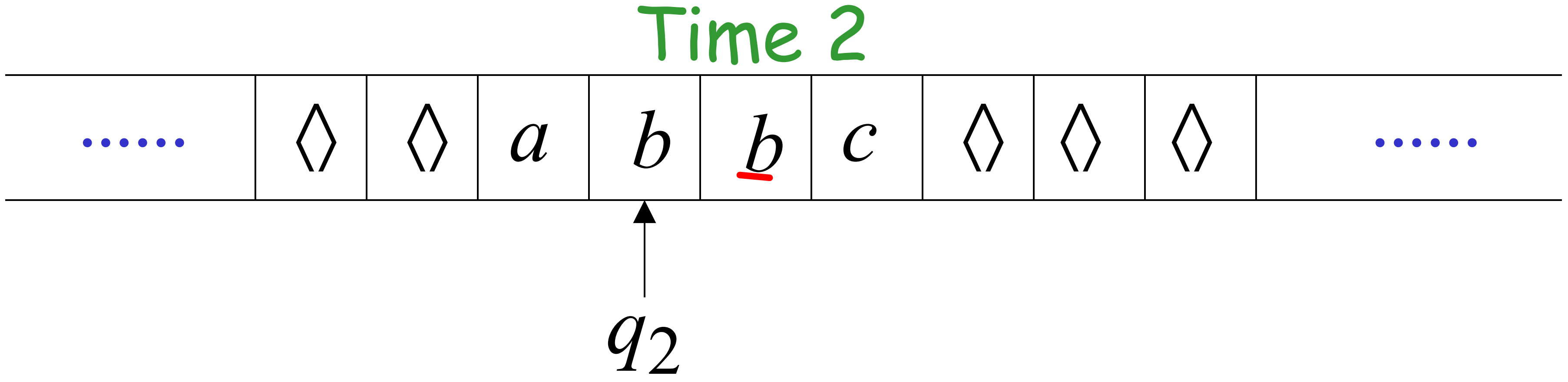
Time 2

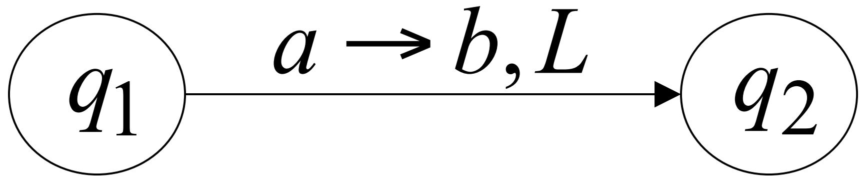


$$\begin{tikzcd} q_1 \arrow[r, "a \rightarrow b{,} R"] & q_2 \end{tikzcd}$$

Example:

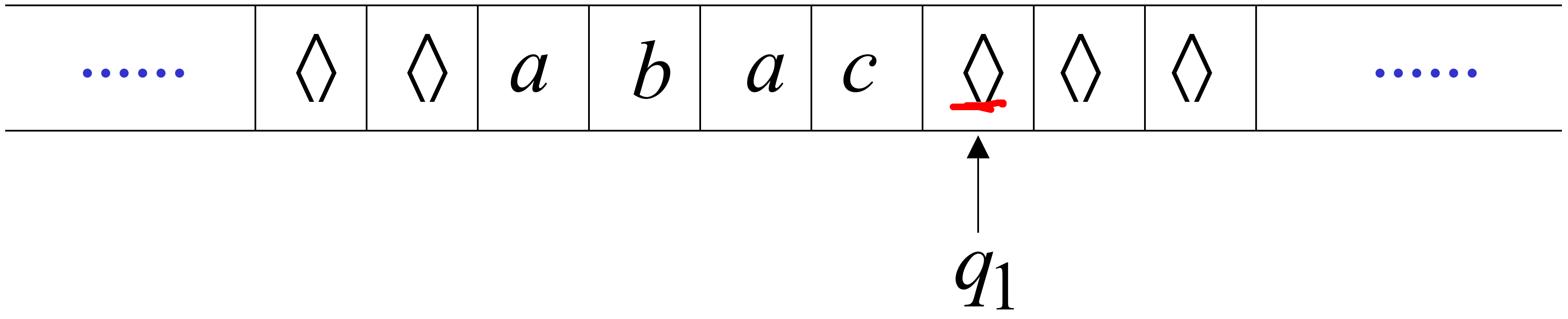




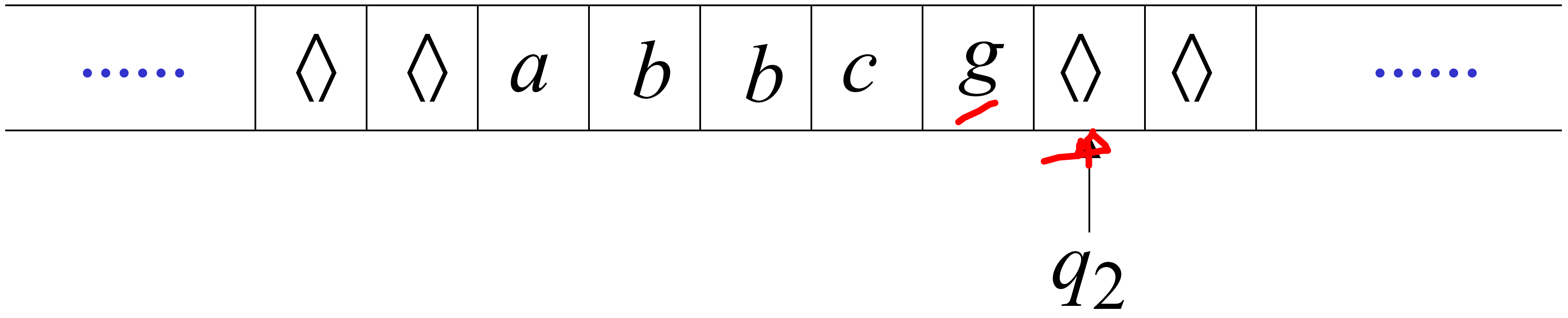


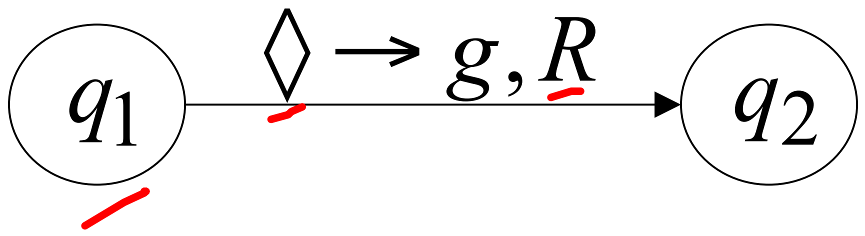
Example:

Time 1



Time 2

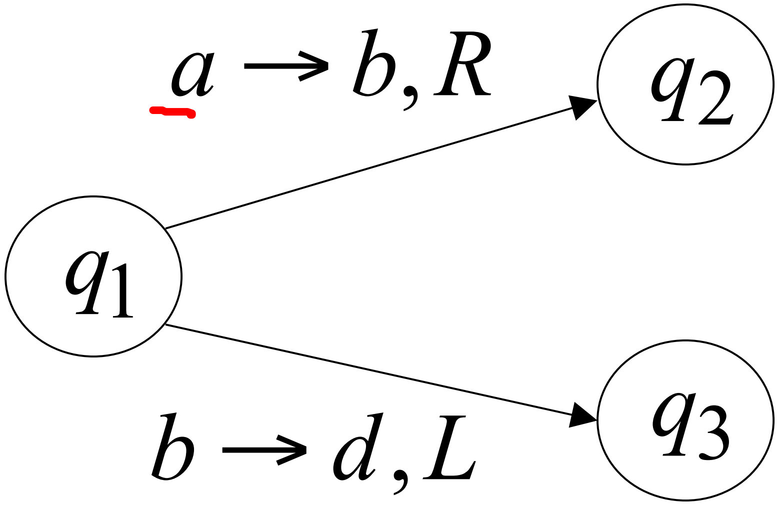




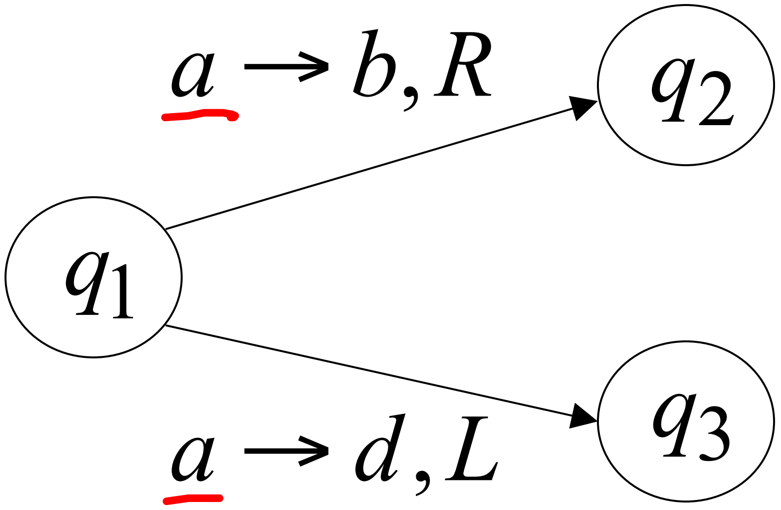
Determinism

Turing Machines are deterministic

Allowed



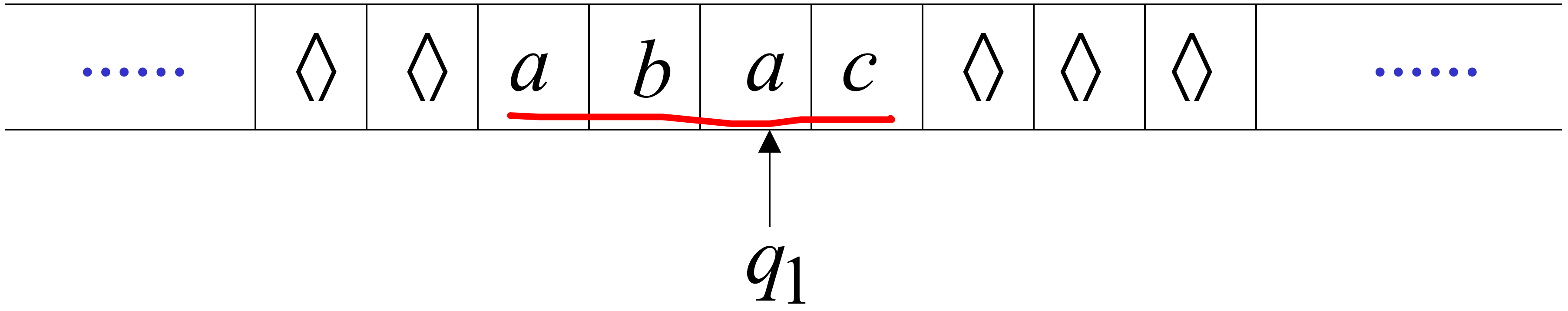
Not Allowed

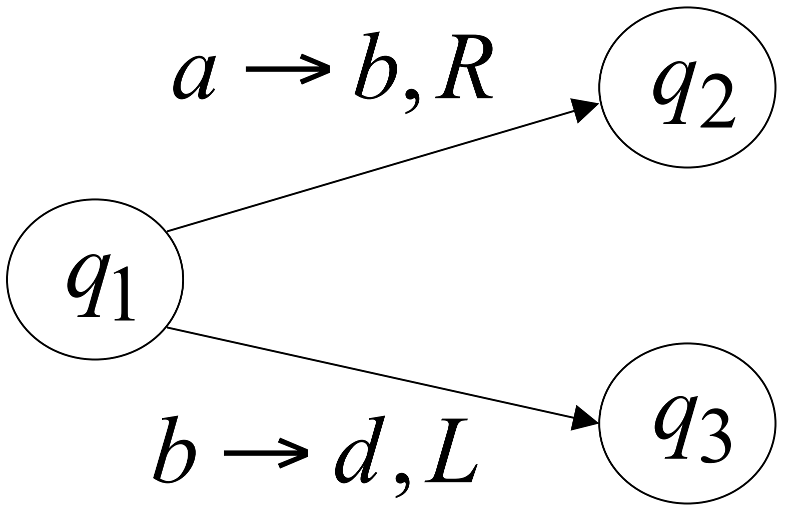


No lambda transitions allowed

Partial Transition Function

Example:





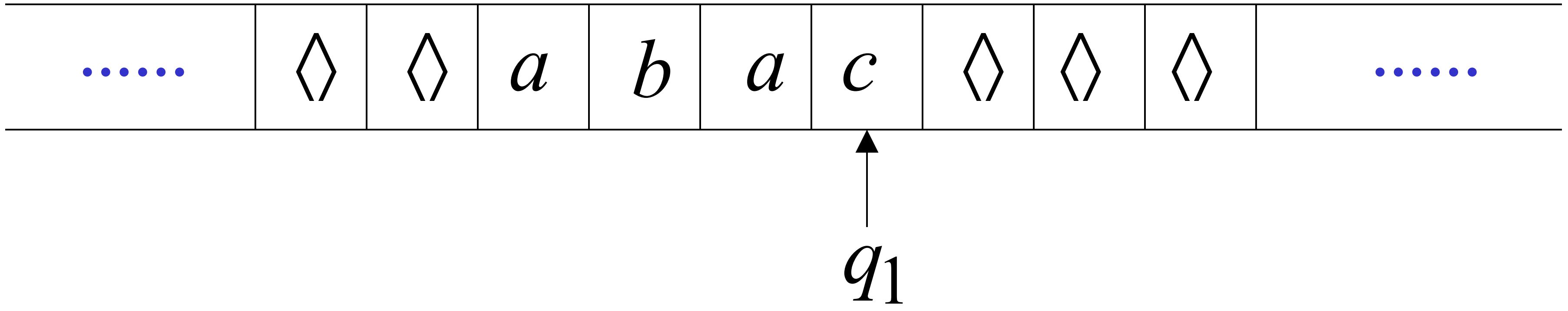
Allowed:

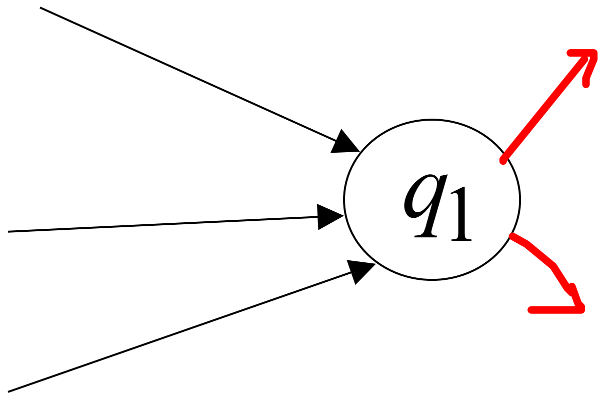
No transition for input symbol c

Halting

The machine ***halts*** in a state if there is no transition to follow

Halting Example 1:

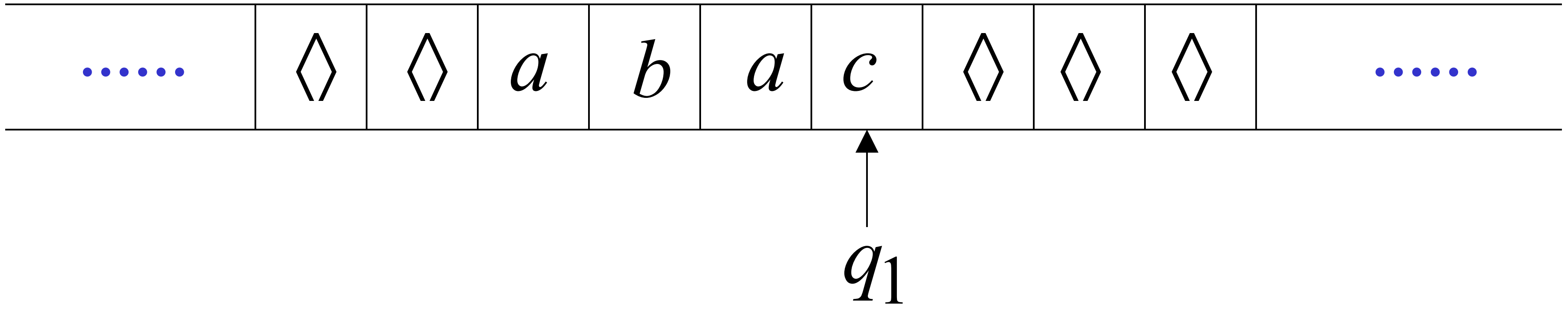


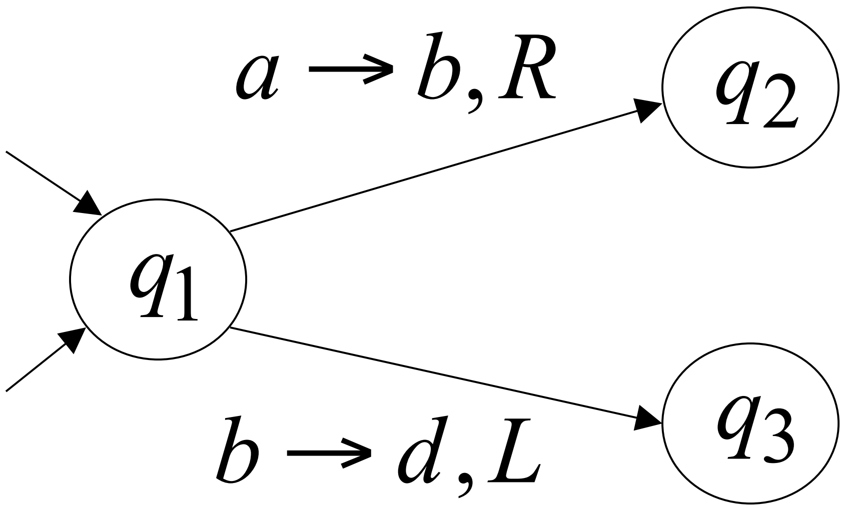


No transition from q_1

HALT!!!

Halting Example 2:

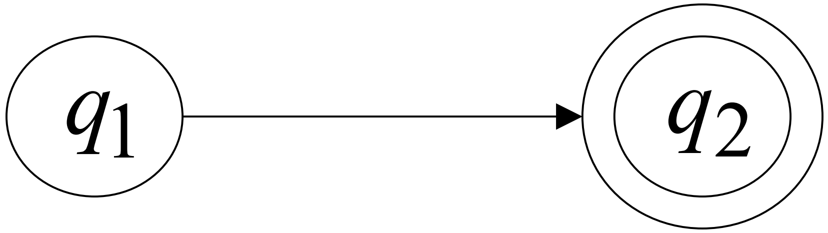




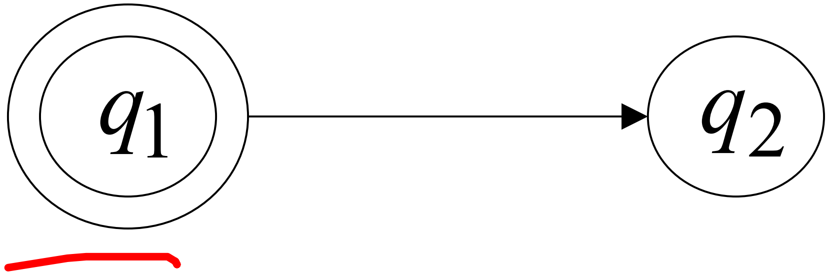
No possible transition from q_1 and symbol c

HALT!!!

Accepting States



Allowed



Not Allowed

- Accepting states have no outgoing transitions
- The machine halts and accepts

Acceptance

Accept Input string $\Rightarrow$ If machine halts in an accept state

Reject Input string $\Rightarrow$ If machine halts in a non-accept state

or

If machine enters an *infinite loop*

Observation:

In order to accept an input string, it is not necessary to scan all the symbols in the string

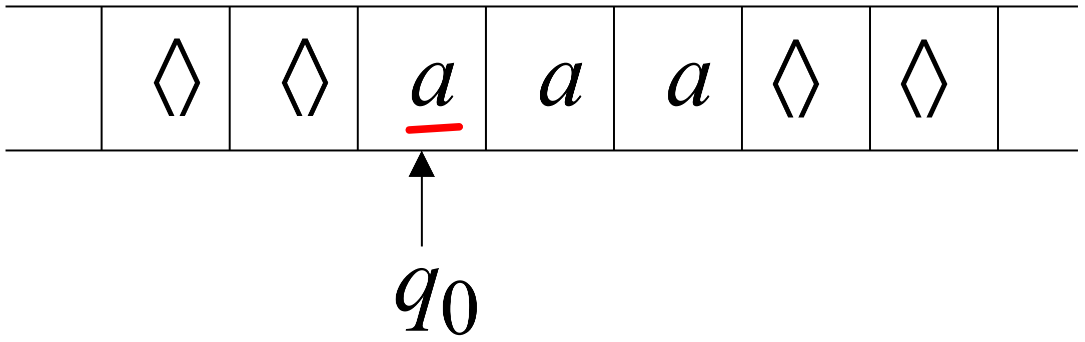
Turing Machine Example

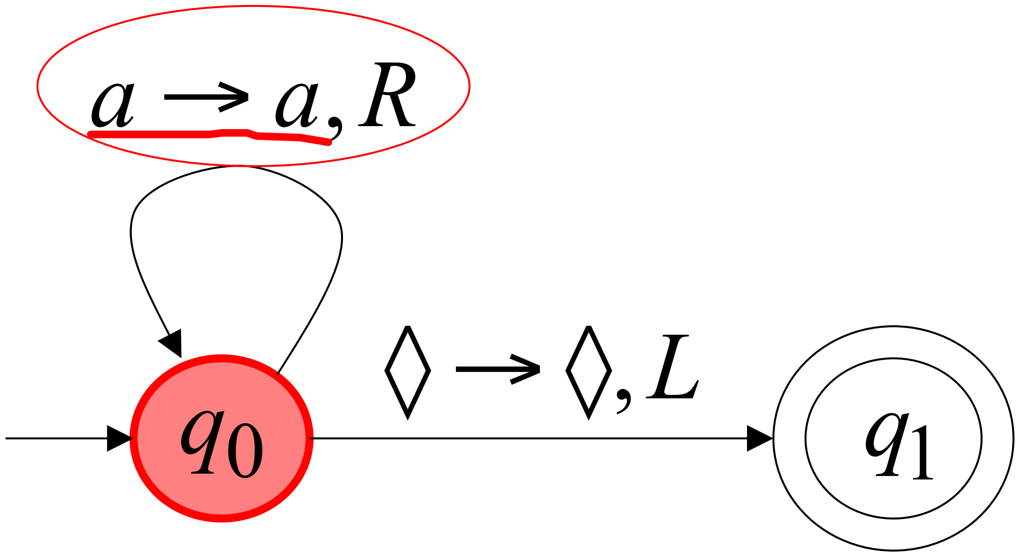
Input alphabet $\Sigma = \{\underline{a, b}\}$

Accepts the language: $\underline{a^*}$

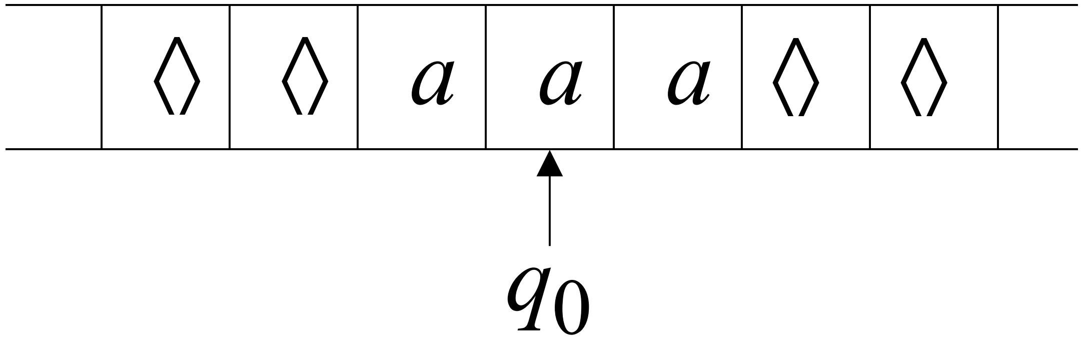
$$\begin{tikzcd} {} \arrow[r] & q_0 \arrow[loop above, "\underline{a} \rightarrow \underline{a}, \underline{R}"] \arrow[rr, "\underline{\Diamond} \rightarrow \underline{\Diamond}, \underline{L}"] & & \underline{q_1} \end{tikzcd}$$

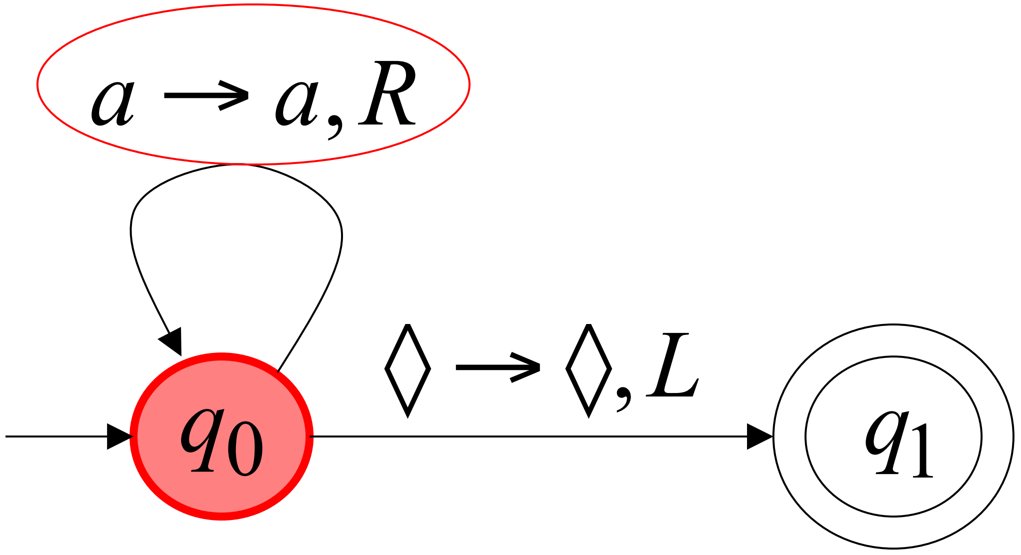
Time 0



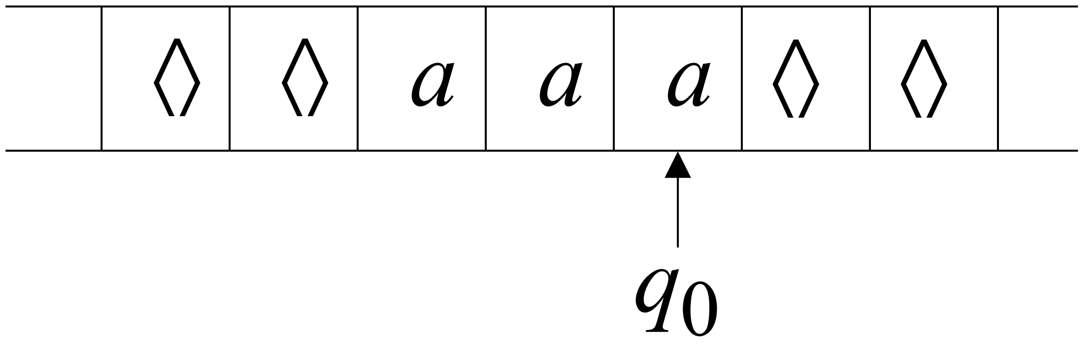


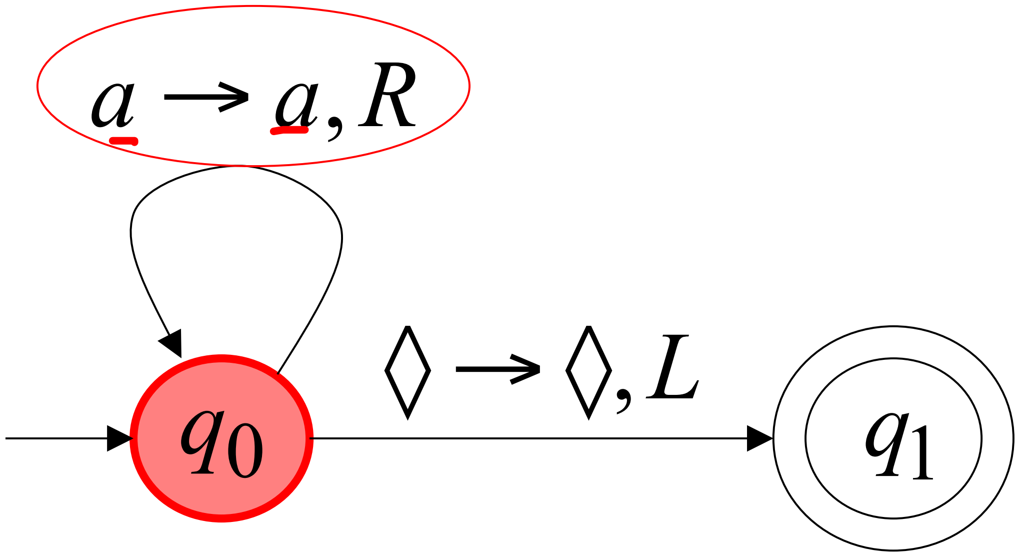
Time 1



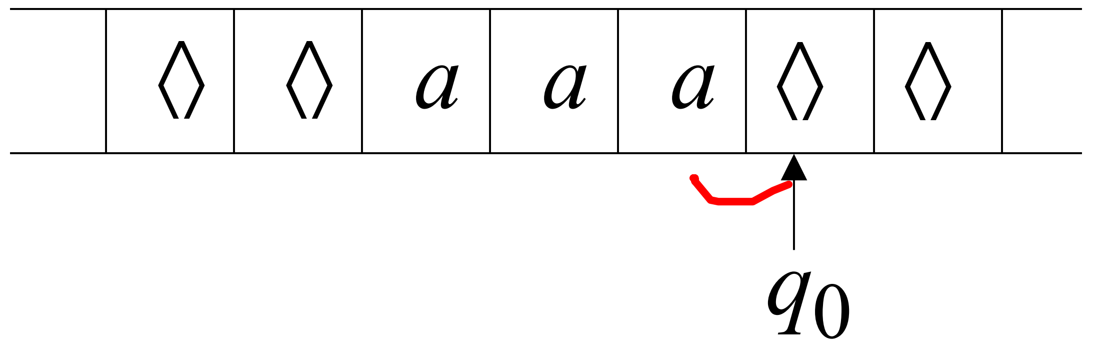


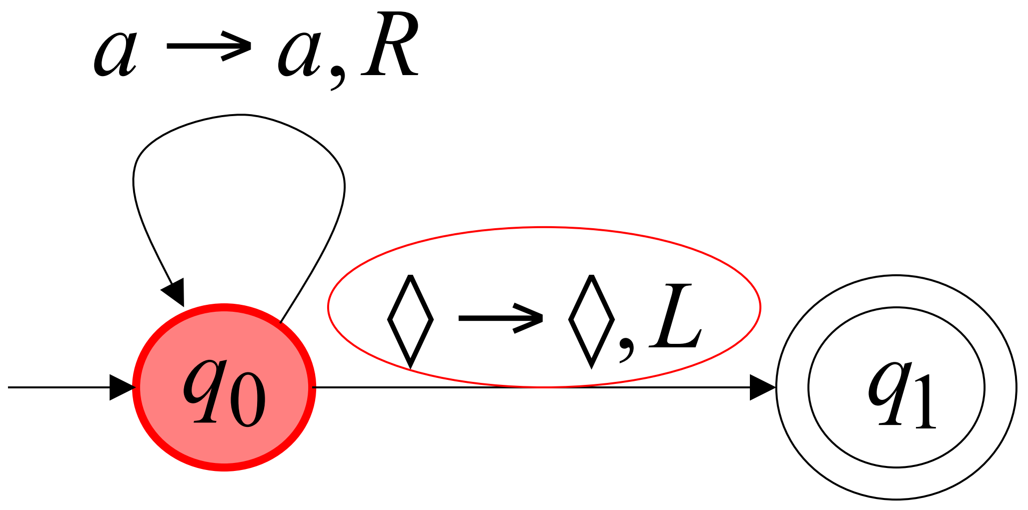
Time 2



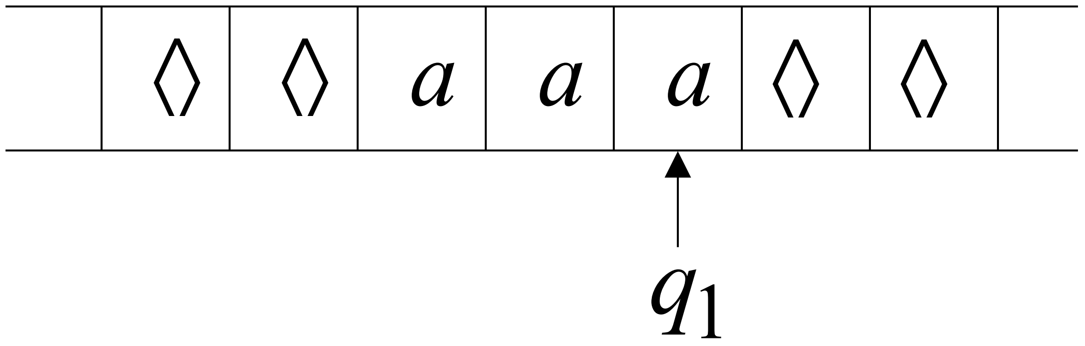


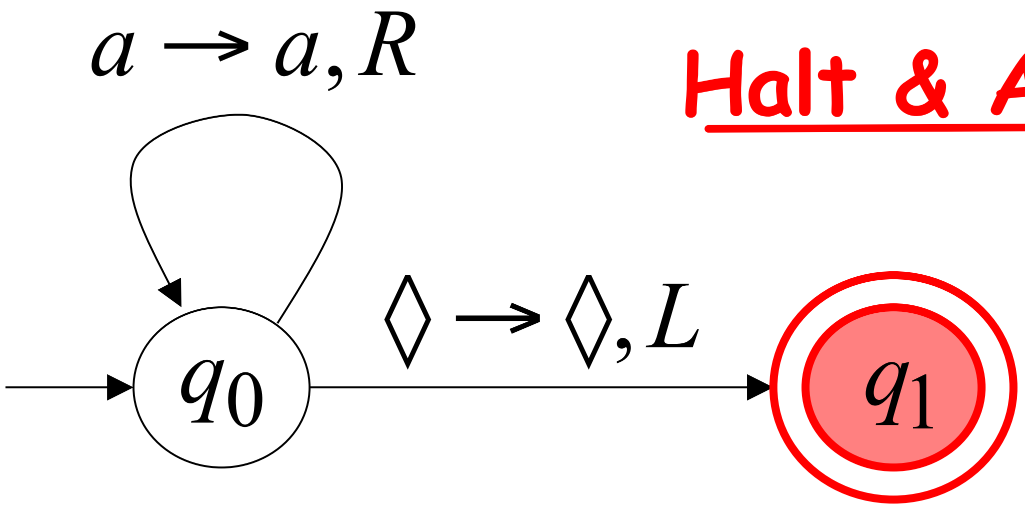
Time 3





Time 4

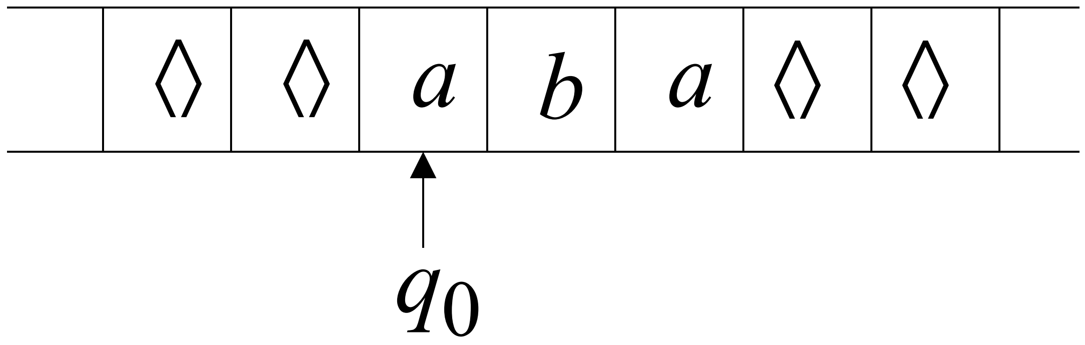


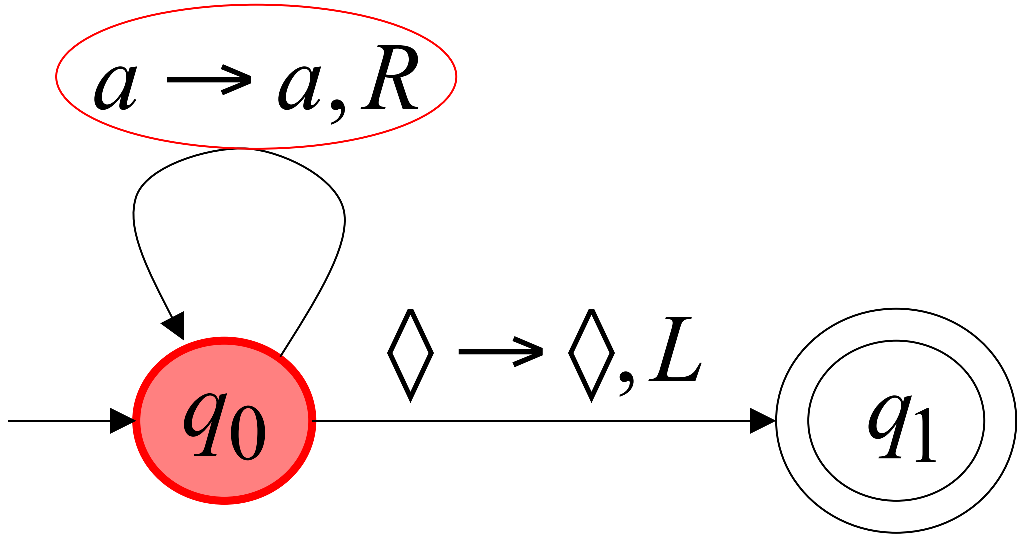


Halt & Accept

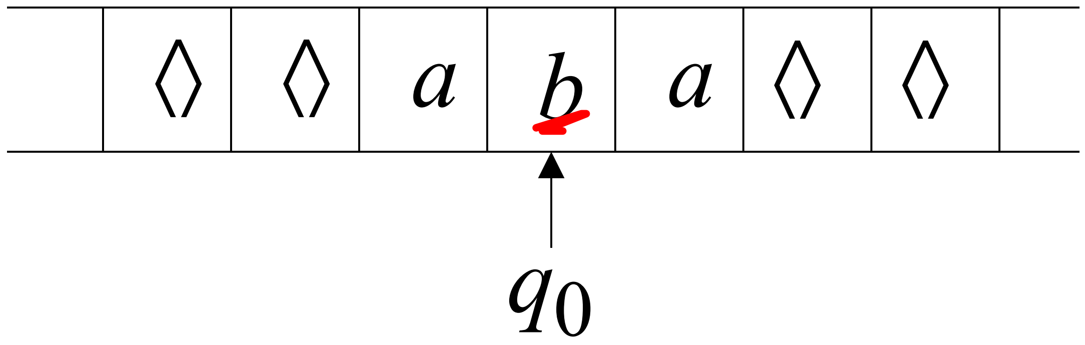
Rejection Example

Time 0



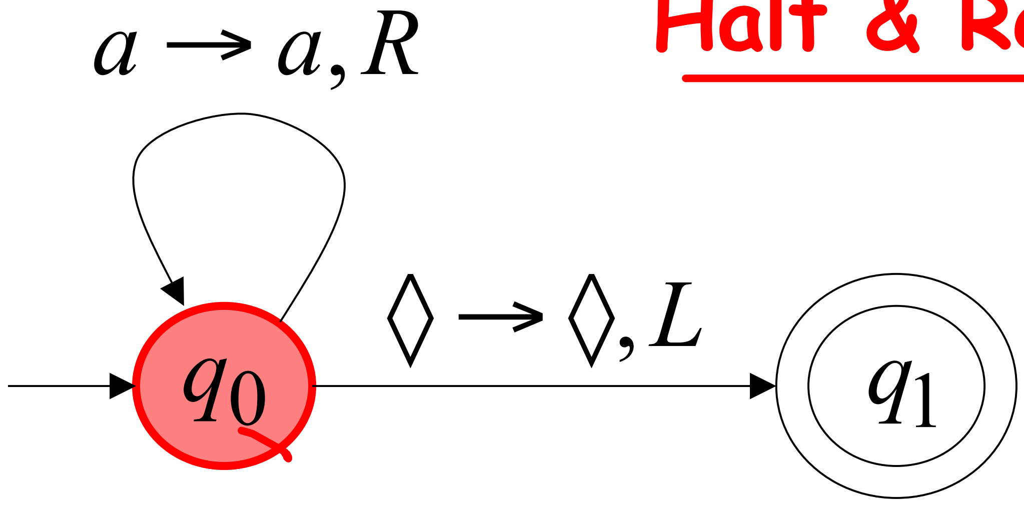


Time 1



No possible Transition

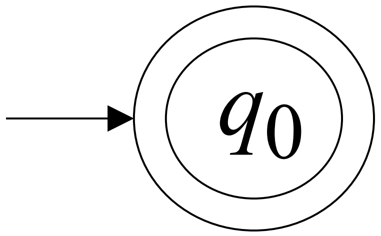
Halt & Reject



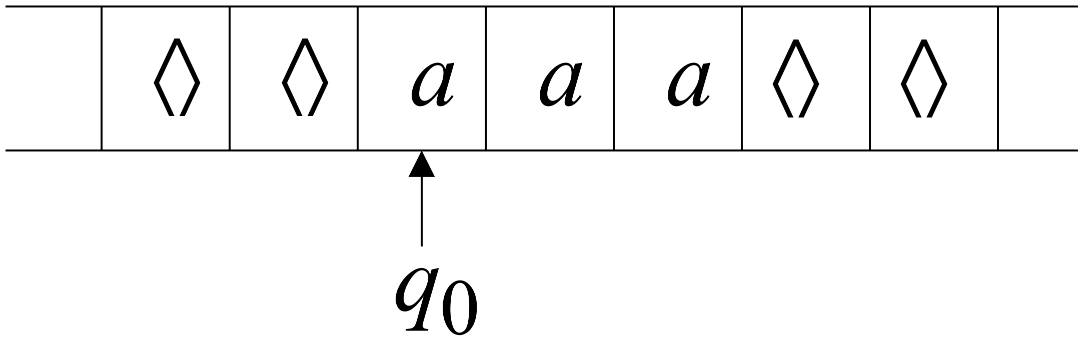
A simpler machine for same language

but for input alphabet $\Sigma = \{a\}$

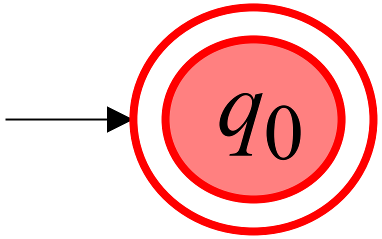
Accepts the language: a^* λ



Time 0



Halt & Accept

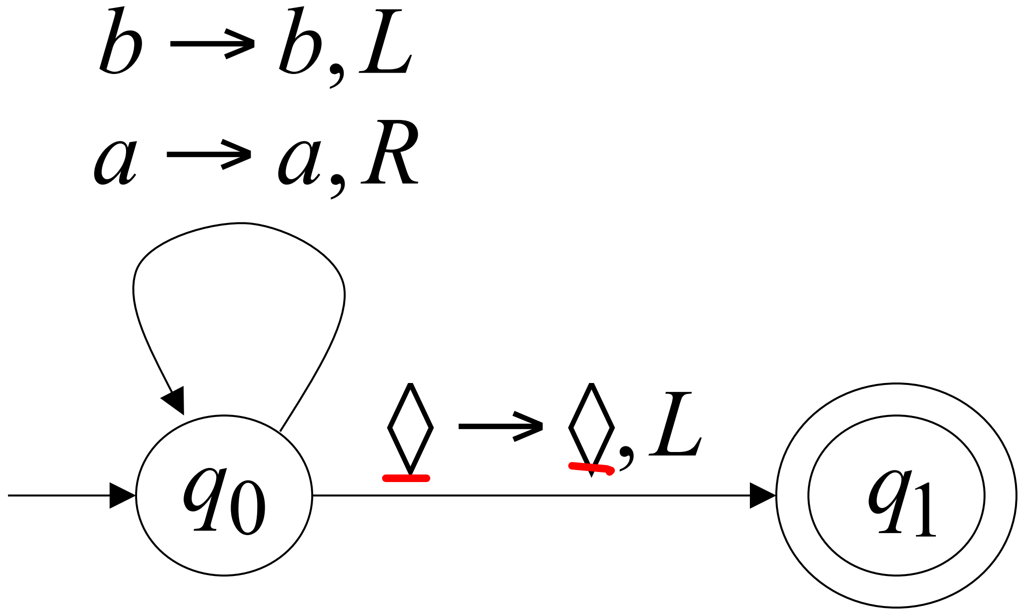


Not necessary to scan input

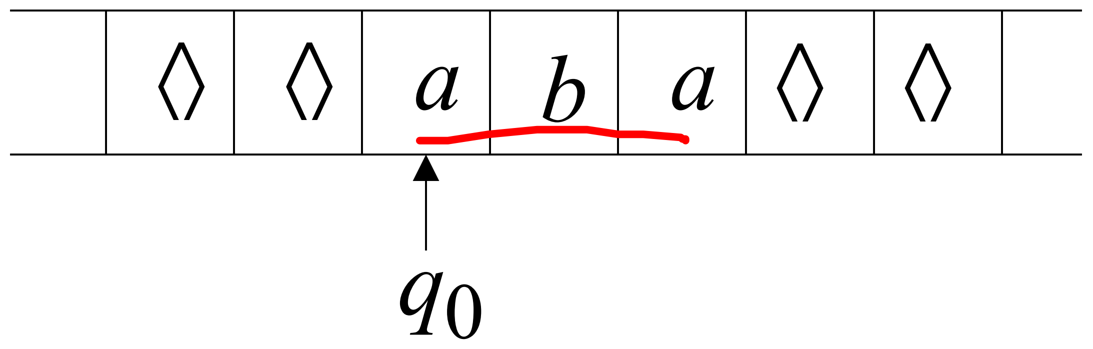
Infinite Loop Example

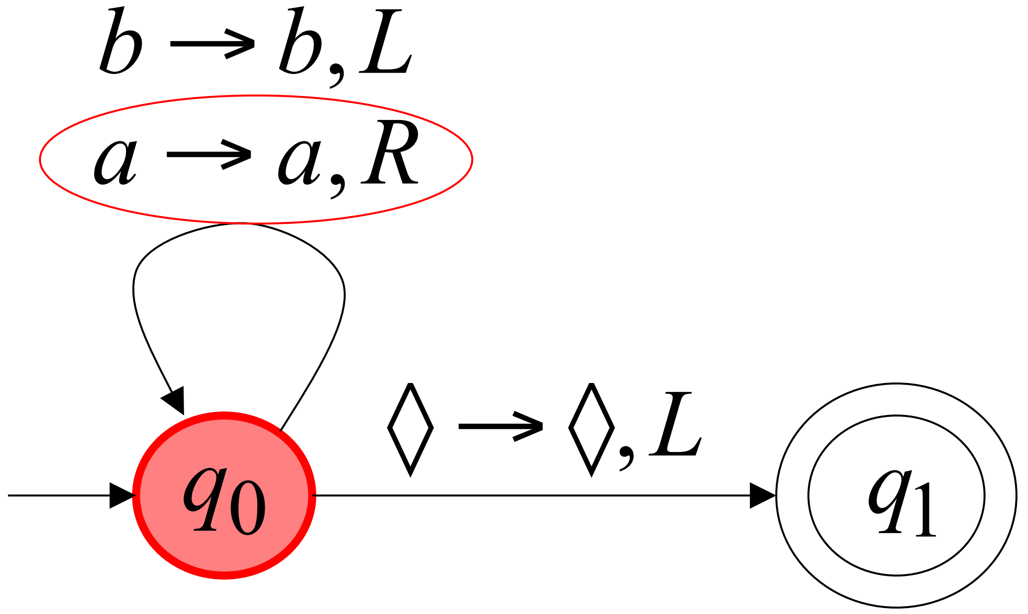
a

A Turing machine
for language $a^* + b(a+b)^*$

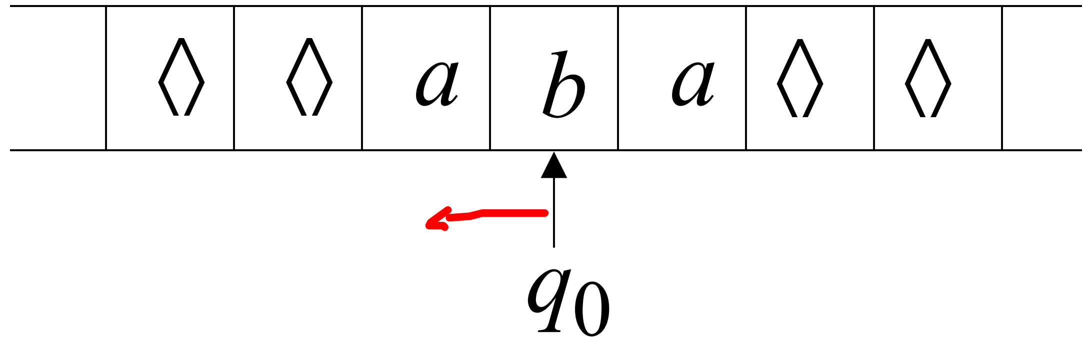


Time 0

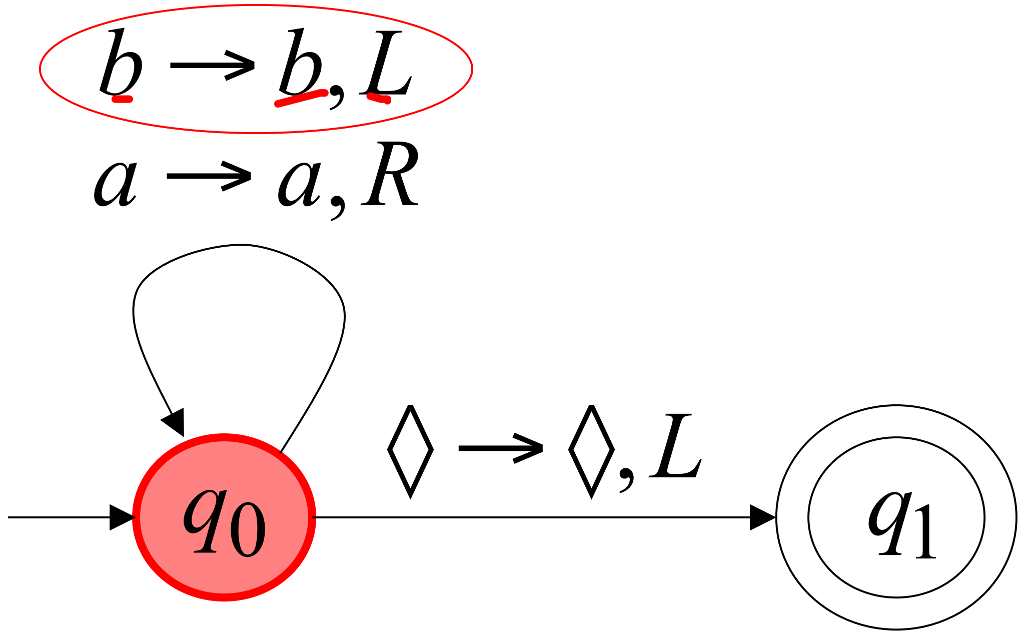




Time 1

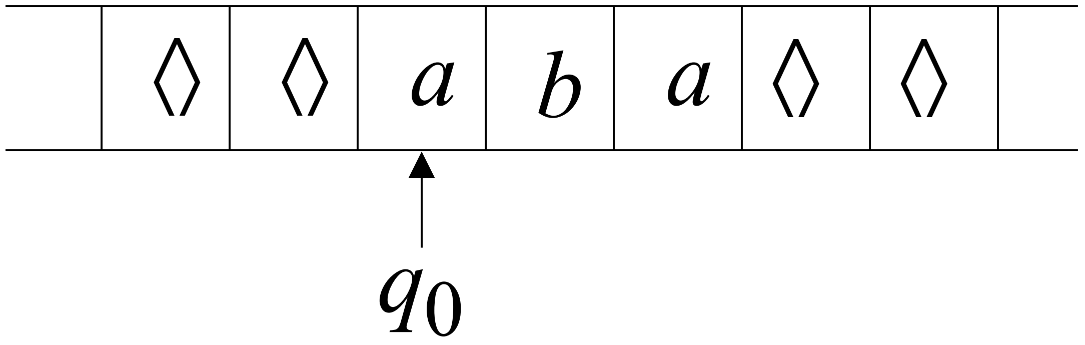


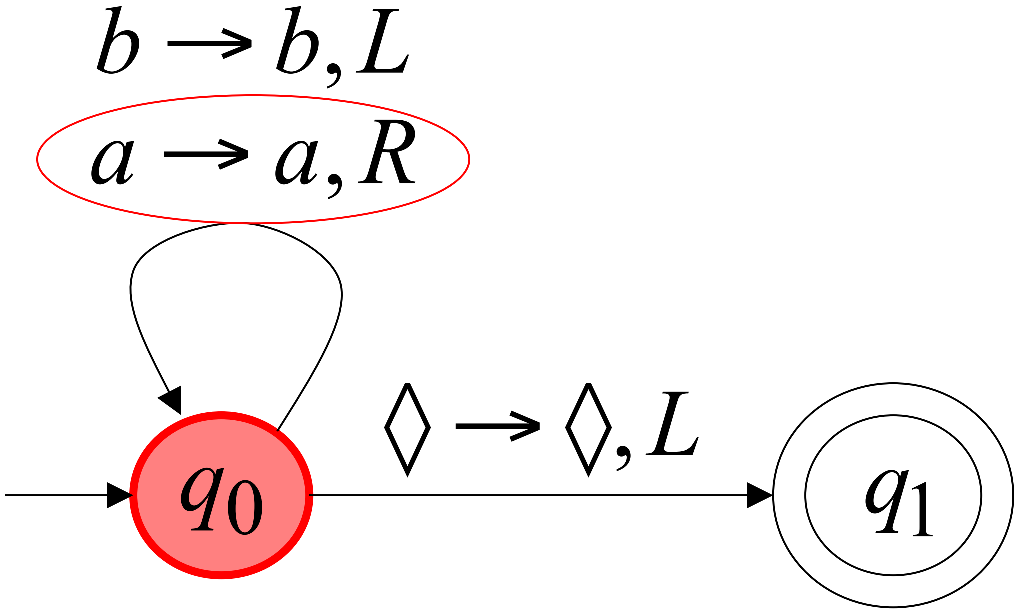
◊ ◊ a b a ◊ ◊
q_0



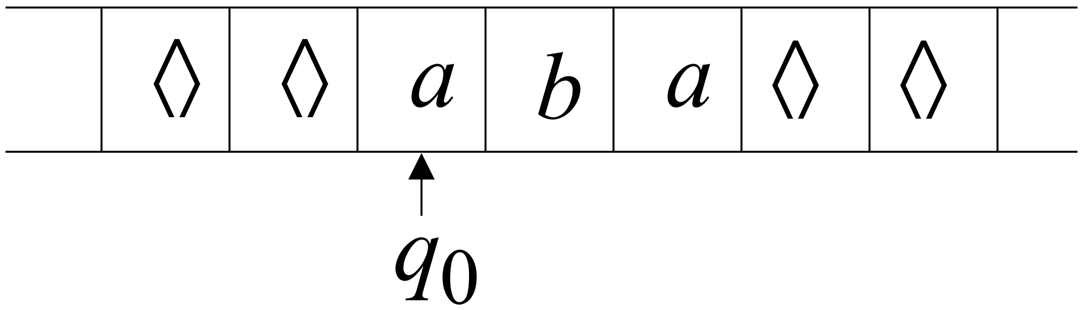
$b \rightarrow b, L$
$a \rightarrow a, R$
$\lozenge \rightarrow \lozenge, L$
q_0
q_1

Time 2

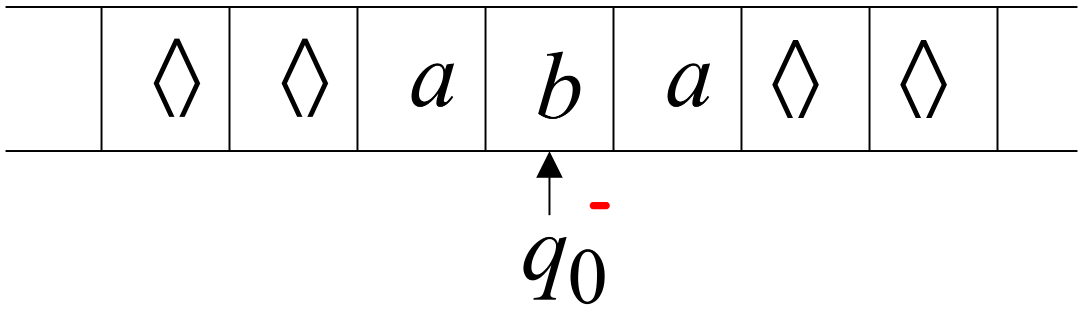




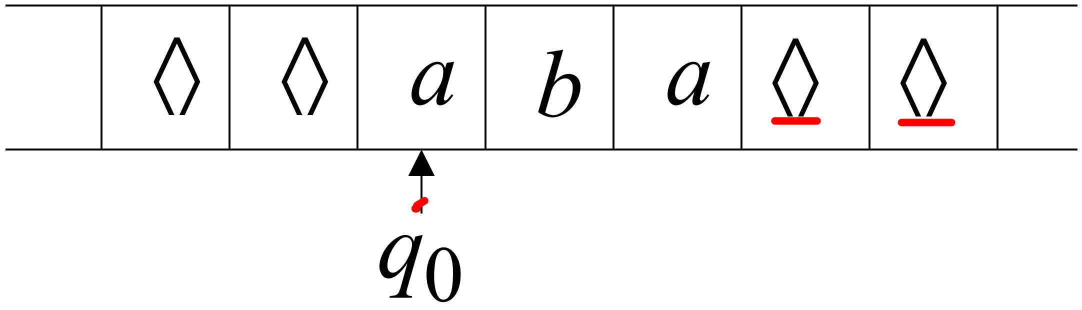
Time 2



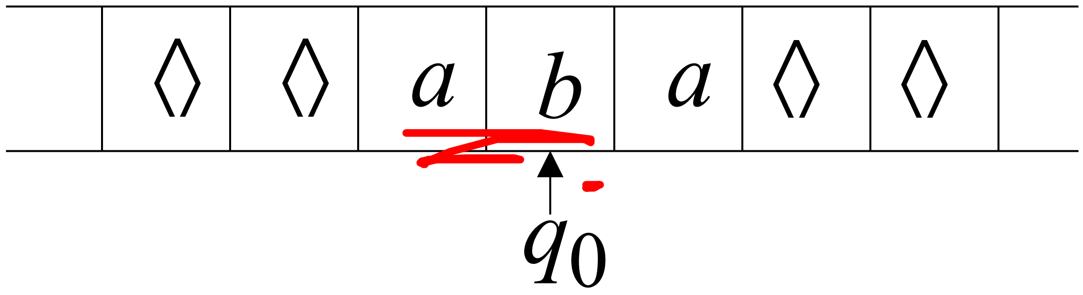
Time 3



Time 4



Time 5



Infinite loop

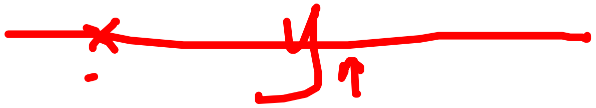
Because of the infinite loop:

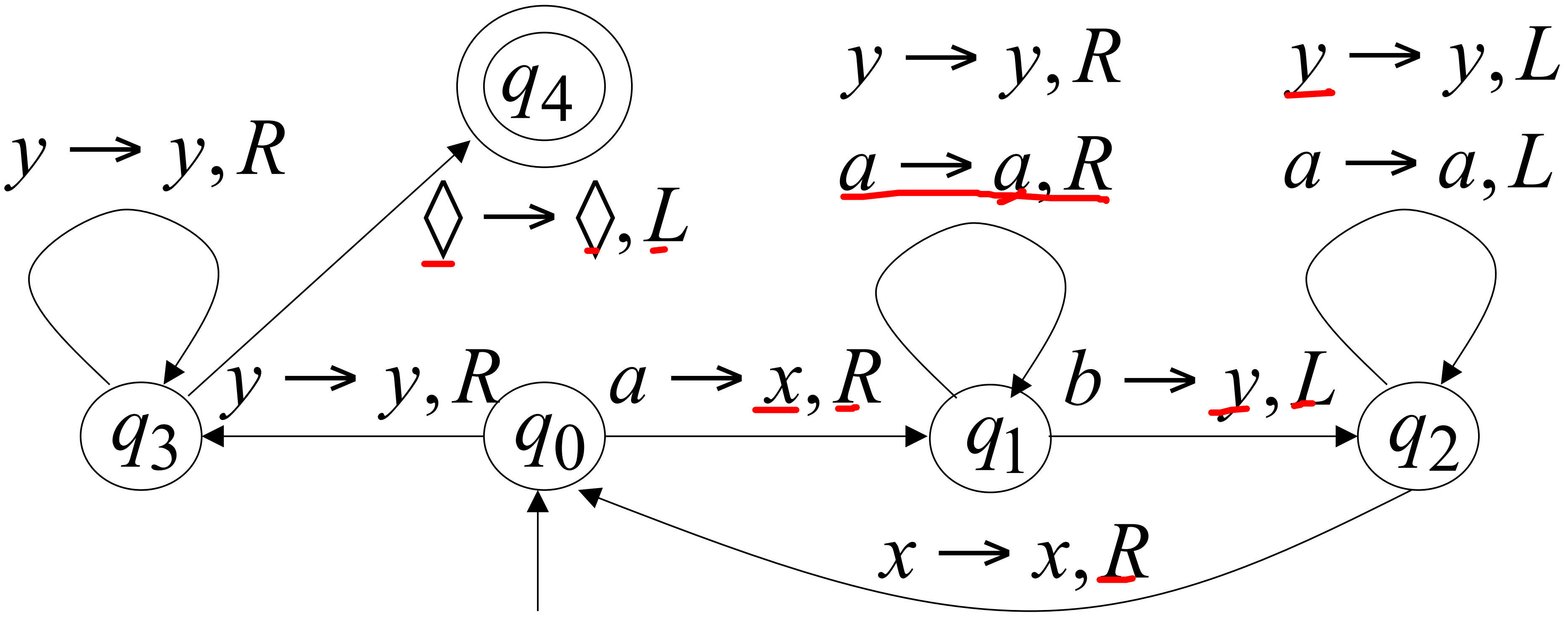
- The accepting state cannot be reached
- The machine never halts
- The input string is rejected

Another Turing Machine Example

Turing machine for the language $\{a^n b^n\}$

$n \geq 1$





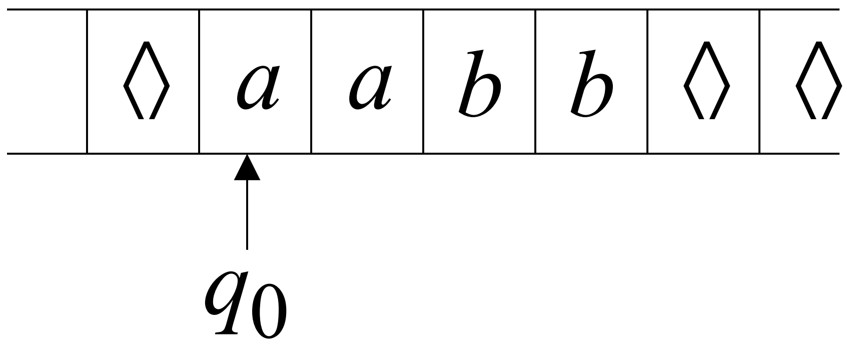
Basic Idea:

Match a's with b's:

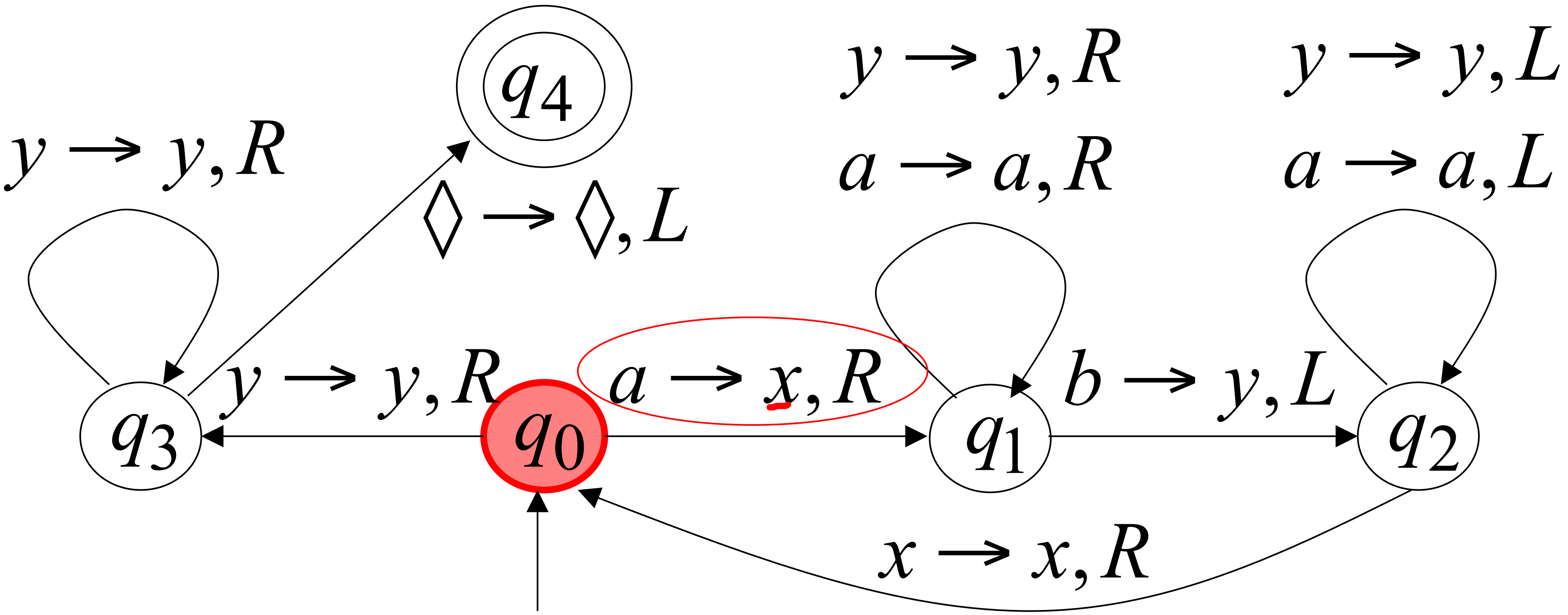
```
Repeat:
    replace leftmost a with x
    find leftmost b and replace it with y
Until there are no more a's or b's
```

If there is a remaining a or b reject

Time 0

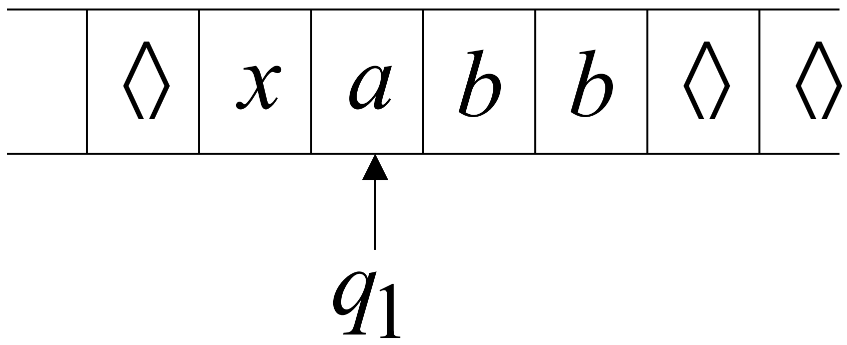


$\Diamond$ a a b b $\Diamond$ $\Diamond$
q_0

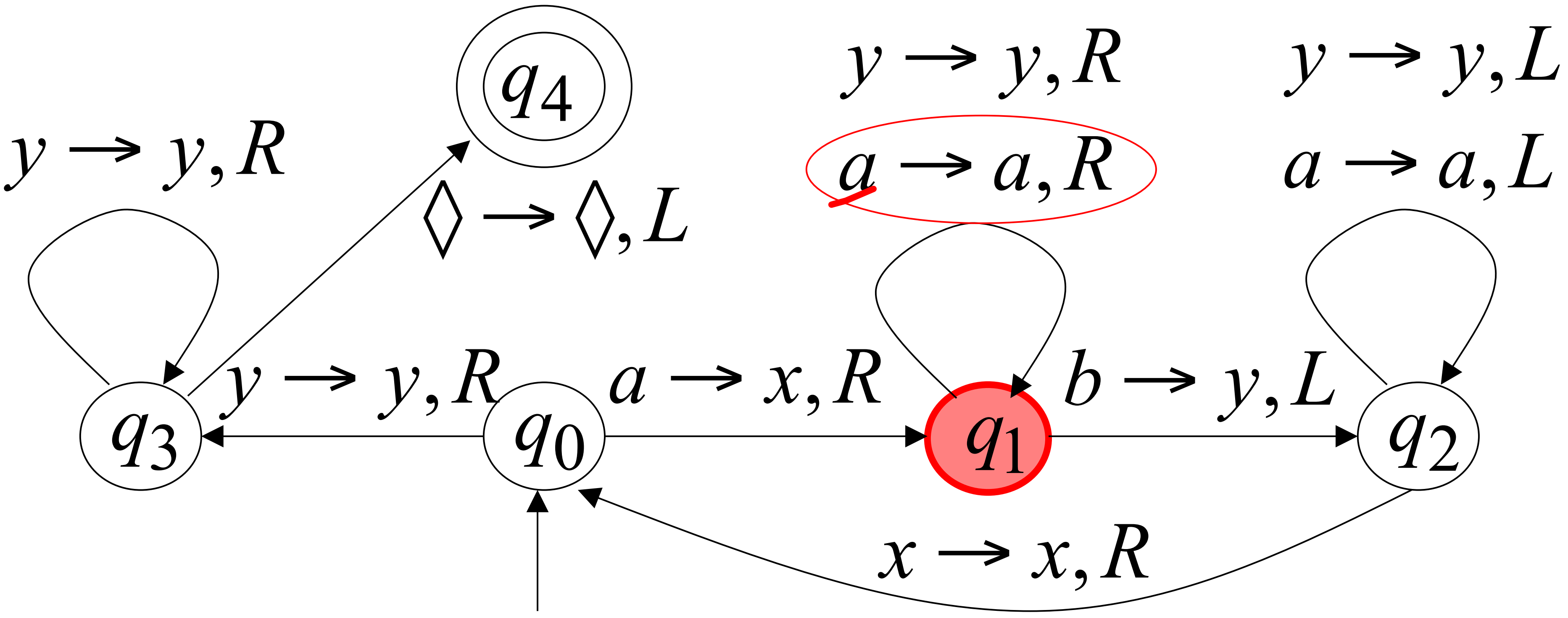


q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

Time 1

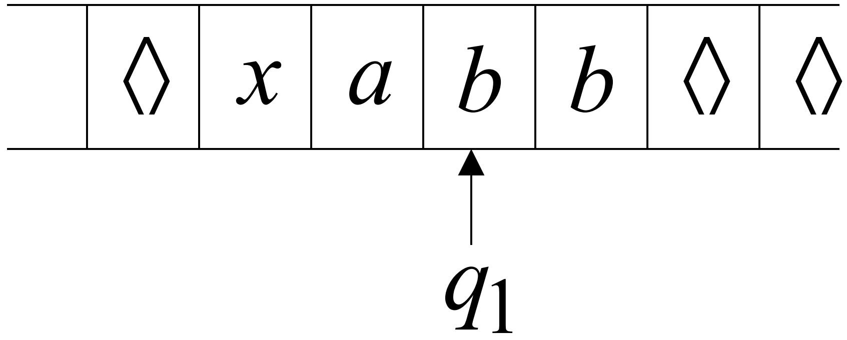


◊
x
a
b
b
◊
◊
q_1

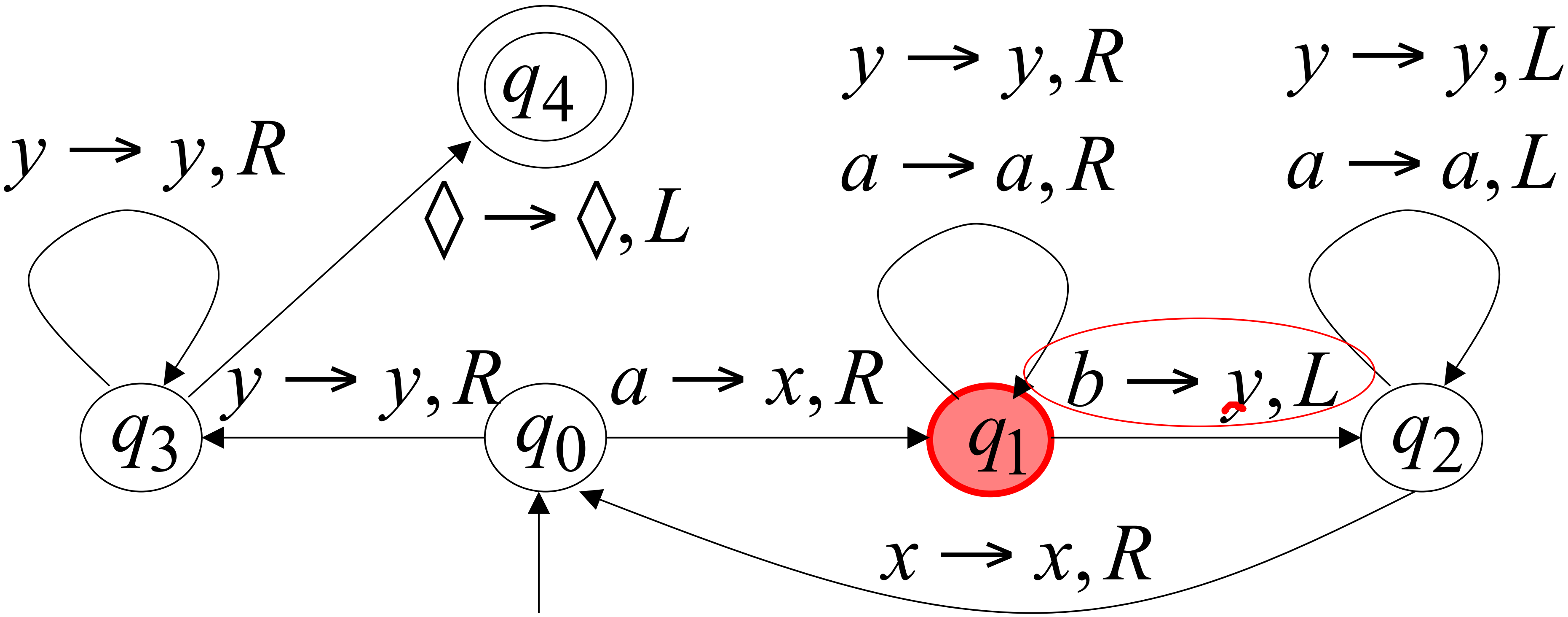


q_4
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$

Time 2

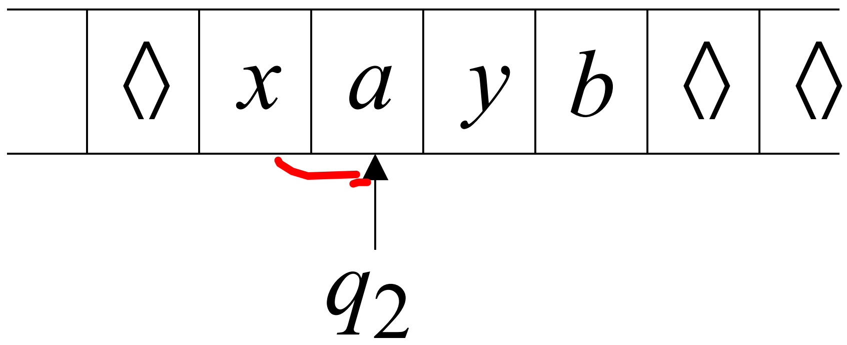


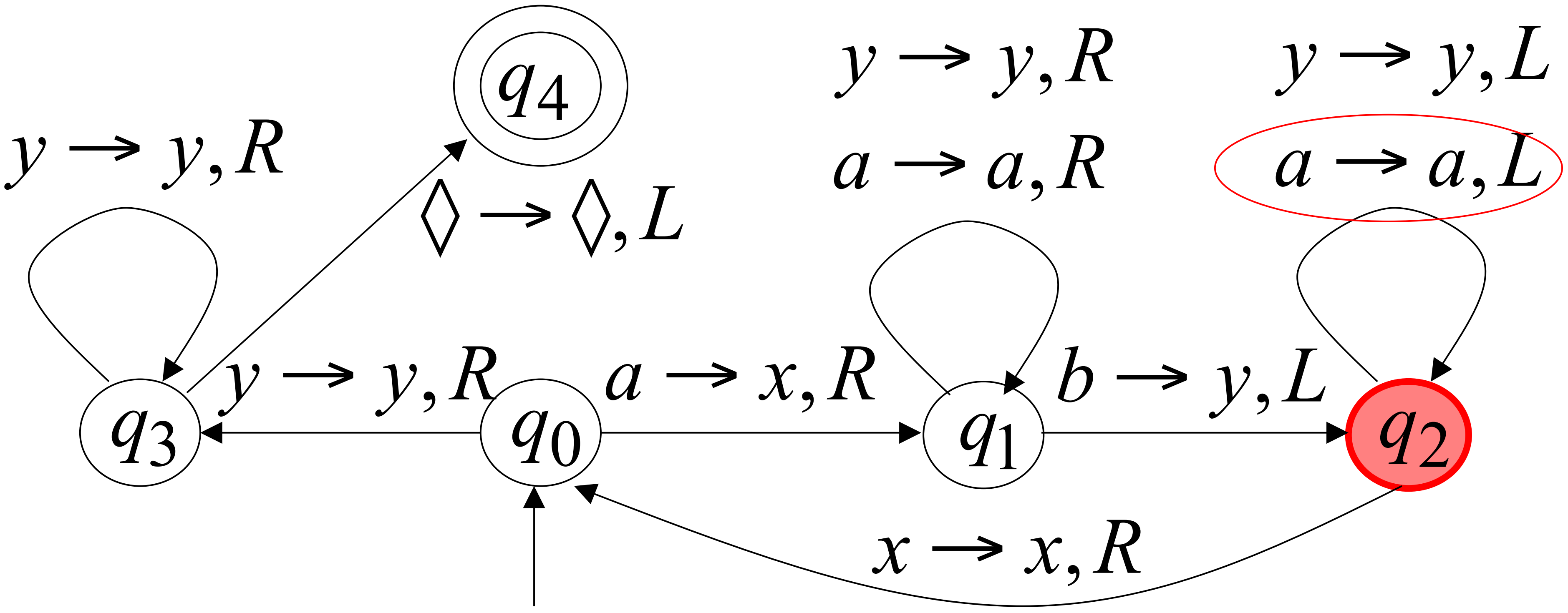
◊
x
a
b
b
◊
◊
q_1



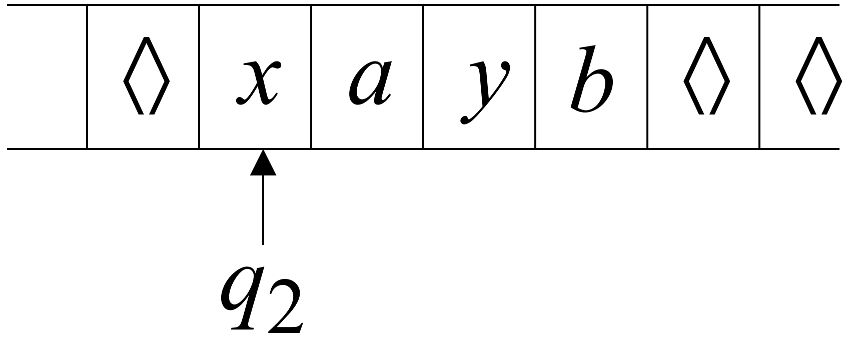
q_0
q_1
q_2
q_3
q_4
$a \rightarrow x, R$
$b \rightarrow y, L$
$x \rightarrow x, R$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$y \rightarrow y, R$
$◊ \rightarrow ◊, L$

Time 3

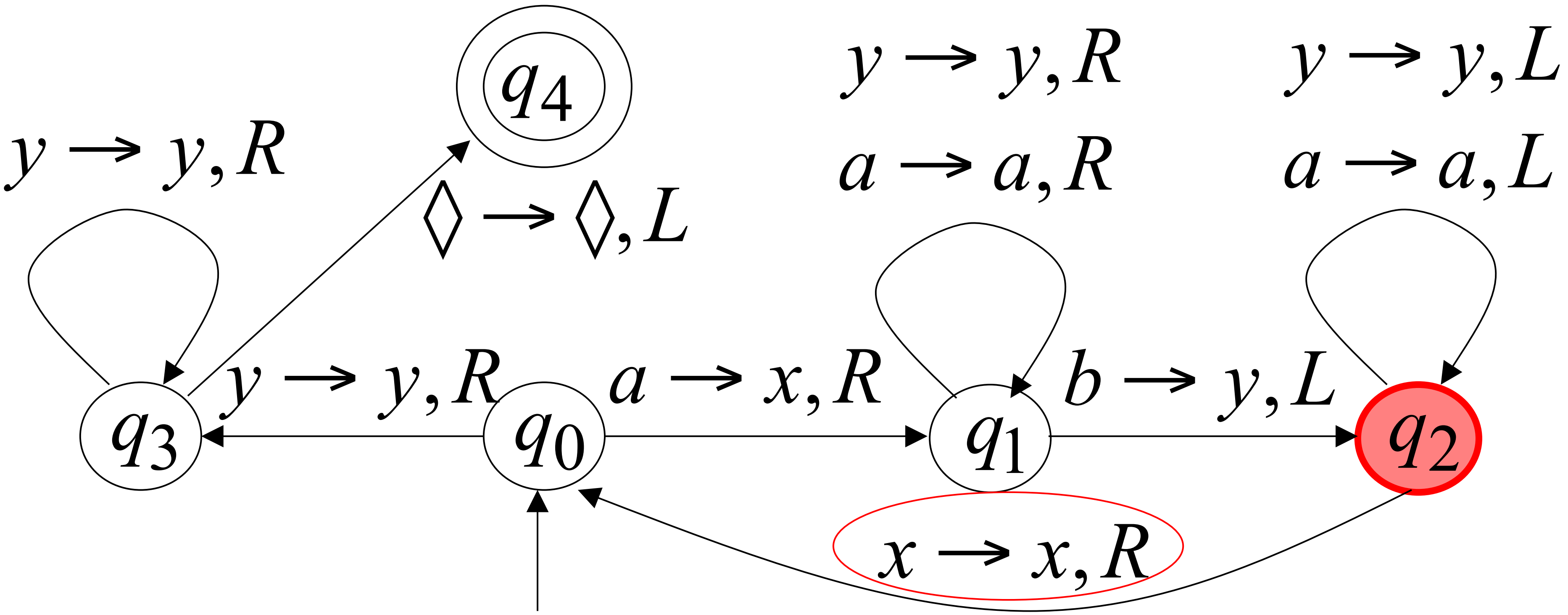

◊
x
a
y
b
◊
◊
q_2


q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
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 4

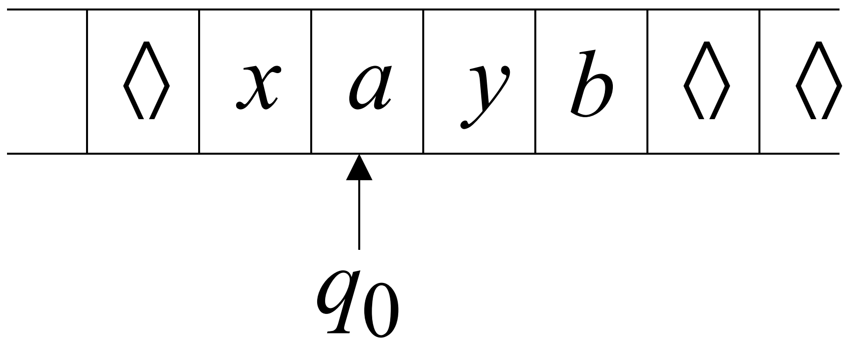


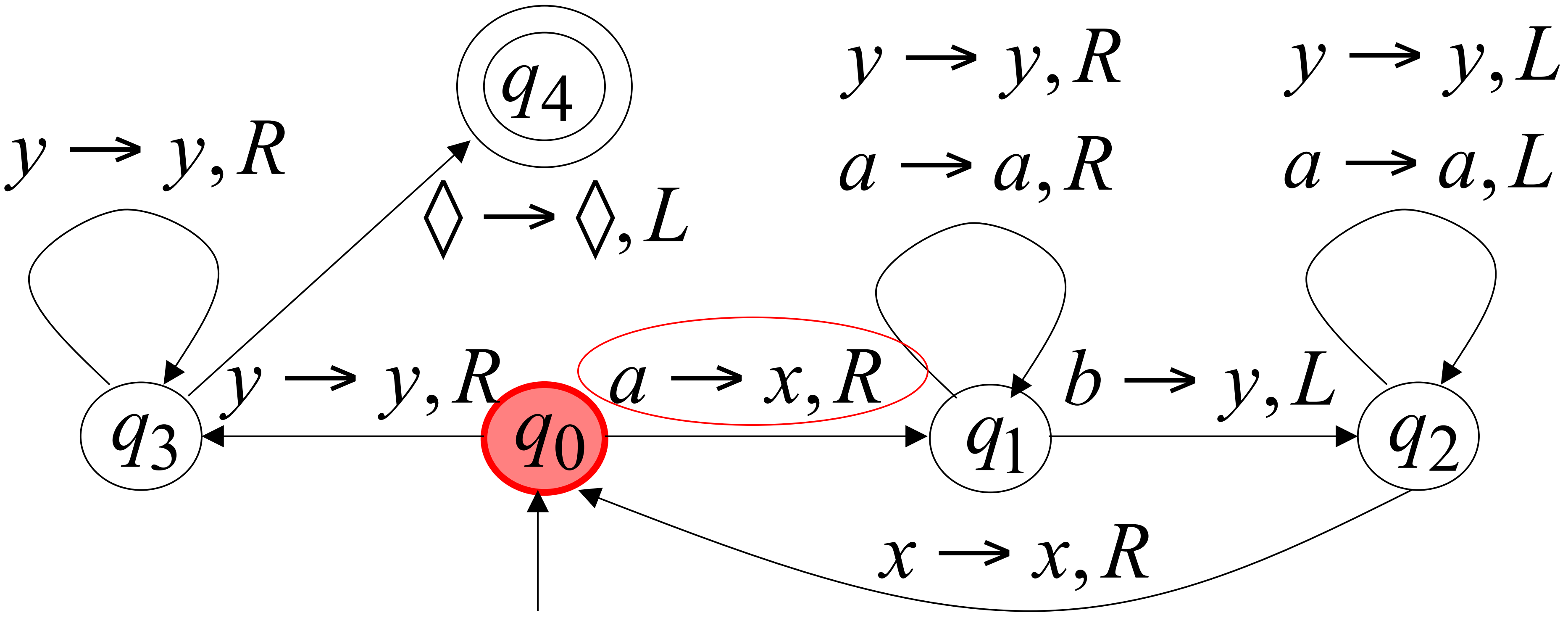
◊
x
a
y
b
◊
◊
q_2



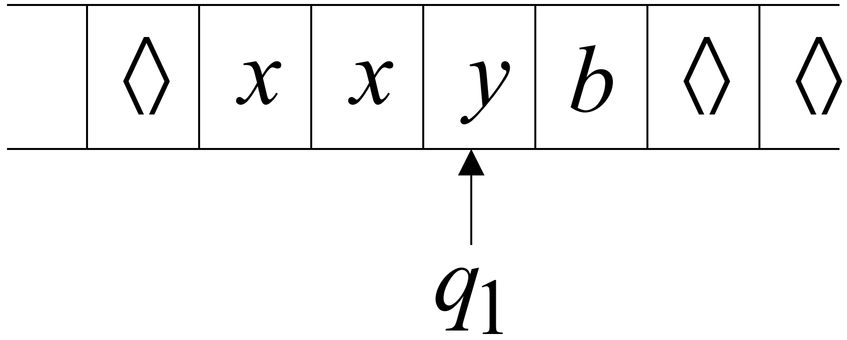
q_4
q_3
q_0
q_1
q_2
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
$x \rightarrow x, R$

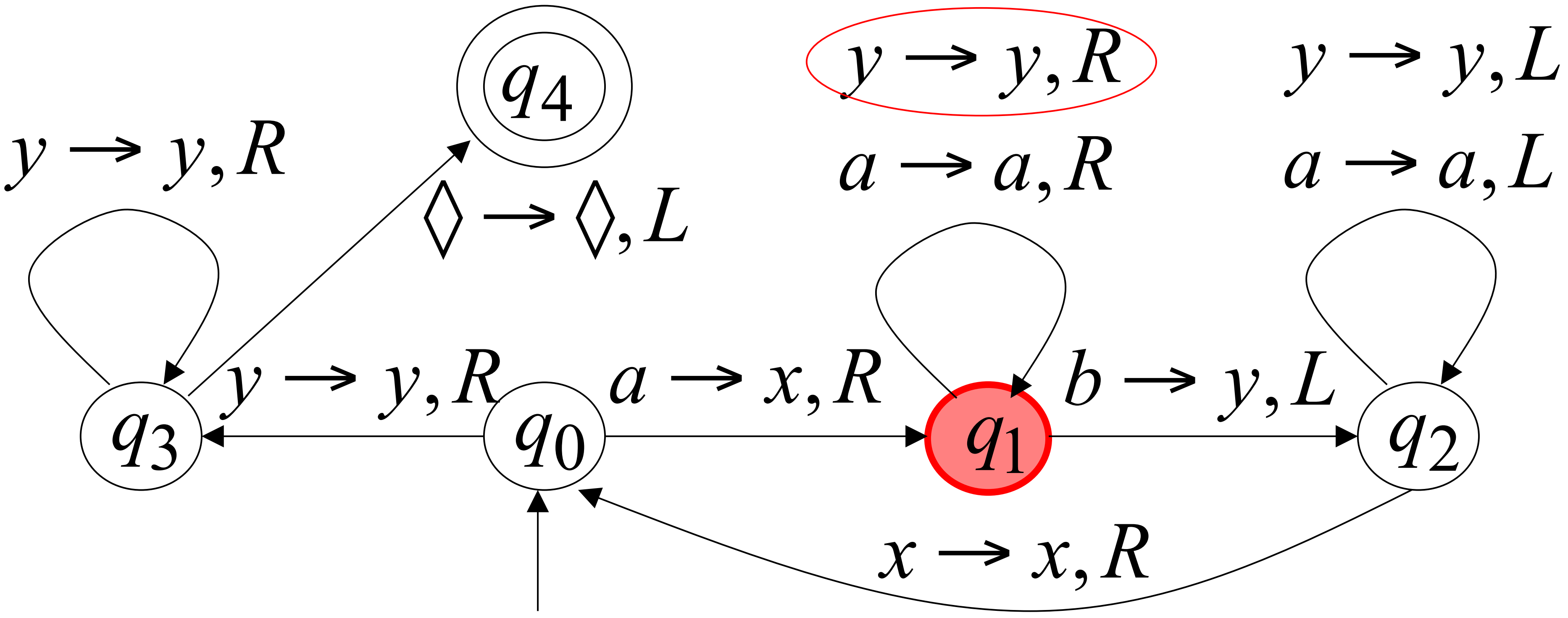
Time 5


◊
x
a
y
b
◊
◊
q_0

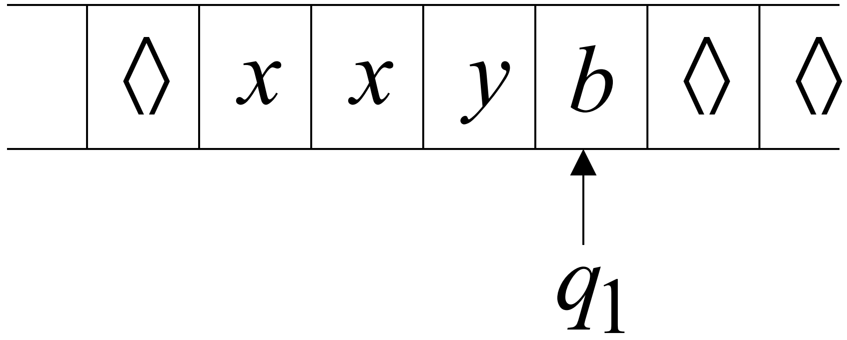

q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

Time 6

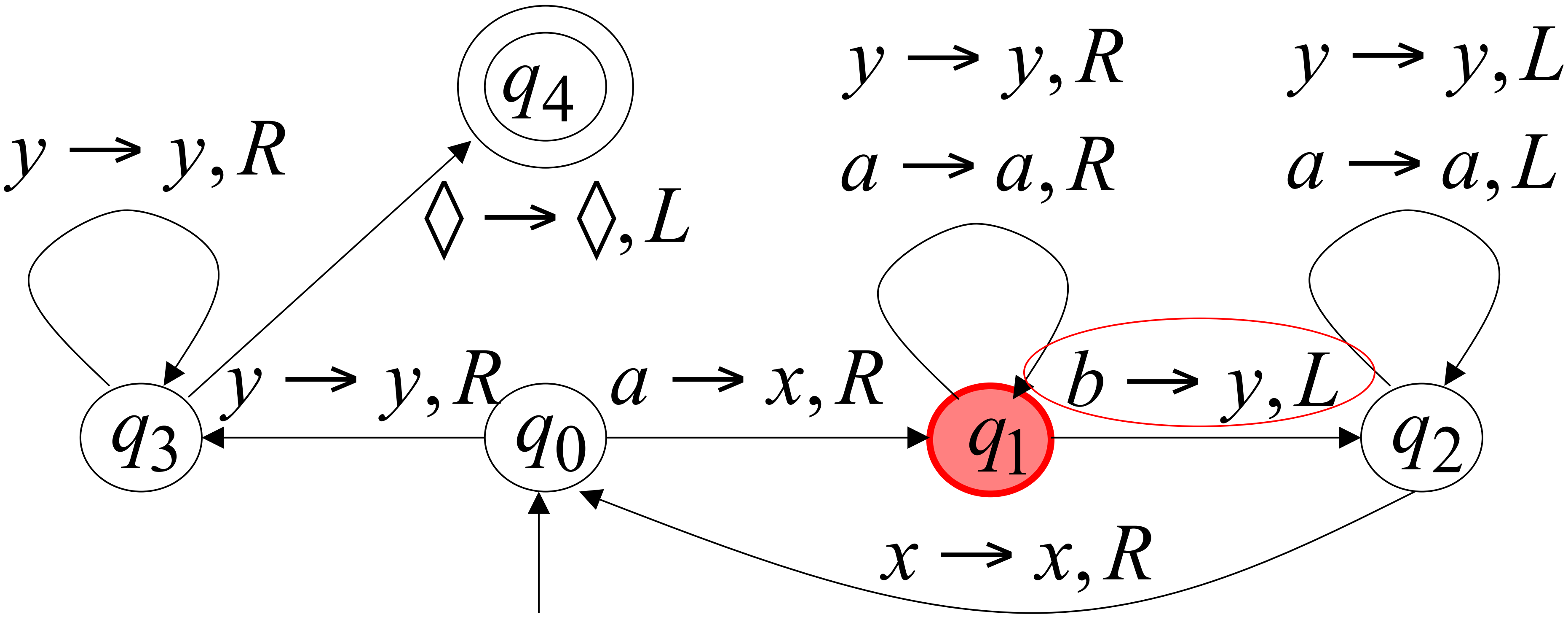

◊
x
x
y
b
◊
◊
q_1


q_4
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
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 7

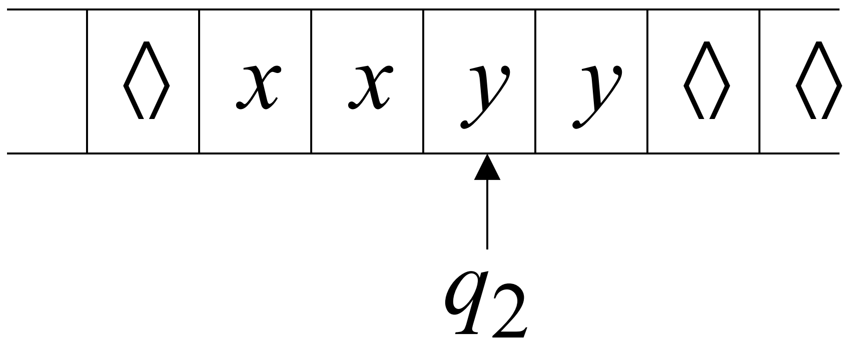


◊
x
x
y
b
◊
◊
q_1

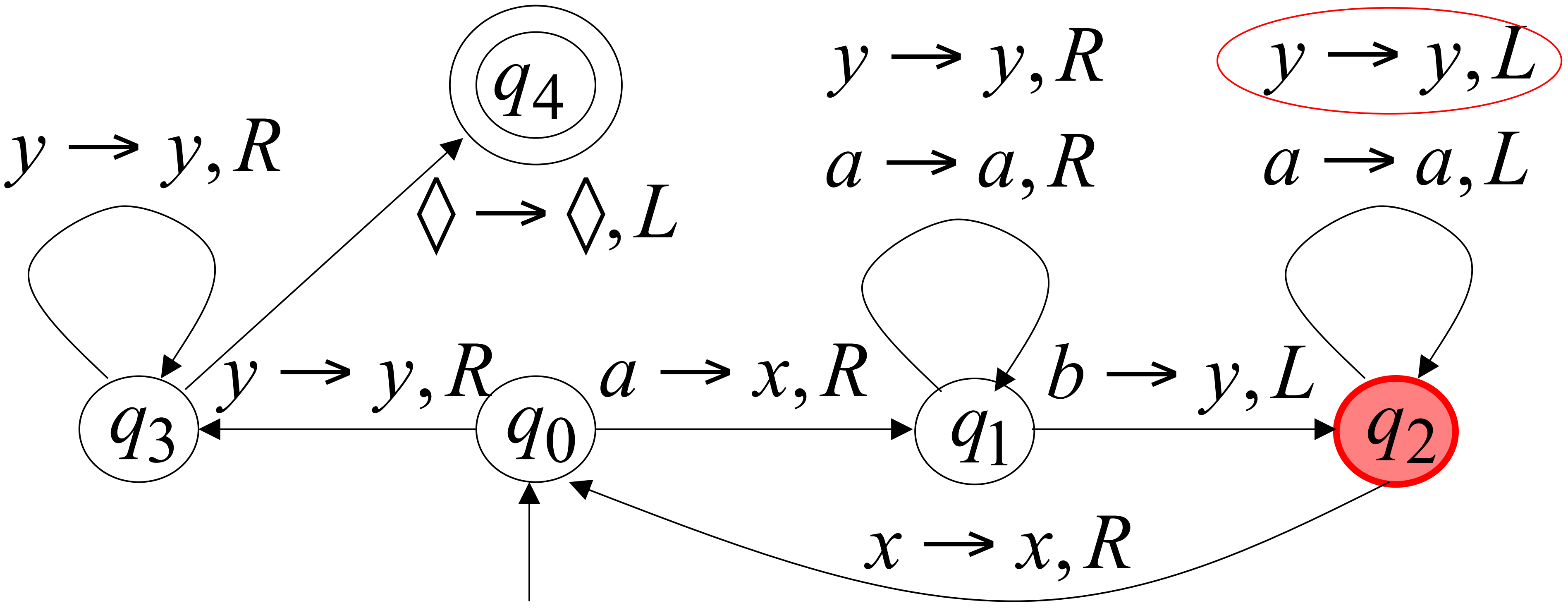


q_4
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
$y \rightarrow y, R$
$a \rightarrow a, R$
q_1
$b \rightarrow y, L$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_2
$x \rightarrow x, R$

Time 8

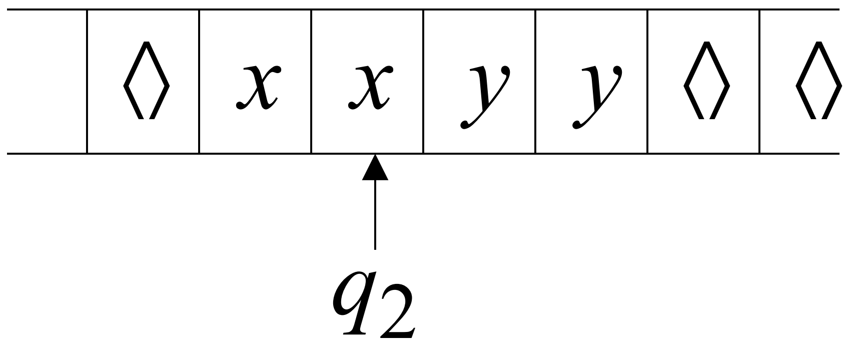


◊
x
x
y
y
◊
◊
q_2

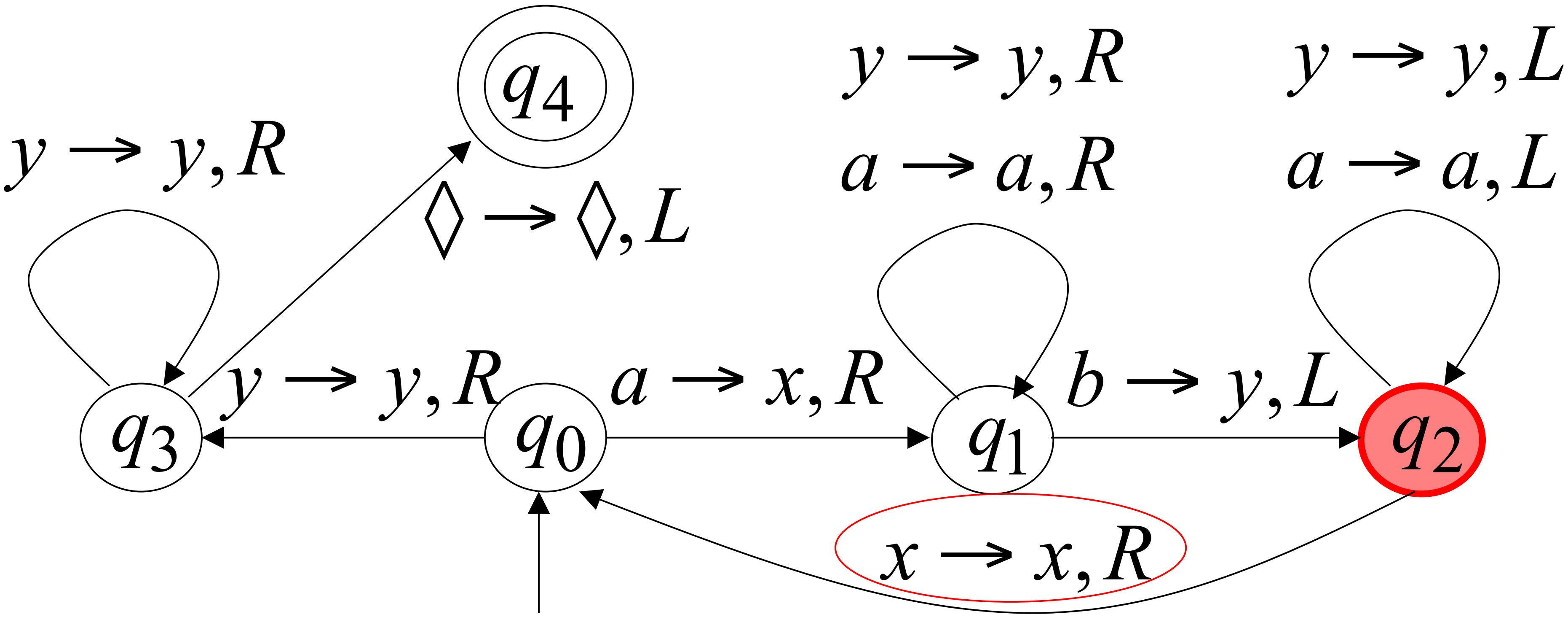


q_0
q_1
q_2
q_3
q_4
$a \rightarrow x, R$
$y \rightarrow y, R$
$a \rightarrow a, R$
$b \rightarrow y, L$
$y \rightarrow y, L$
$a \rightarrow a, L$
$x \rightarrow x, R$
$y \rightarrow y, R$
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$

Time 9

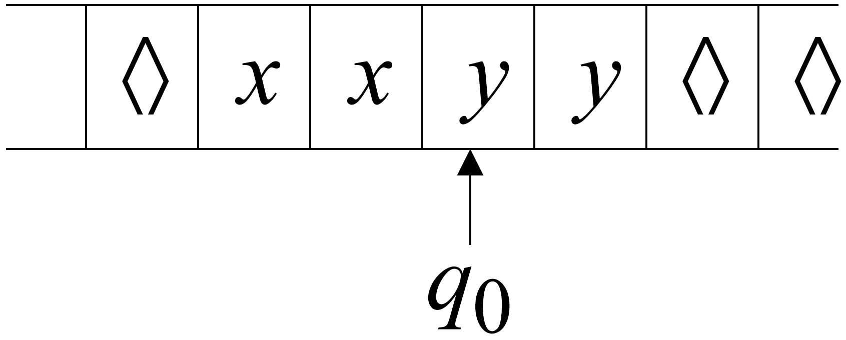


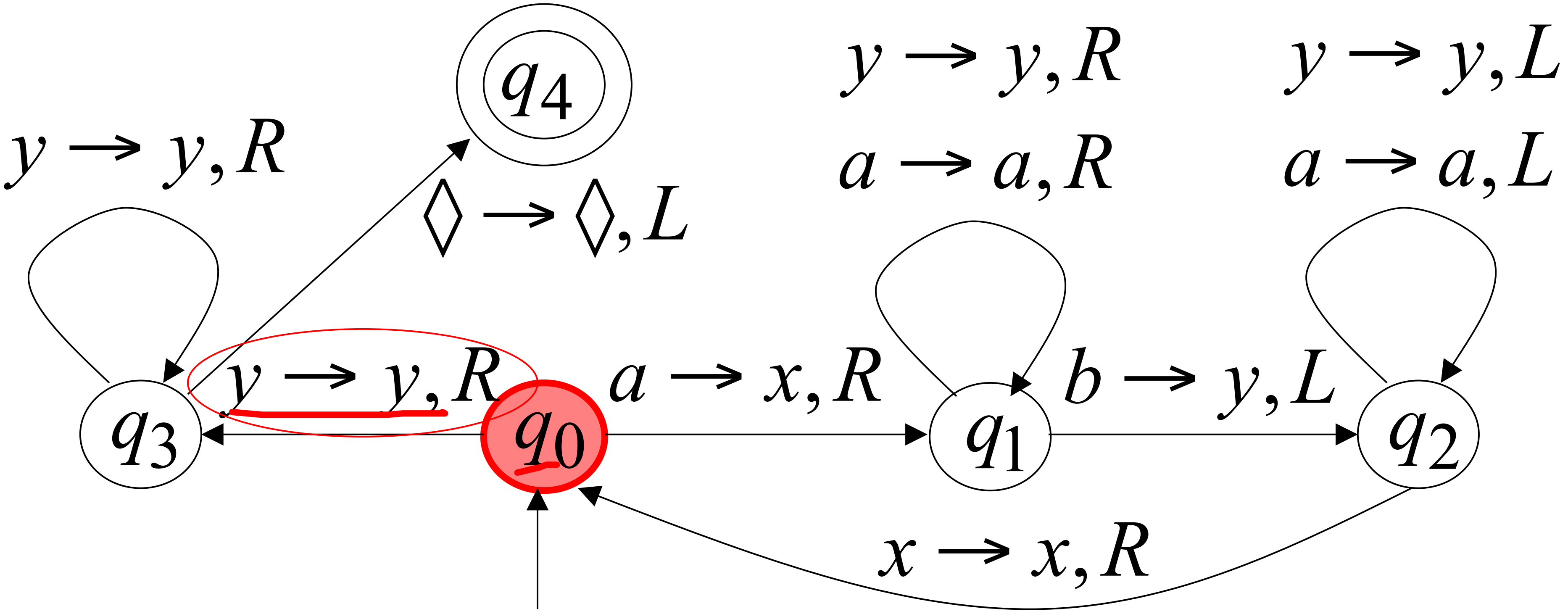
$\lozenge$
x
x
y
y
$\lozenge$
$\lozenge$
q_2



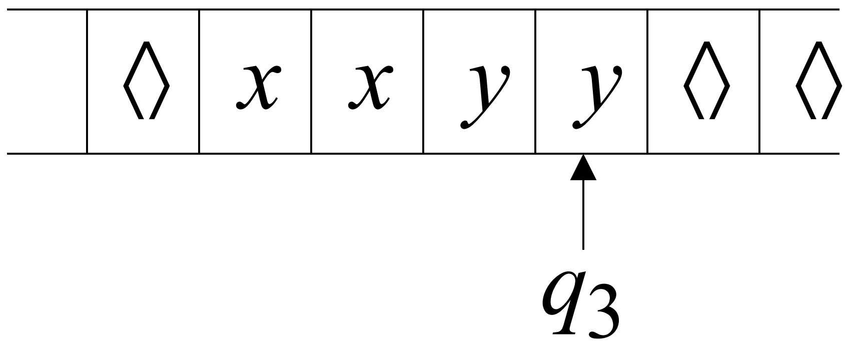
q_4
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

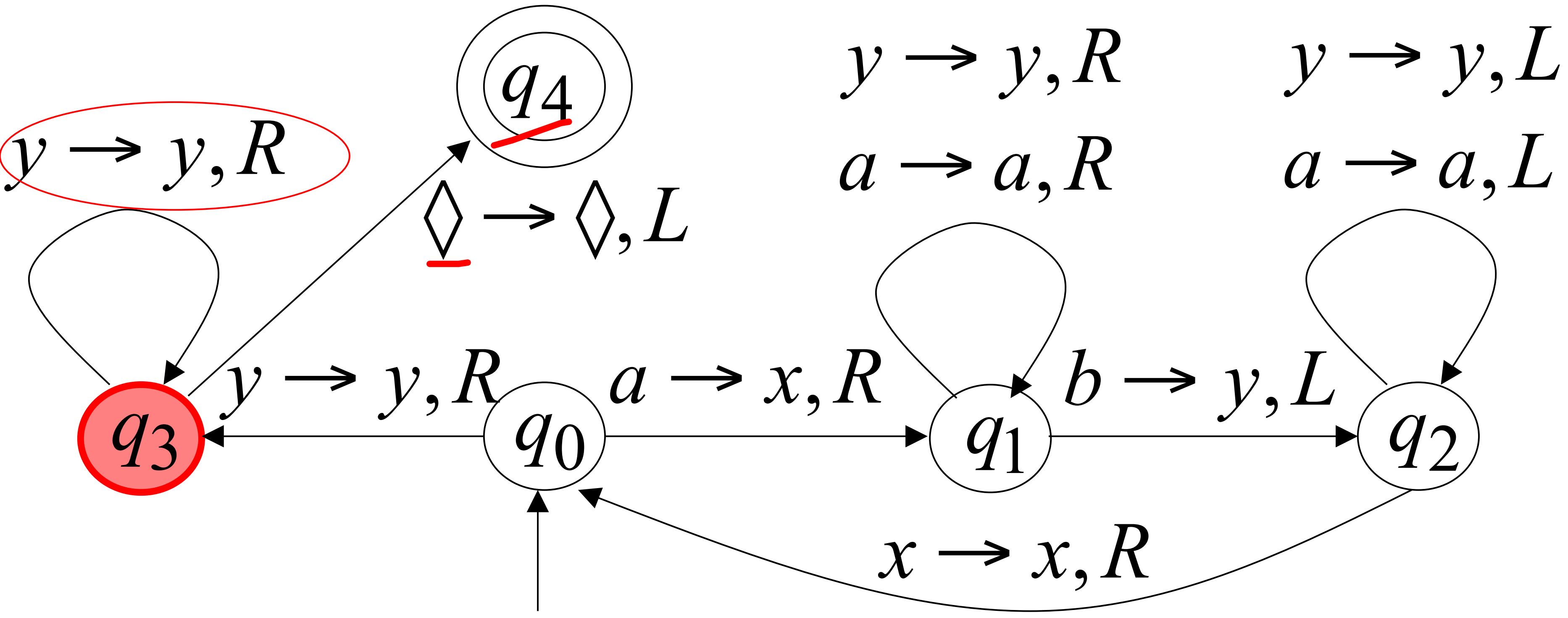
Time 10


◊
x
x
y
y
◊
◊
q_0

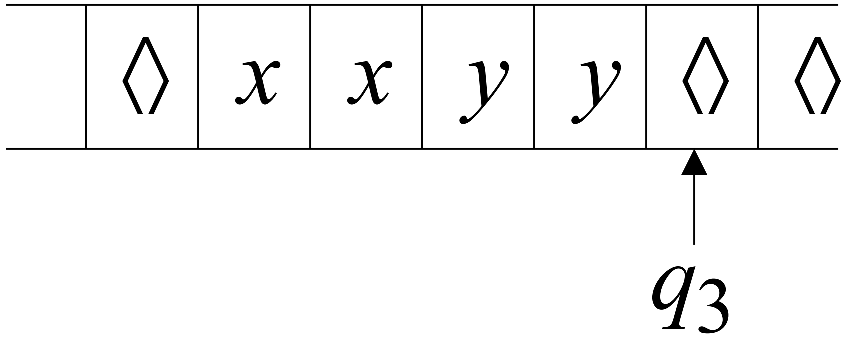

q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

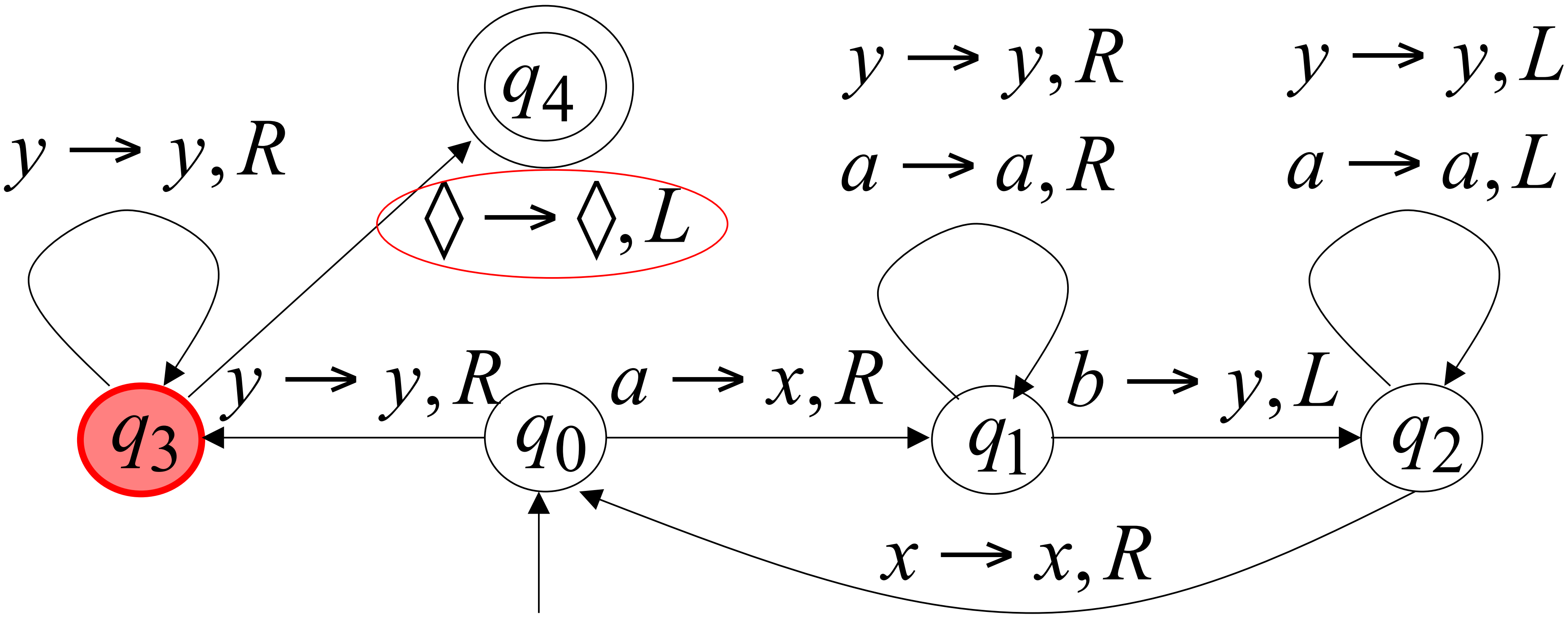
Time 11


◊
x
x
y
y
◊
◊
q_3

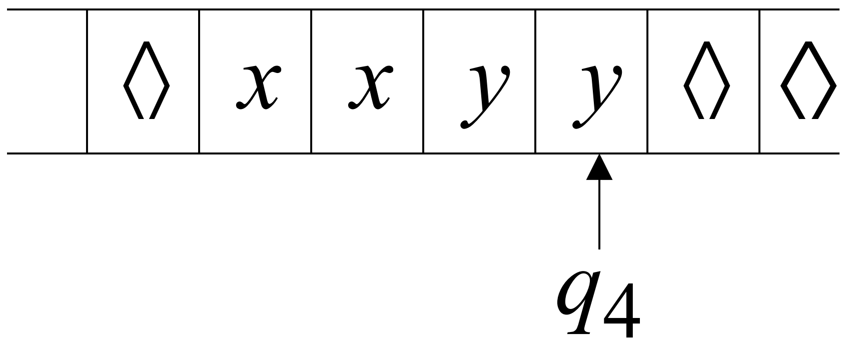

q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$

Time 12

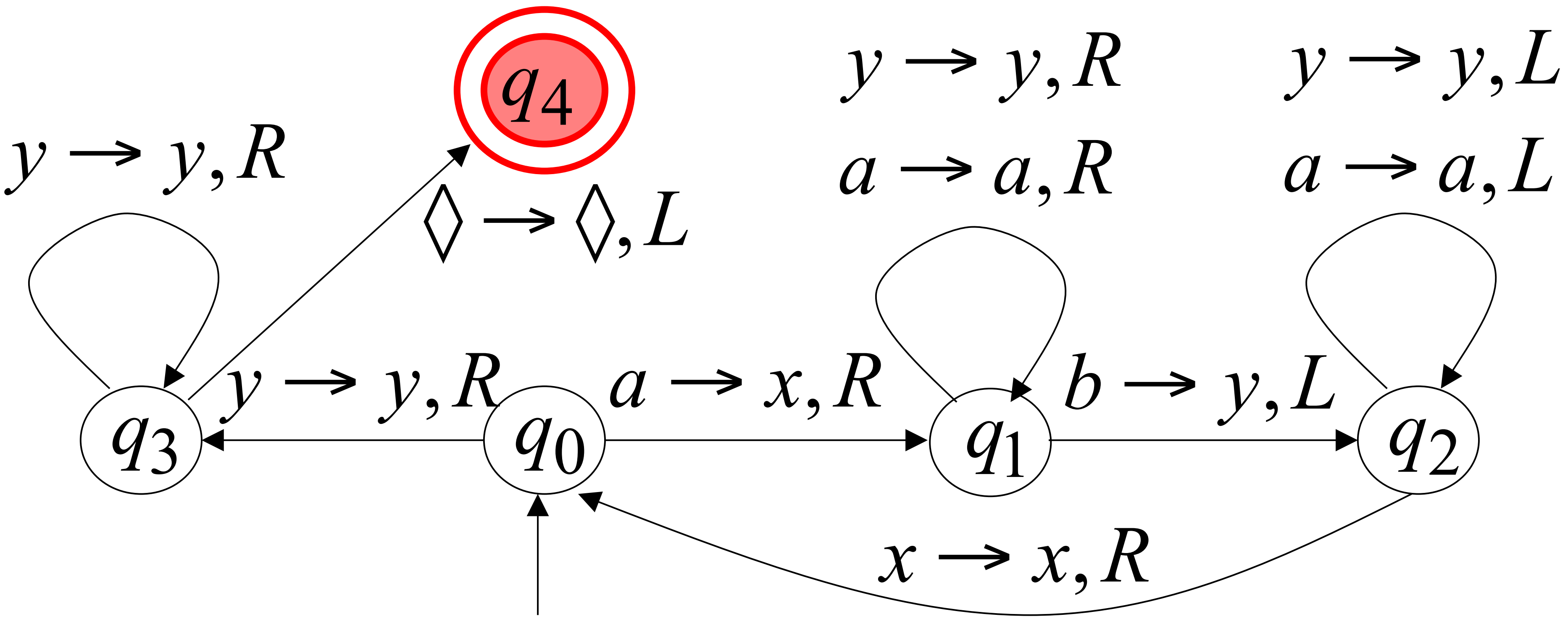

◊
x
x
y
y
◊
◊
q_3


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

Time 13



Halt & Accept



Observation:

If we modify the
machine for the language $\{a^n b^n\}$

we can easily construct
a machine for the language $\{\underline{a^{\underline{n}} b^{\underline{n}} c^{\underline{n}}}\}$

Formal Definitions for Turing Machines

Transition Function

$$\begin{tikzcd}
q_1 \arrow[rr, "\underline{a} \rightarrow \underline{b}{,}\, \underline{R}"] & & q_2
\end{tikzcd}$$ ✓

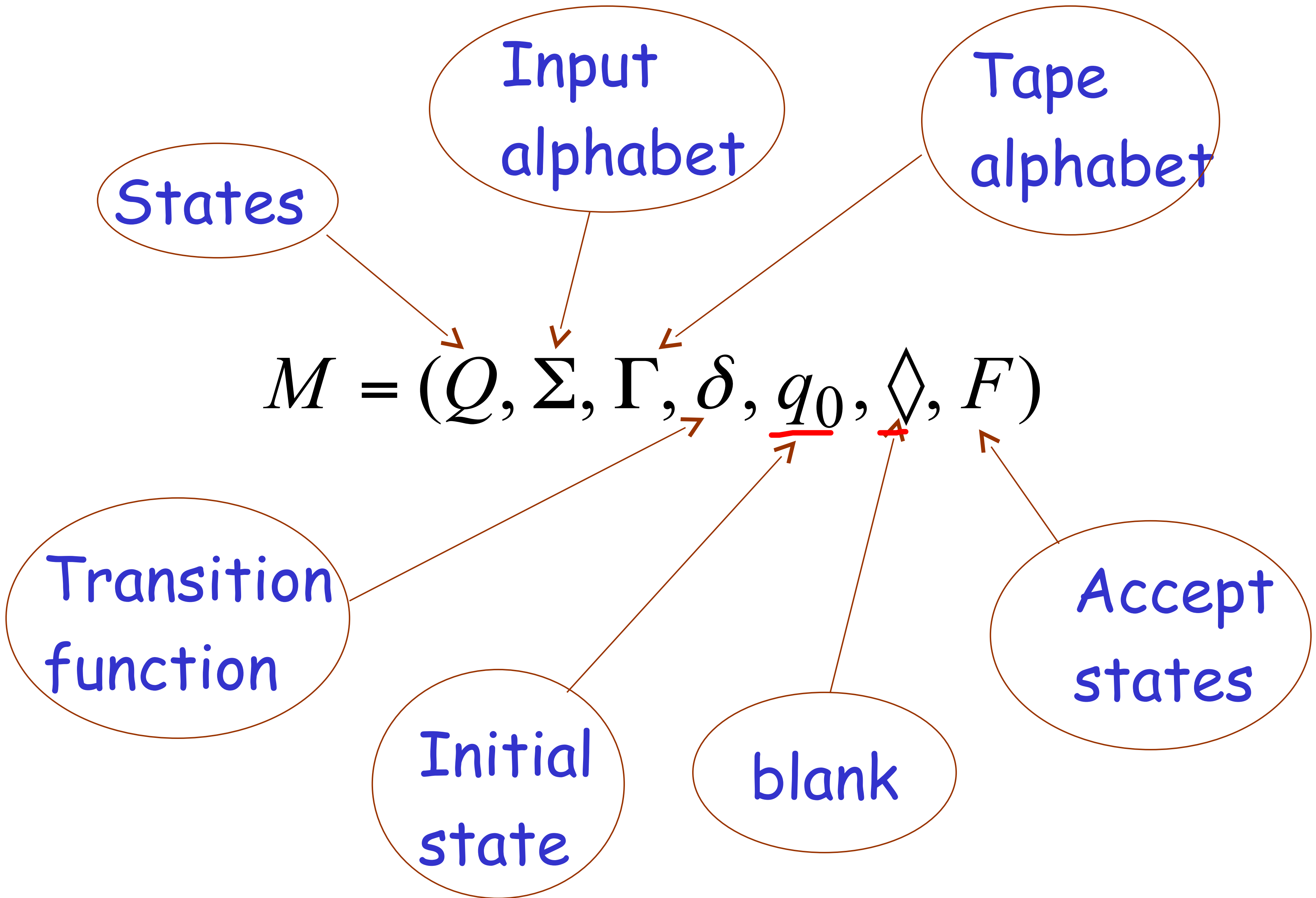
$$\underline{\delta(q_1, a) = (q_2, b, R)}$$ ✓

Transition Function

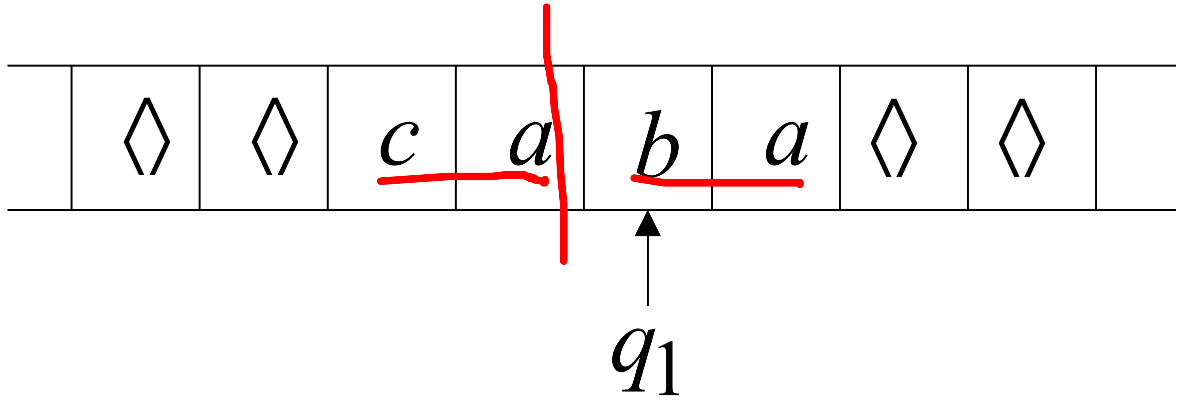
$$\begin{tikzcd} \boxed{\underline{q_1}} \arrow[r, "c \rightarrow d\,,L"] & \boxed{q_2} \end{tikzcd}$$

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

Turing Machine:


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

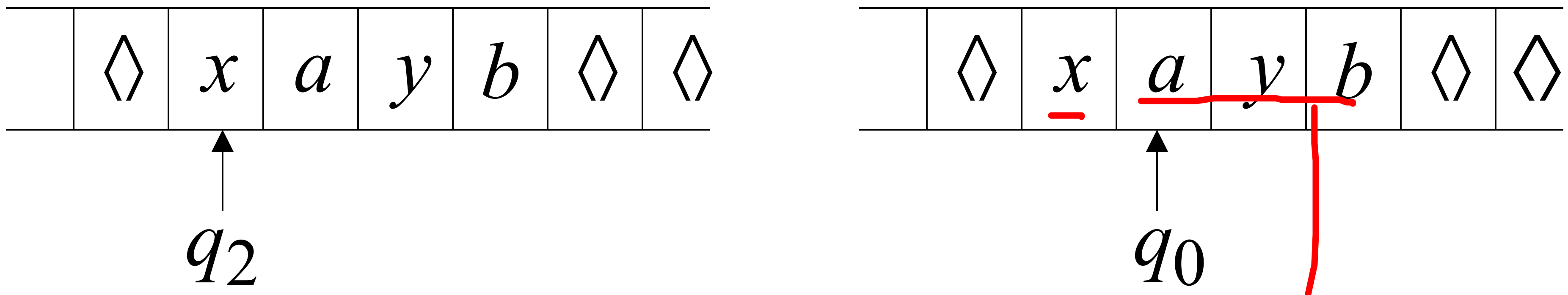
Configuration



Instantaneous description: $\underline{ca}\,q_1\,\underline{ba}$

Time 4

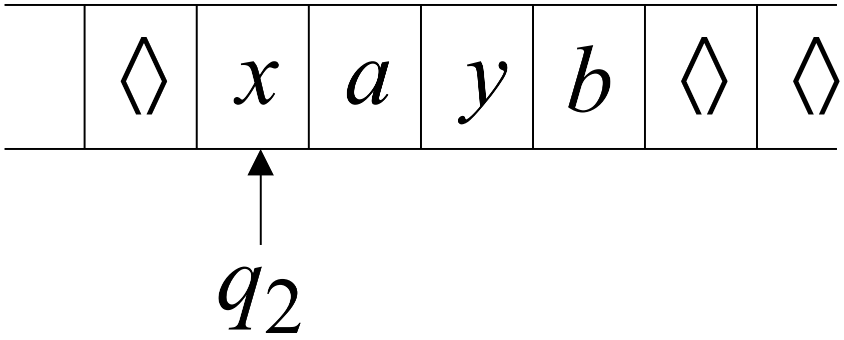
Time 5



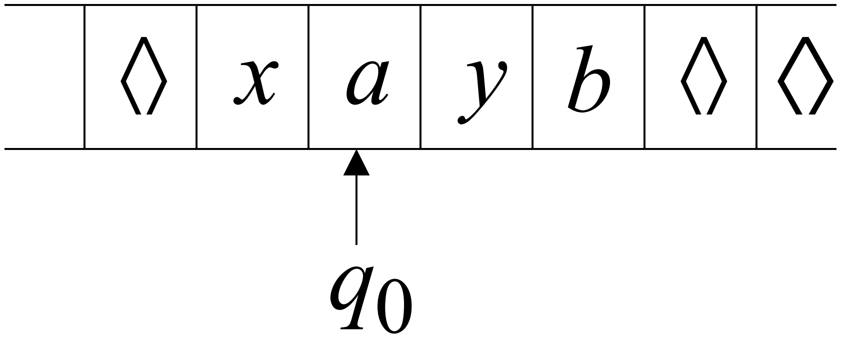
A Move: $\underline{}\,q_2\,\underline{xayb} \succ \underline{x}\,q_0\,ayb$

(yields in one mode)

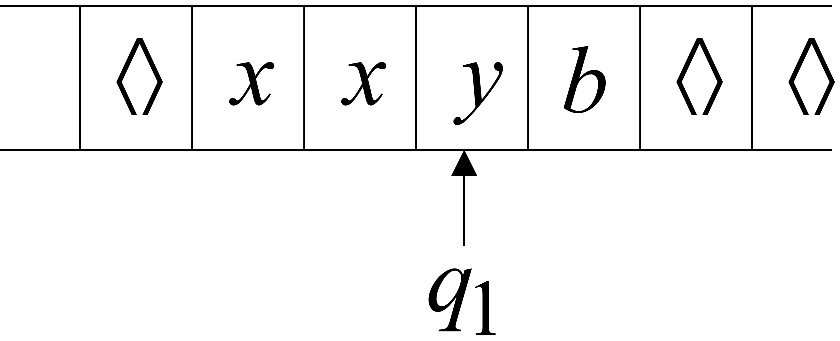
Time 4



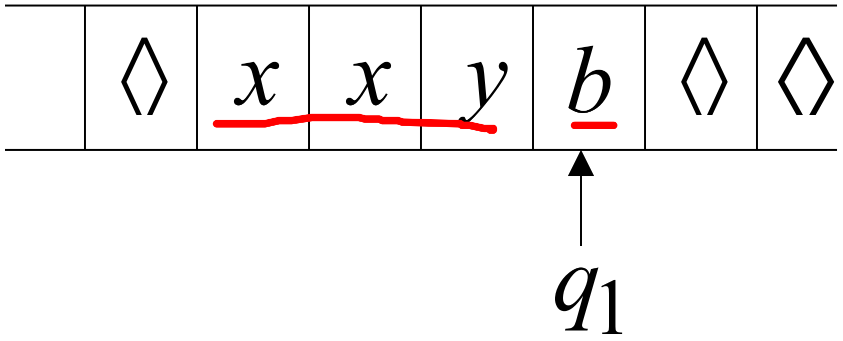
Time 5



Time 6



Time 7



A computation

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

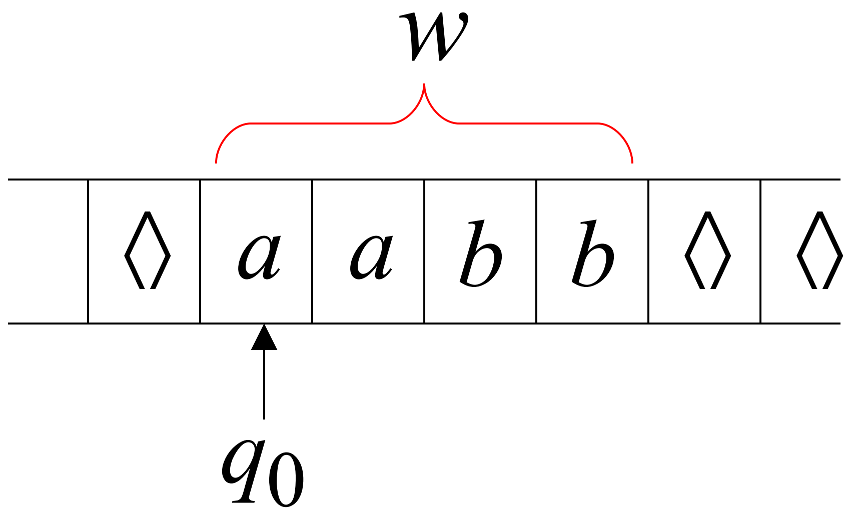
$$\underline{q_2\, xayb \succ x\, q_0\, ayb \succ xx\, q_1\, yb} \succ \underline{xxy\, q_1\, b}$$

Equivalent notation:

$$\underline{q_2\, xayb} \overset{*}{\succ} \underline{xxy\, q_1\, b}$$

Initial configuration: $\underline{q_0\ w}$

Input string



The Accepted Language

For any Turing Machine M

$$L(M) = \{ w : \underbrace{q_0\, w}_{\text{Initial state}} \overset{*}{\succ} x_1\, \underbrace{q_f}_{\text{Accept state}}\, x_2 \}$$

If a language L is accepted by a Turing machine M then we say that L is:

- Turing Recognizable

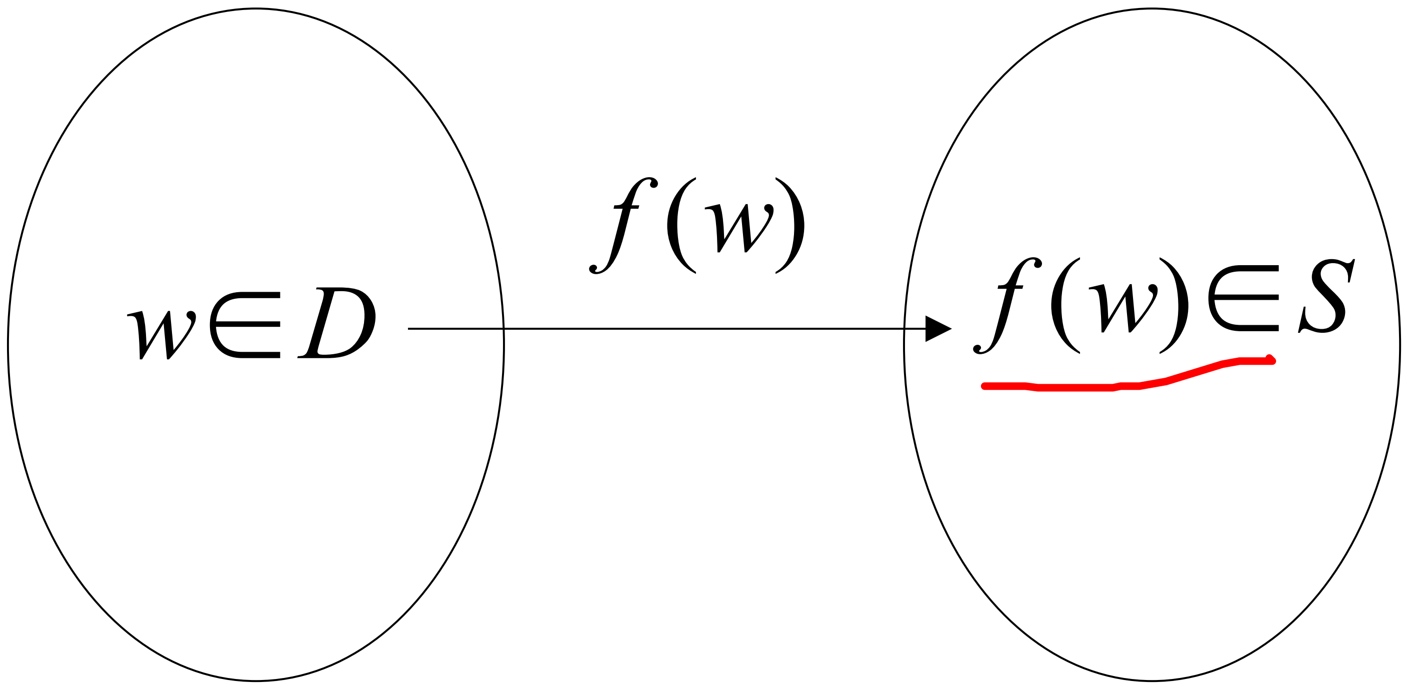
Other names used:

- Turing Acceptable
- Recursively Enumerable

Computing Functions with Turing Machines

A function $f(w)$ has:

Domain: D Result Region: S



A function may have many parameters:

Example: Addition function

$$f(\underline{x, y}) = \underline{x} + \underline{y}$$

Integer Domain

Decimal: $\underline{5}$

Binary: $\underline{101}$

Unary: $\underline{11111}$

We prefer **unary** representation:

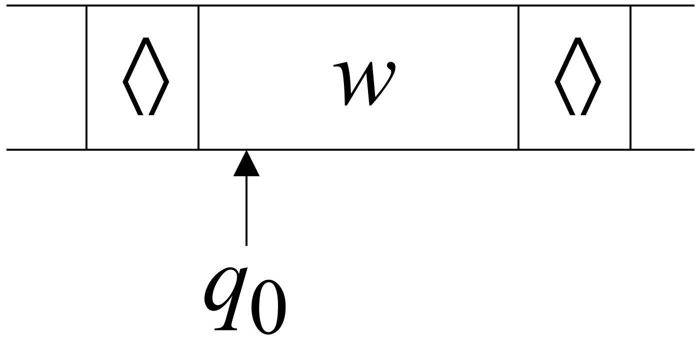
easier to manipulate with Turing machines

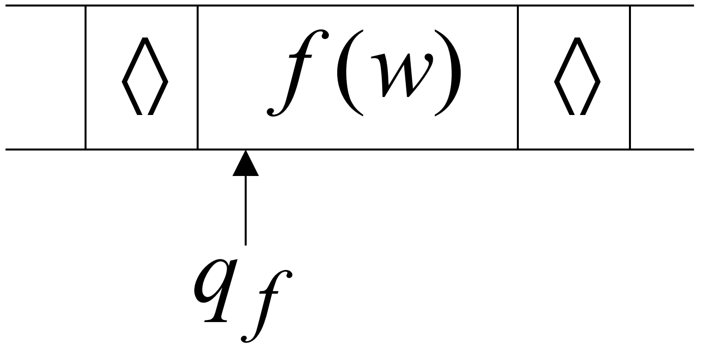
Definition:

A function f is computable if there is a Turing Machine M such that:

Initial configuration

Final configuration





initial state

accept state

For all $\underline{w} \in D$ Domain

In other words:

A function f is computable if there is a Turing Machine M such that:

$$\underbrace{q_0 w}_{\text{Initial Configuration}} \overset{*}{\succ} \underbrace{q_f f(w)}_{\text{Final Configuration}}$$

For all $w \in D$ Domain

Example

The function $f(x, y) = x + y$ is computable

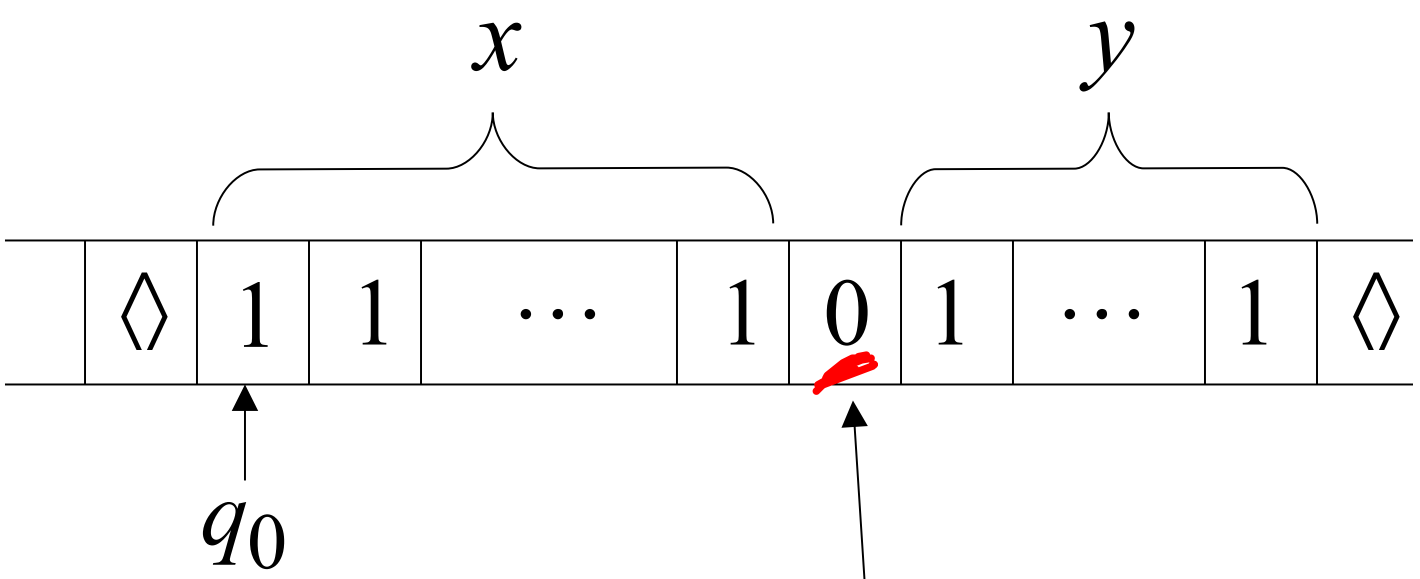
x, y are integers

Turing Machine:

Input string: $\underline{x0y}$ unary

Output string: $\underline{xy}0$ unary

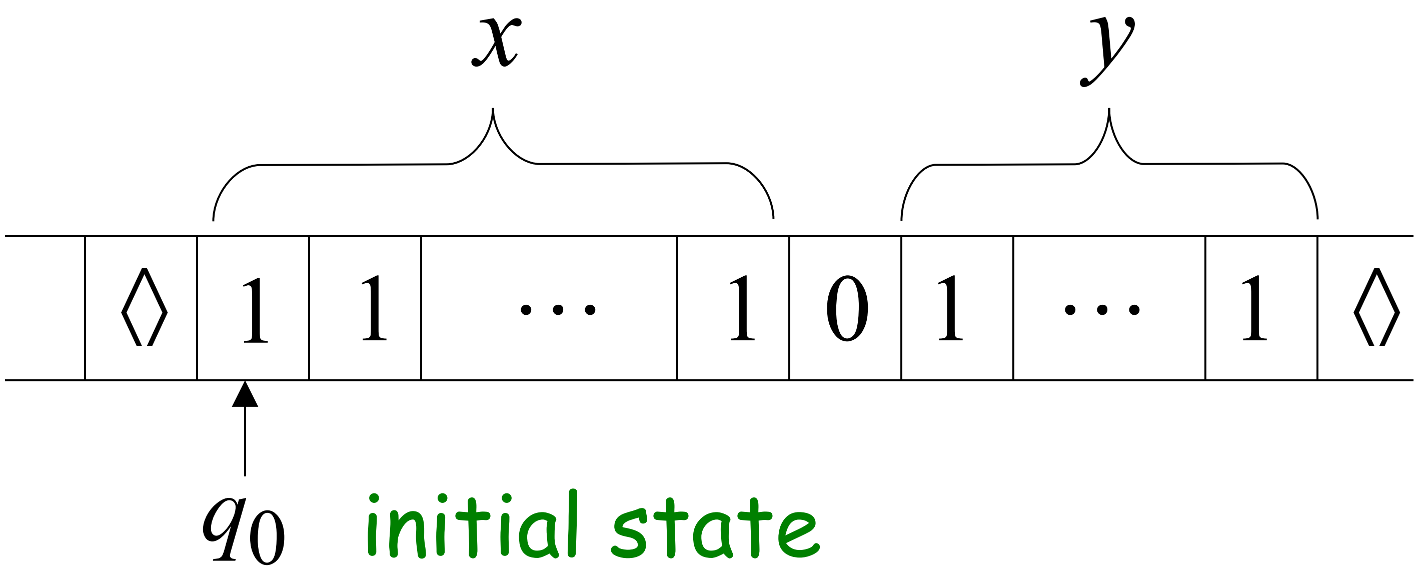
Start



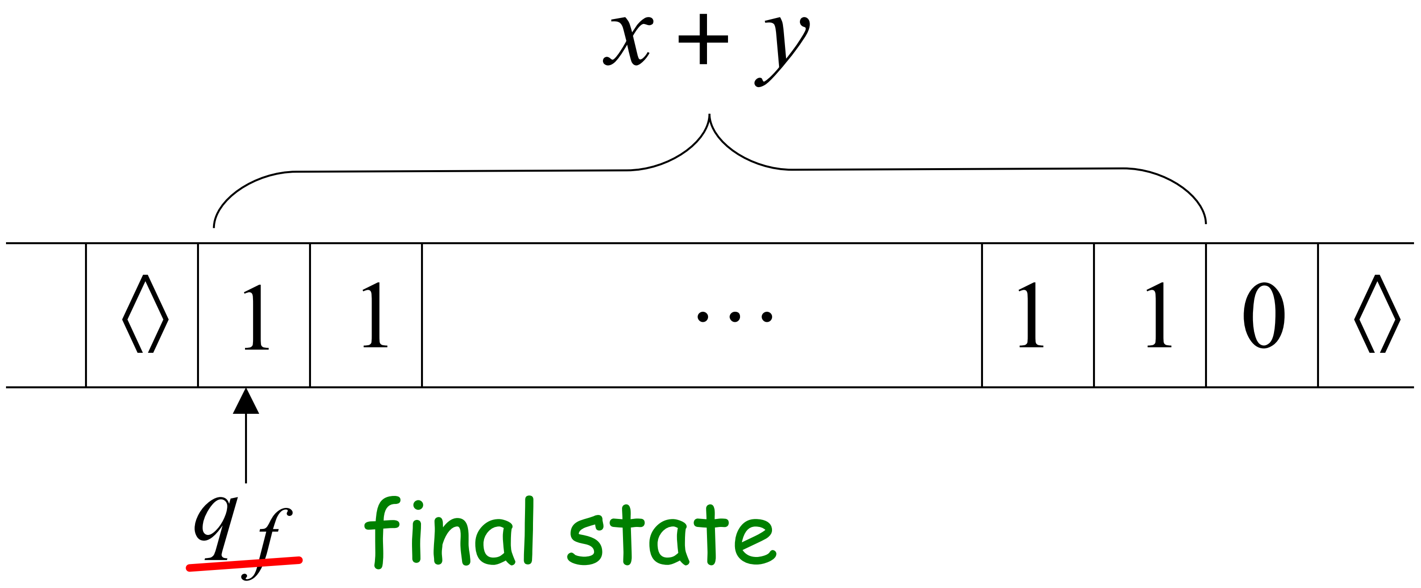
initial state

The 0 is the delimiter that separates the two numbers

Start

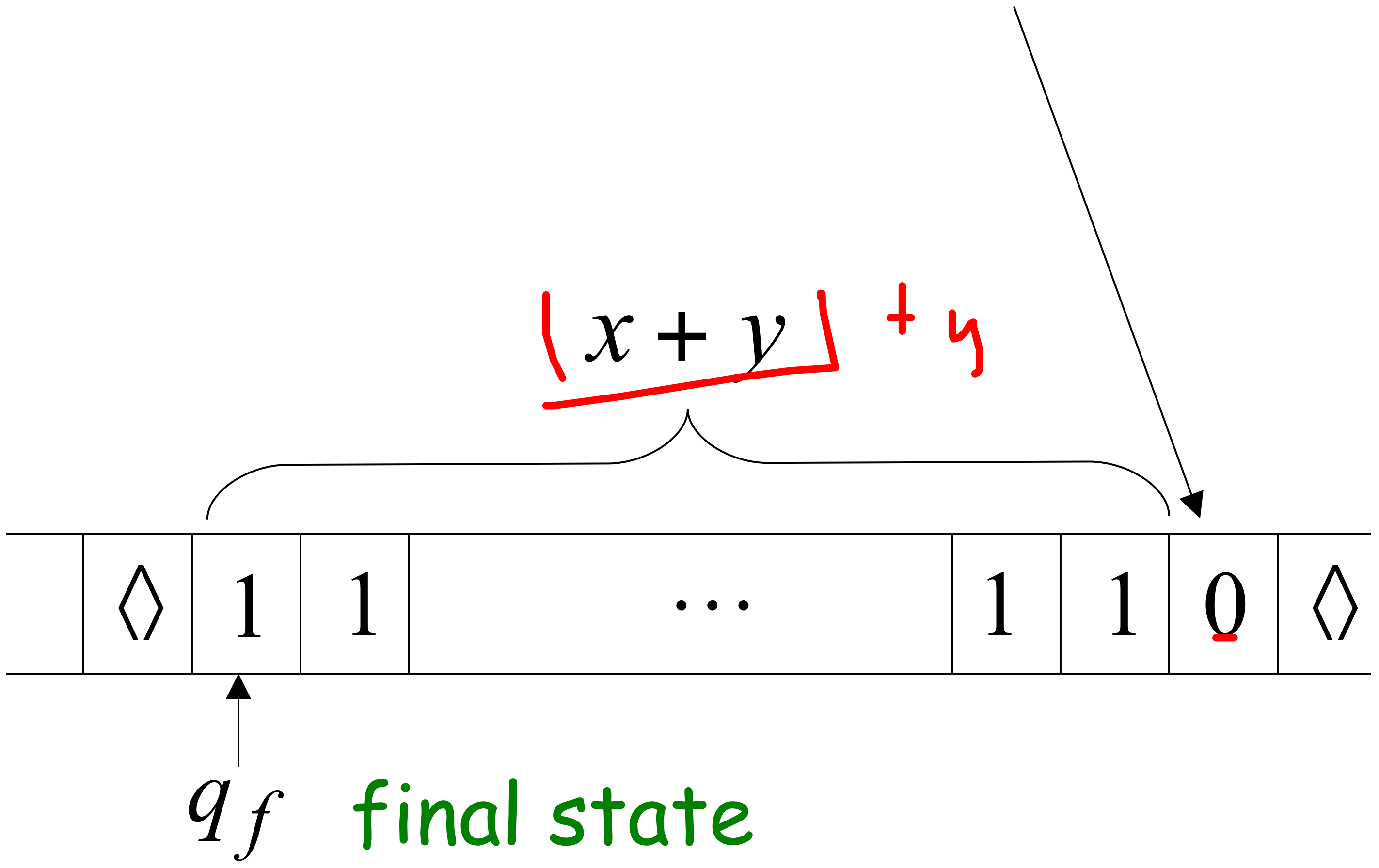


Finish



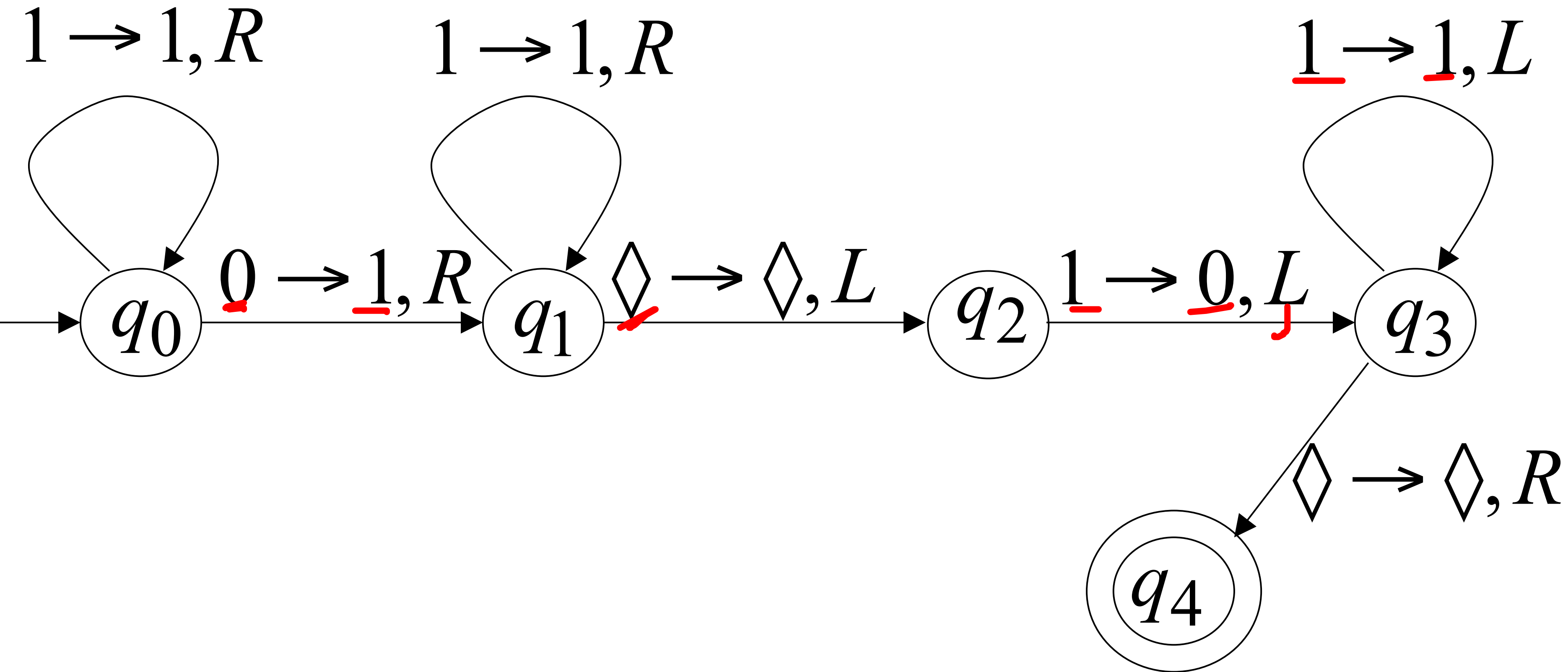
The 0 here helps when we use the result for other operations

Finish



Turing machine for function $f(x,y) = x + y$

$\underline{|||+||} = |||||$ $\quad |||0|| \Rightarrow |||||0$

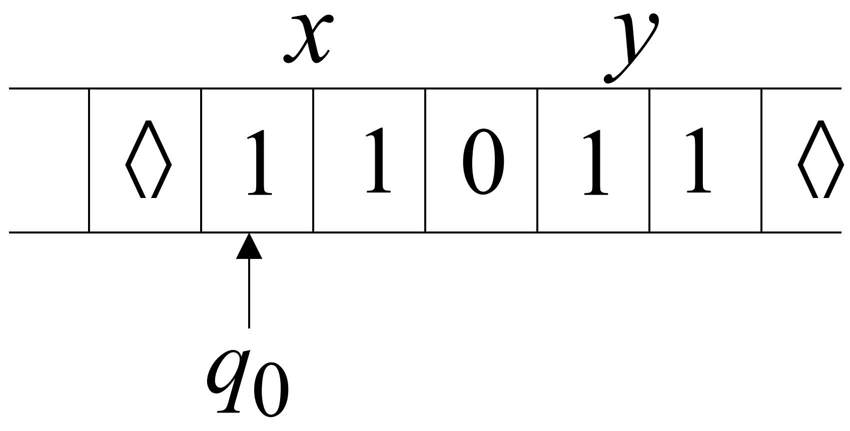


Execution Example:

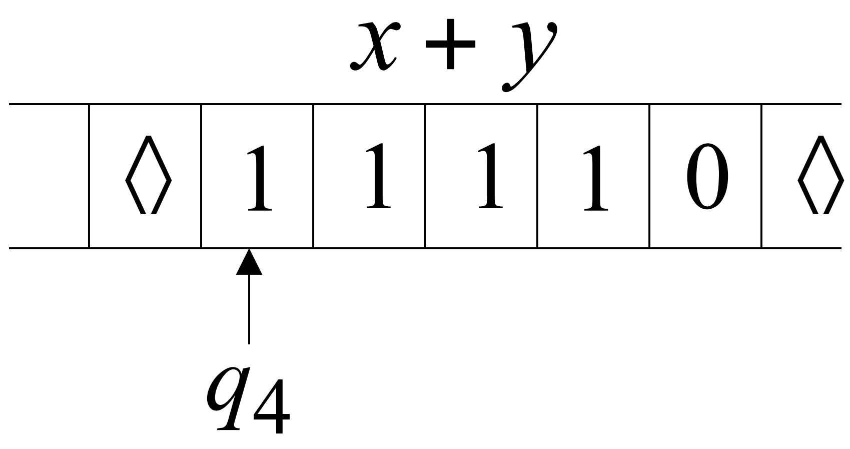
$x = 11$ (=2)

$y = 11$ (=2)

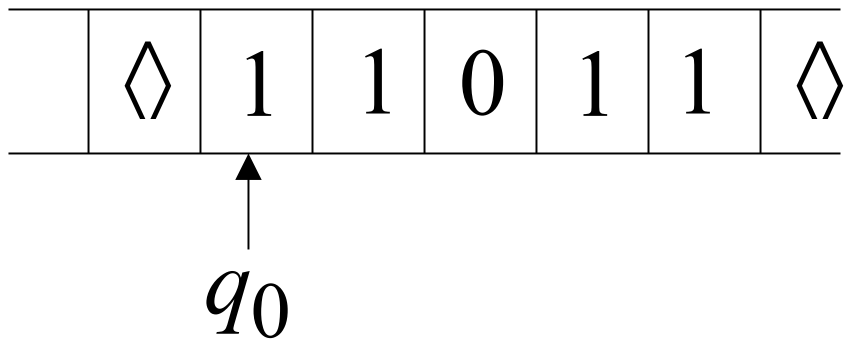
Time 0

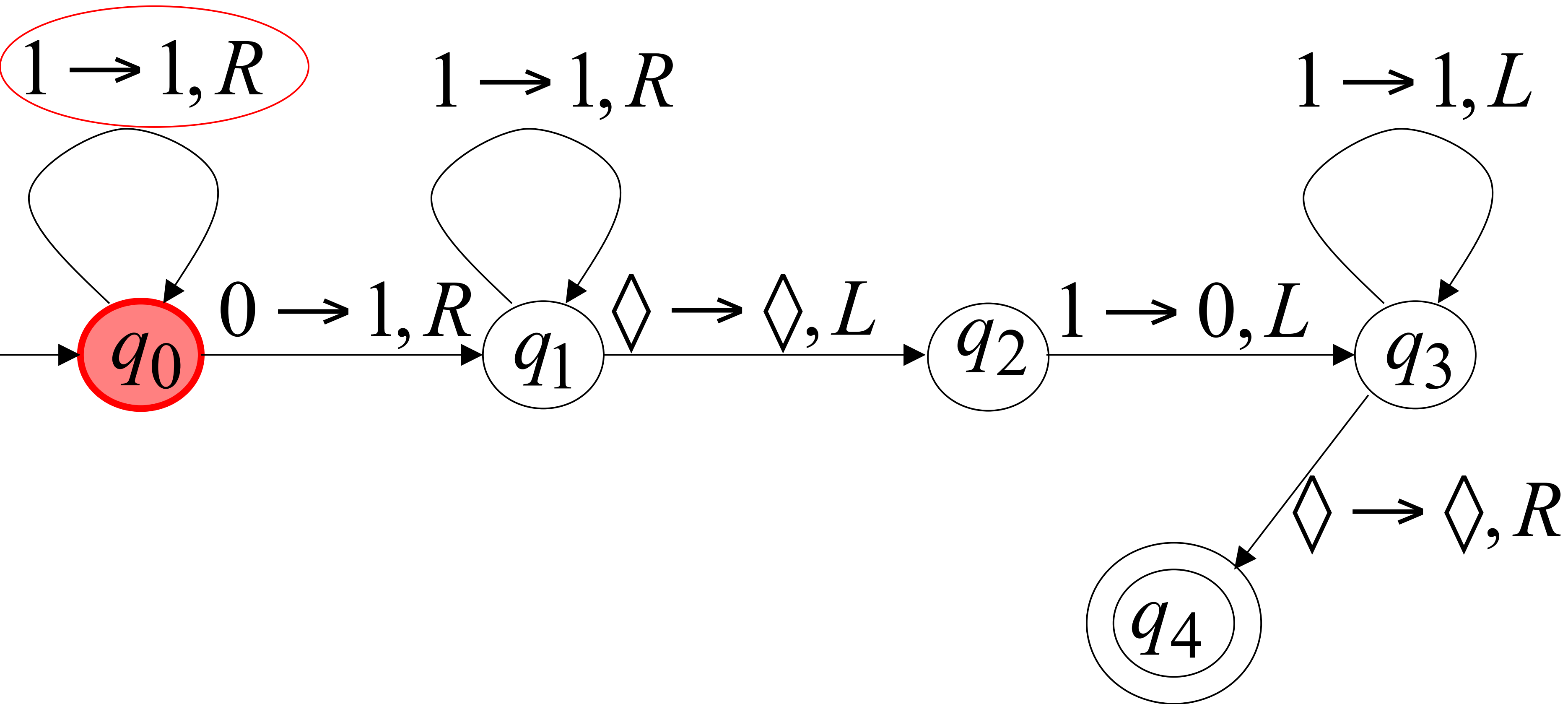


Final Result

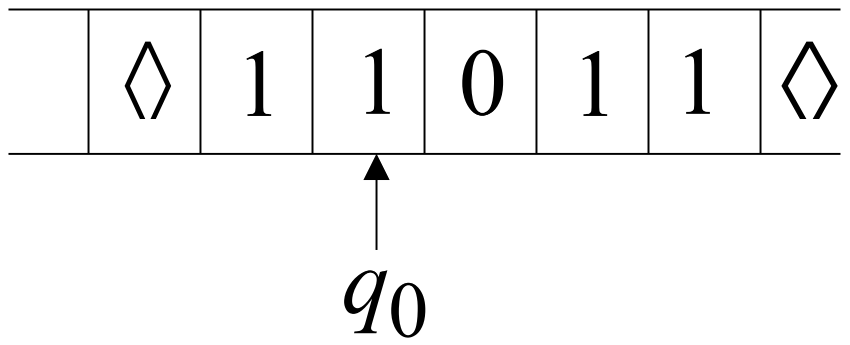


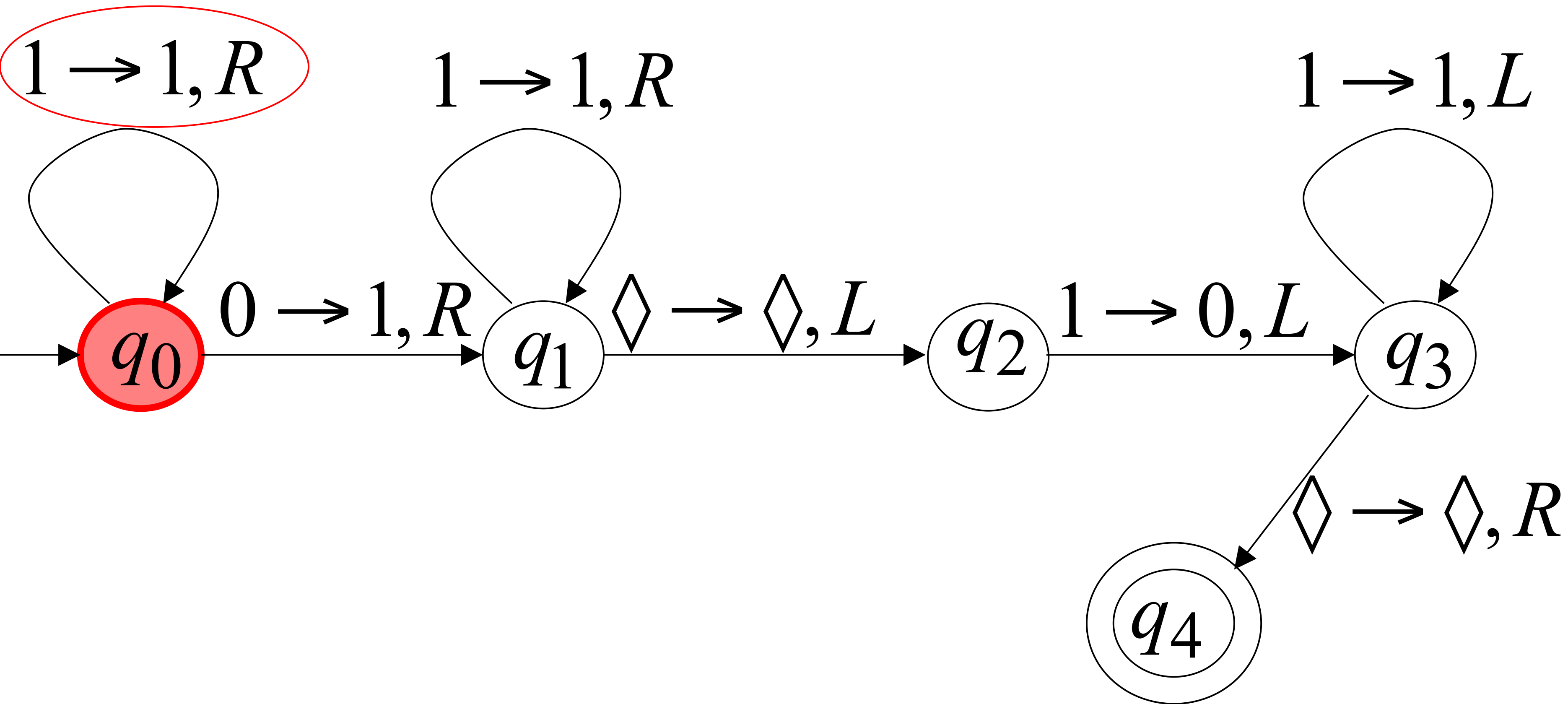
Time 0


◊
1
1
0
1
1
◊
q_0

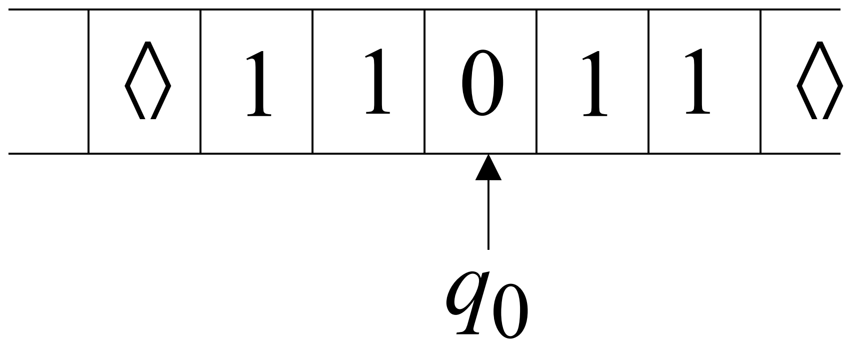

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\lozenge \rightarrow \lozenge, L$
q_2
$1 \rightarrow 0, L$
q_3
$\lozenge \rightarrow \lozenge, R$
q_4

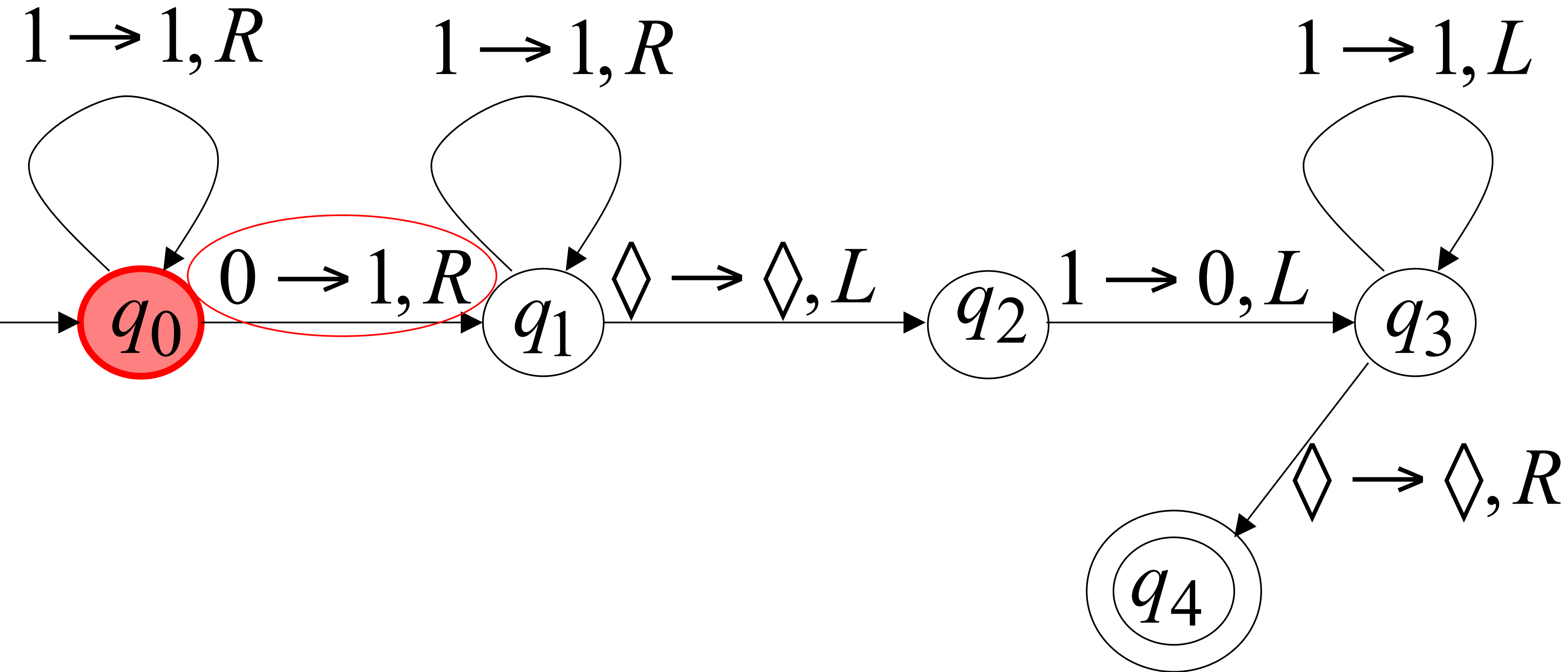
Time 1


◊
1
1
0
1
1
◊
q_0

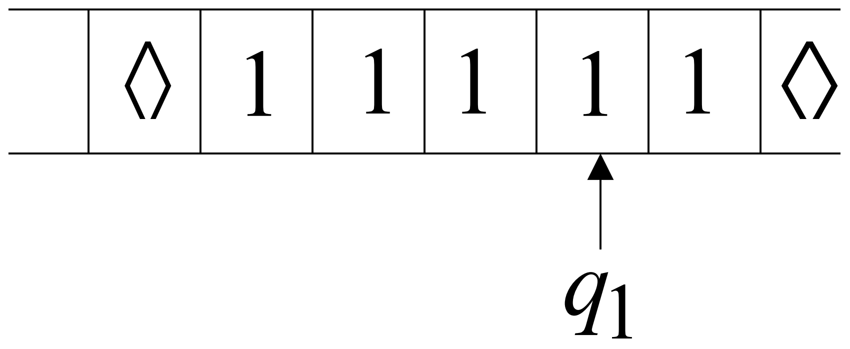

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$◊ \rightarrow ◊, L$
$1 \rightarrow 0, L$
$◊ \rightarrow ◊, R$
q_0
q_1
q_2
q_3
q_4

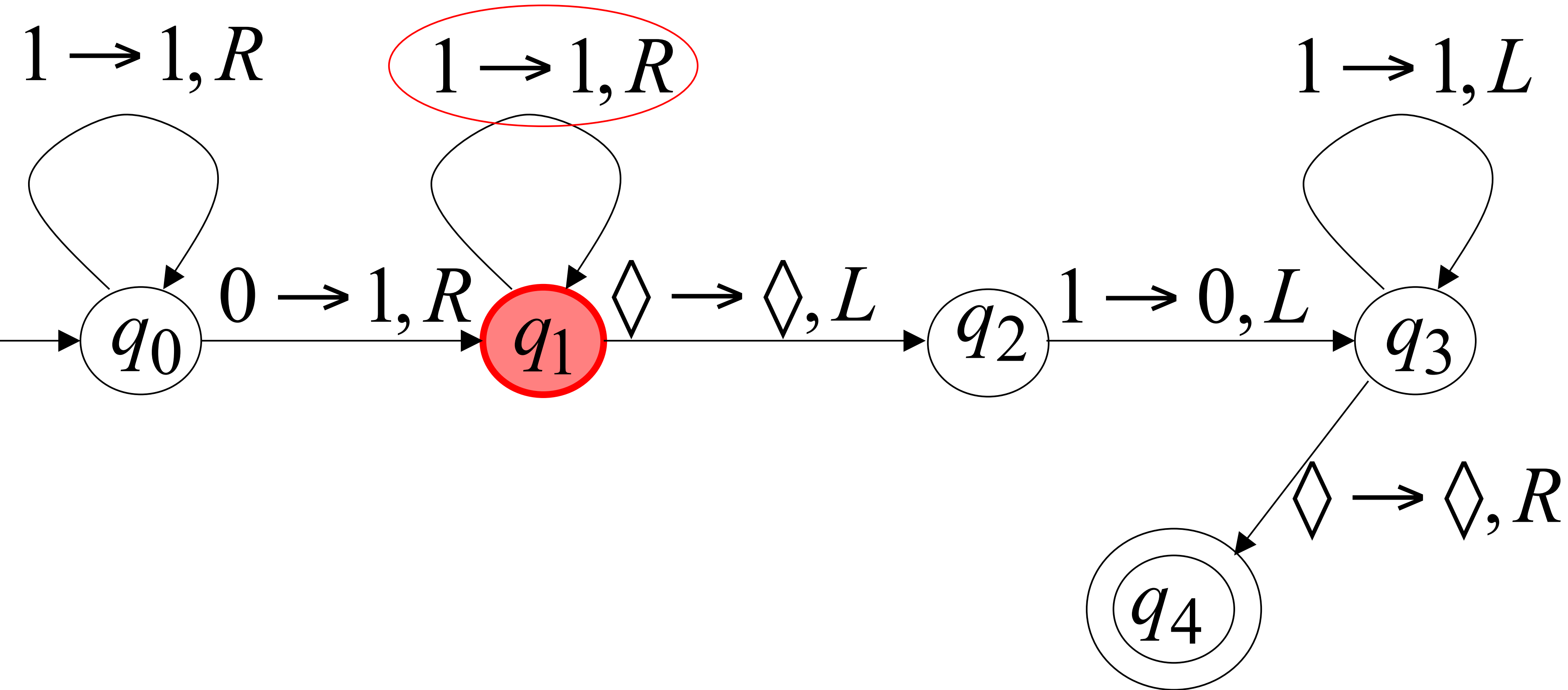
Time 2


◊ 1 1 0 1 1 ◊
q_0

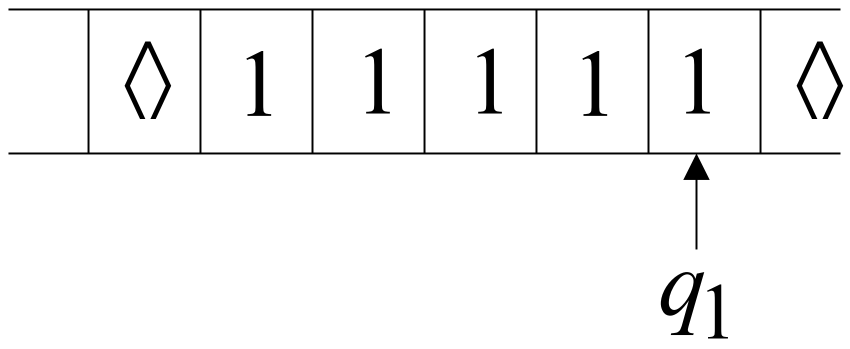

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
$\lozenge \rightarrow \lozenge, R$
q_0
q_1
q_2
q_3
q_4

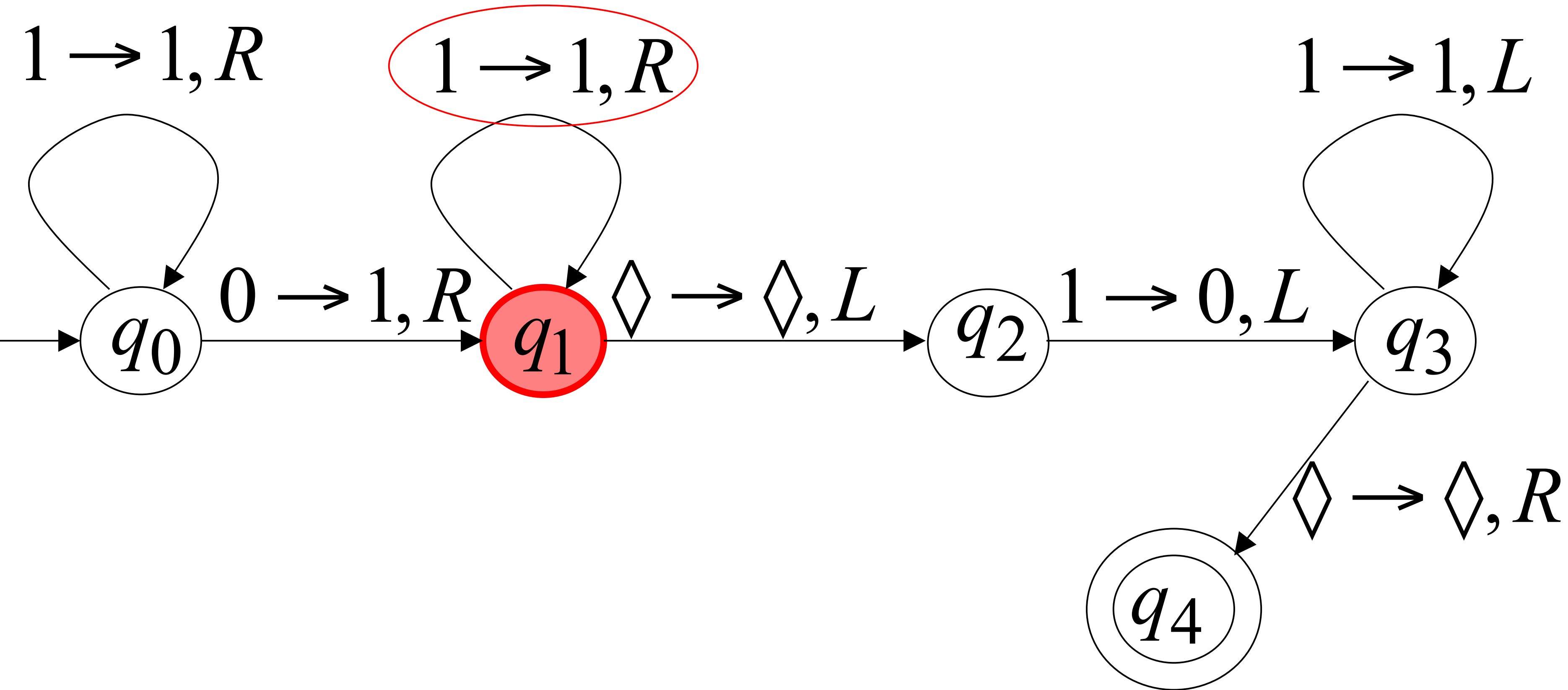
Time 3


◊
1
1
1
1
1
◊
q_1

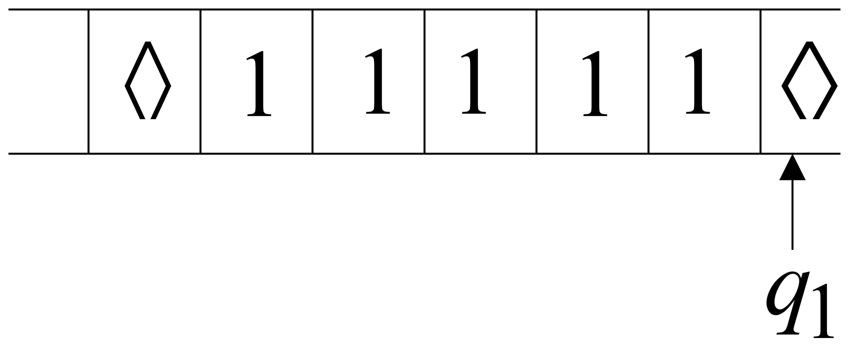

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4

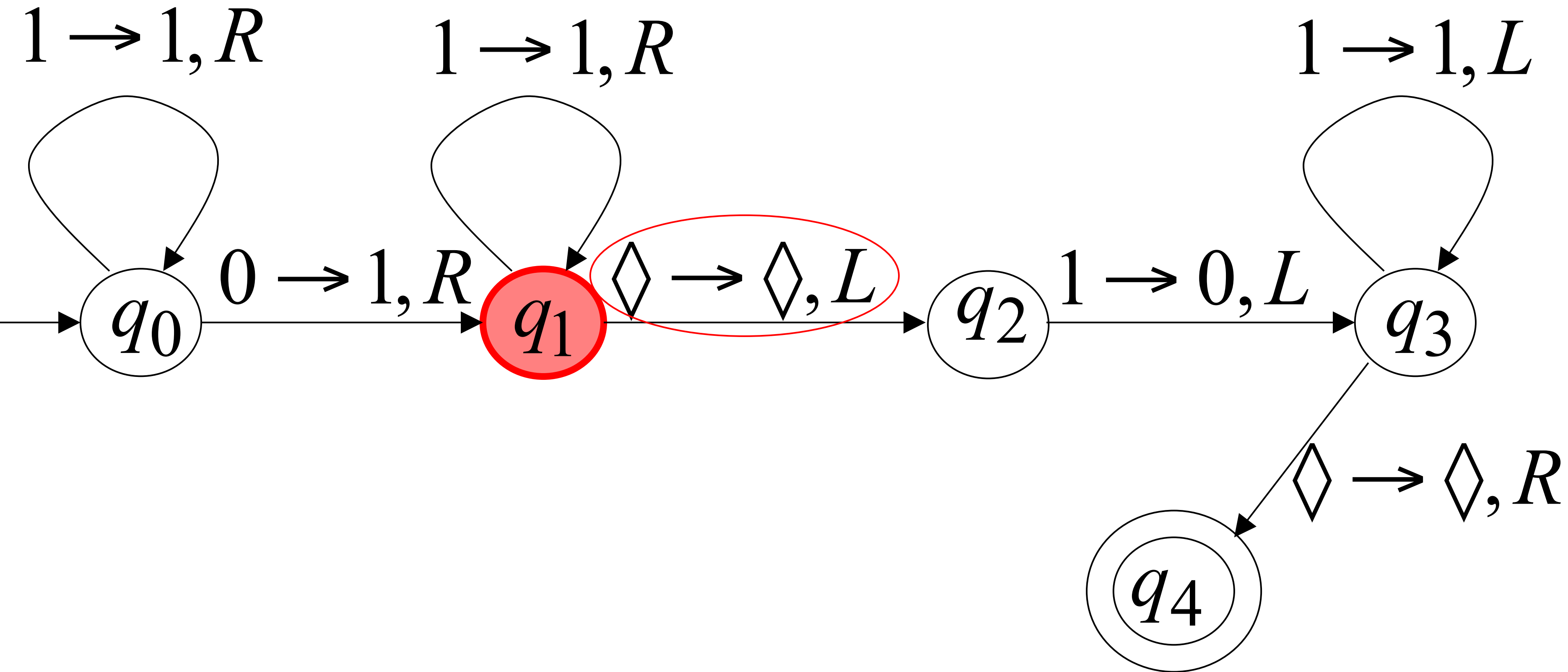
Time 4


◊
1
1
1
1
1
◊
q_1

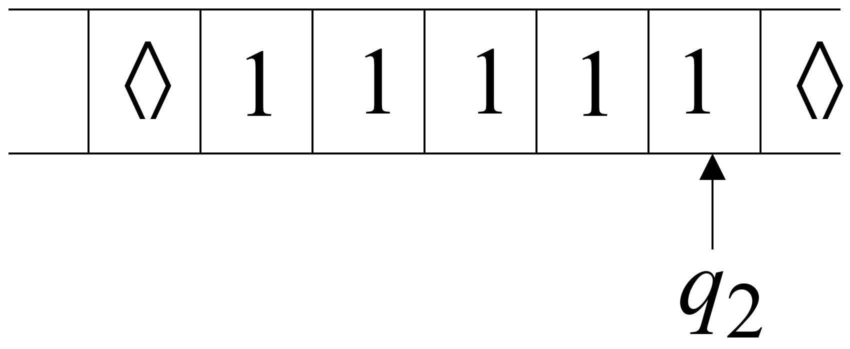

$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
$0 \to 1, R$
$\lozenge \to \lozenge, L$
$1 \to 0, L$
$\lozenge \to \lozenge, R$
q_0
q_1
q_2
q_3
q_4

Time 5

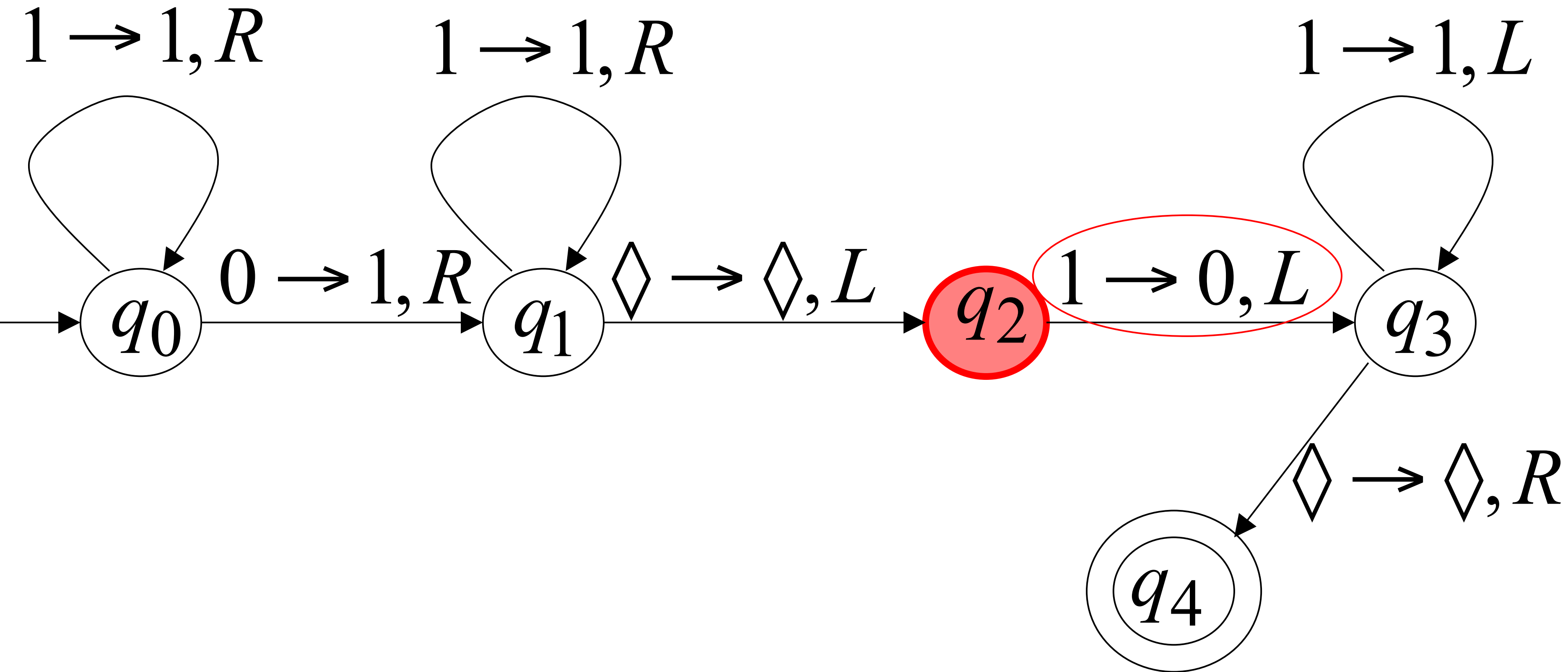

◊
1
1
1
1
1
◊
q_1


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

Time 6

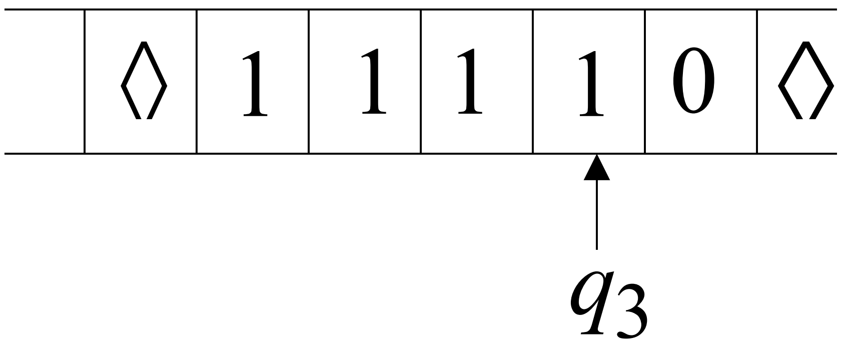


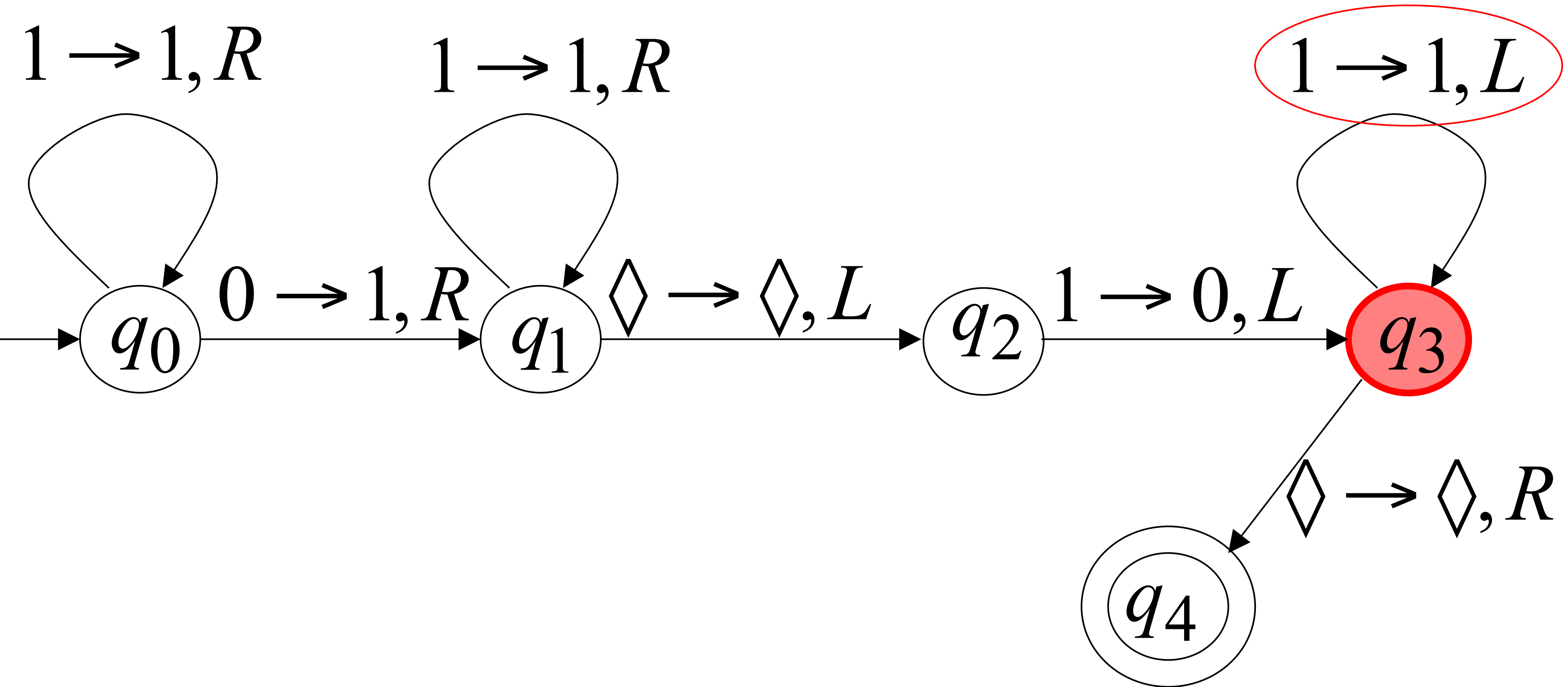
◊
1
1
1
1
1
◊
q_2



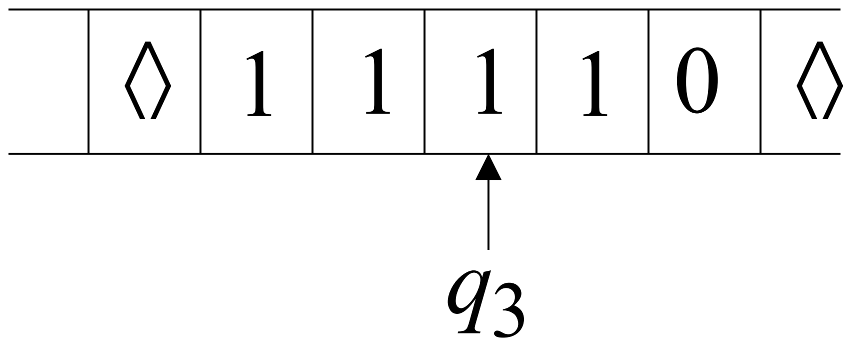
$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
q_0
$0 \to 1, R$
q_1
$\lozenge \to \lozenge, L$
q_2
$1 \to 0, L$
q_3
$\lozenge \to \lozenge, R$
q_4

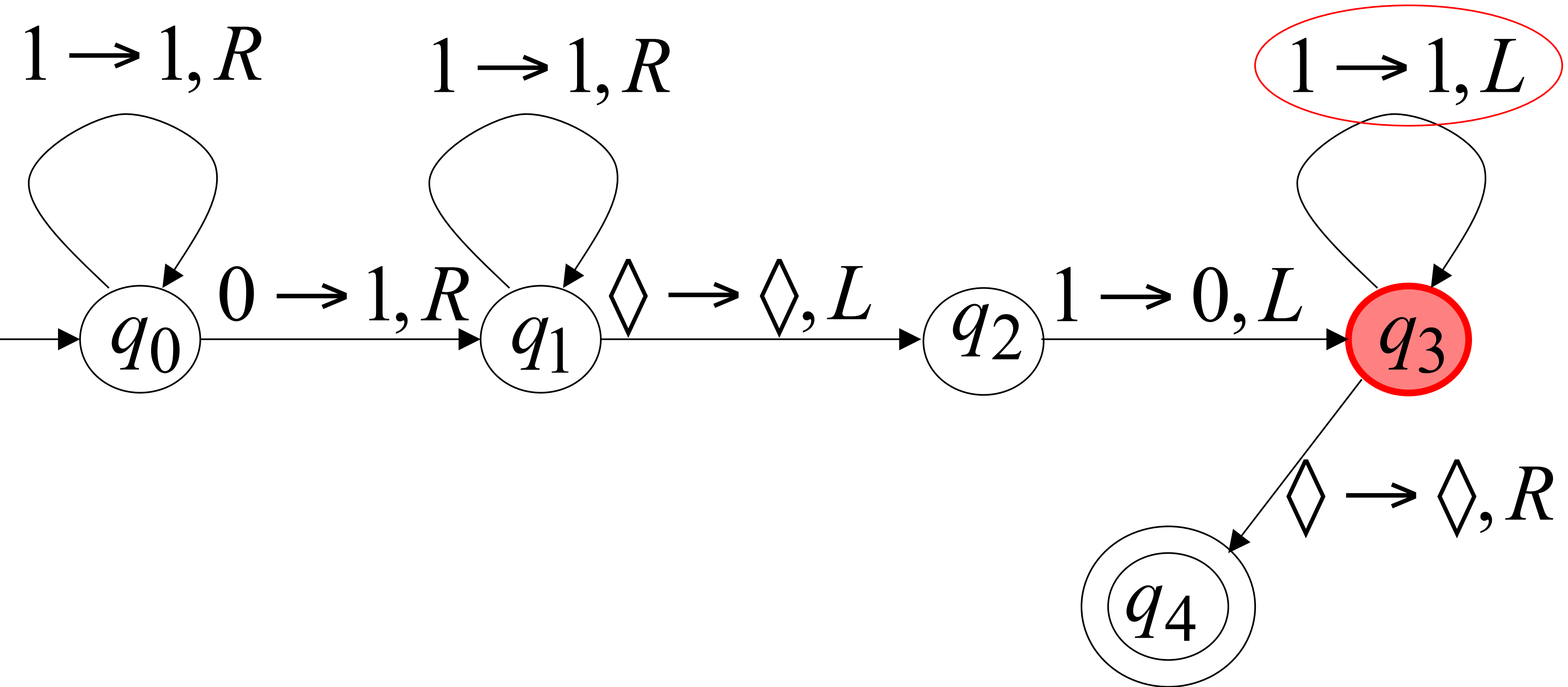
Time 7


◊
1
1
1
1
0
◊
q_3

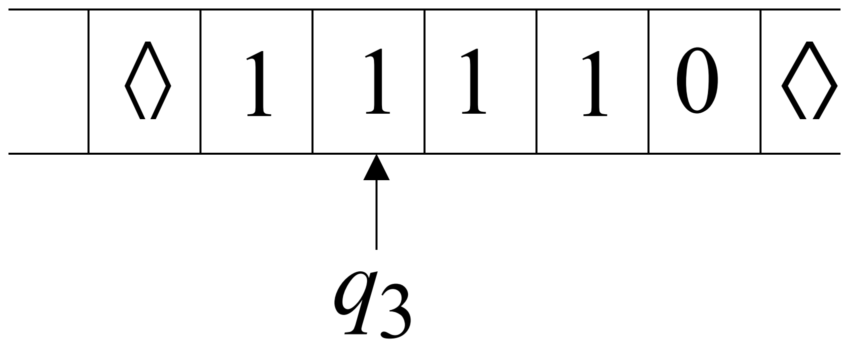

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4

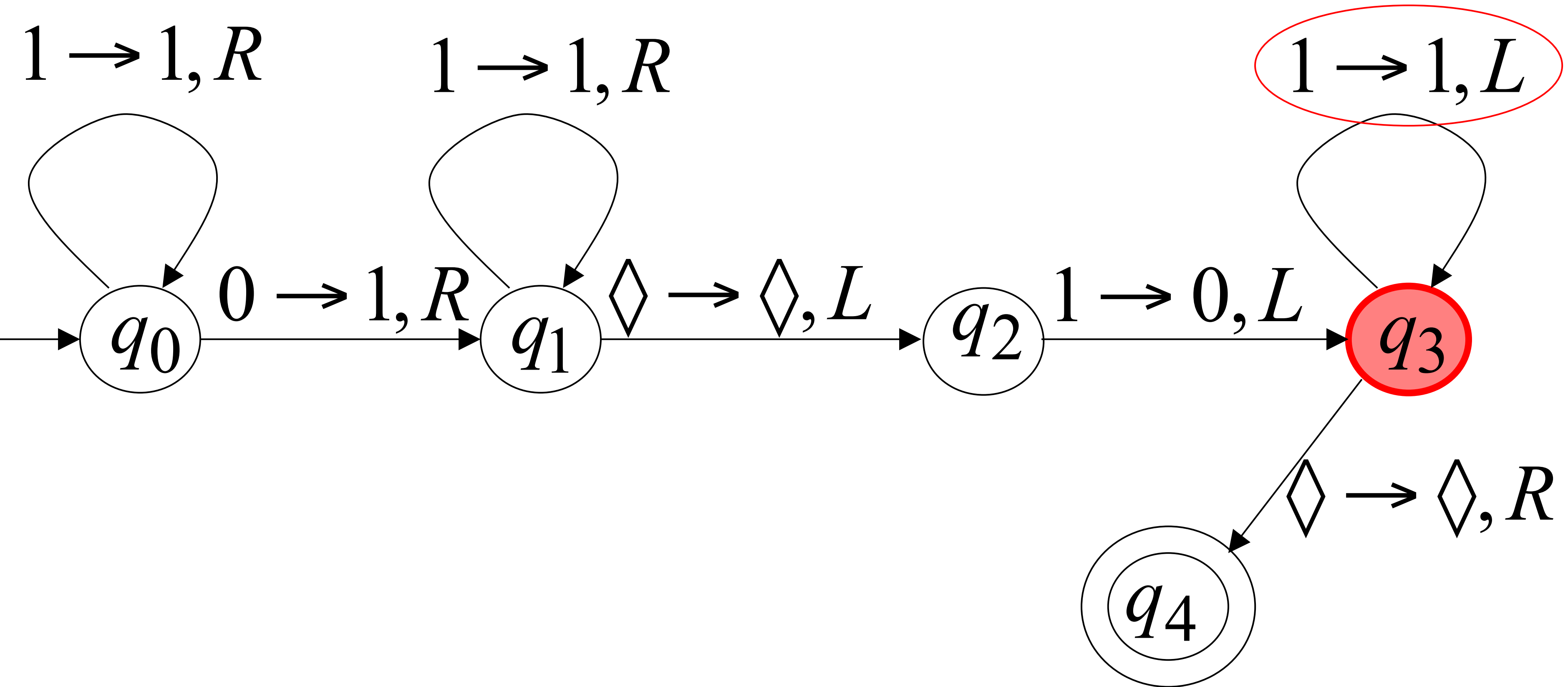
Time 8


◊
1
1
1
1
0
◊
q_3

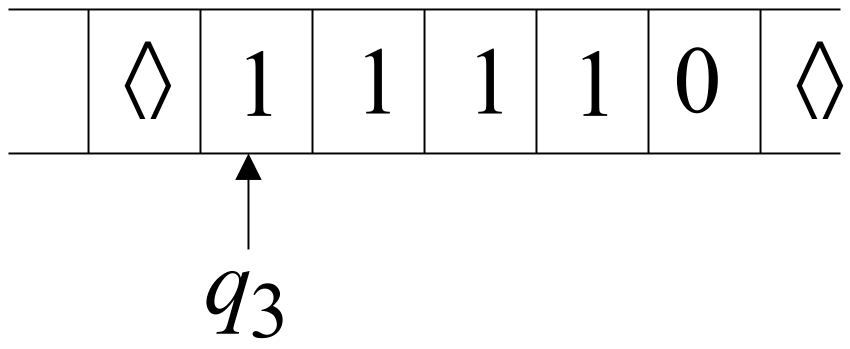

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
$\lozenge \rightarrow \lozenge, R$
q_0
q_1
q_2
q_3
q_4

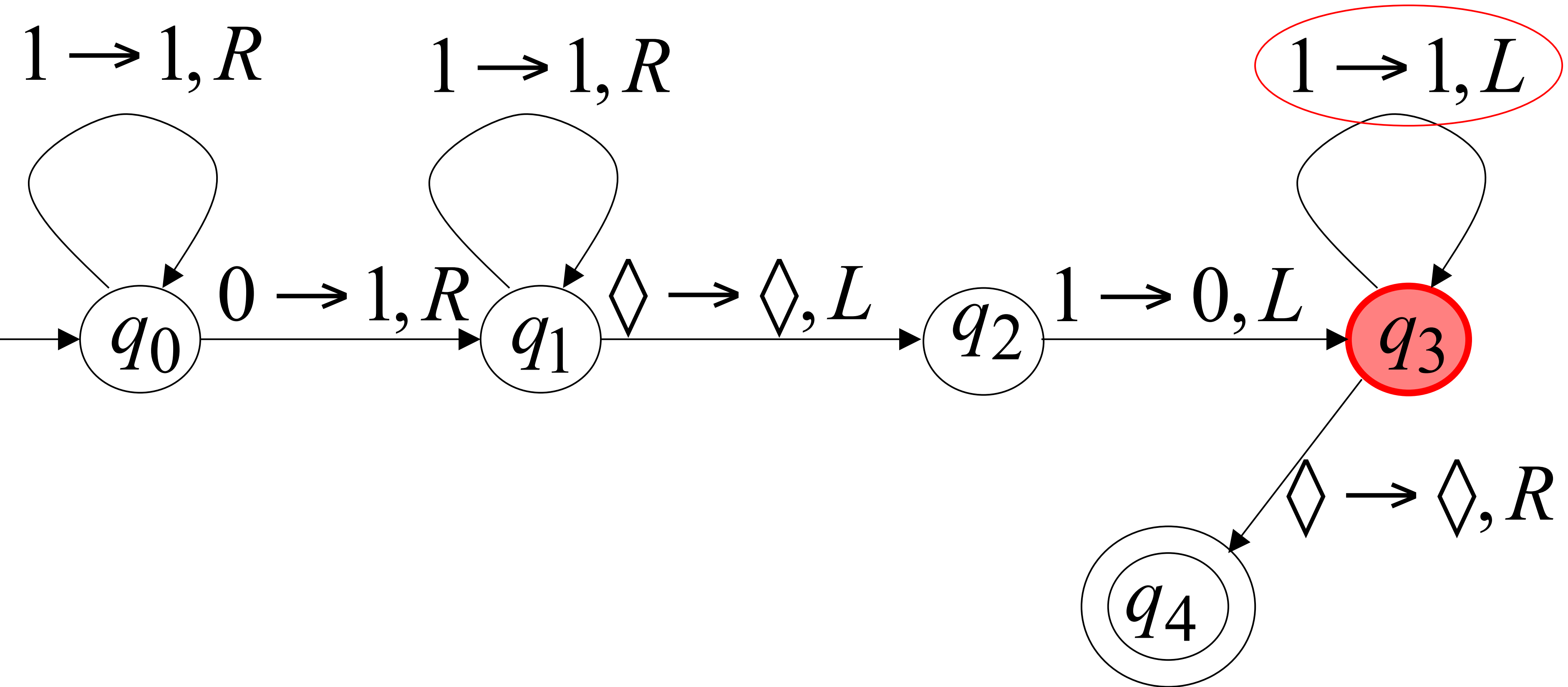
Time 9


◊
1
1
1
1
0
◊
q_3

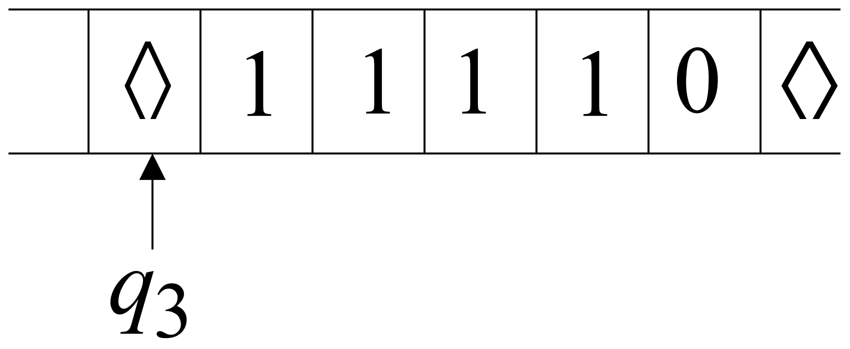

$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$◊ \rightarrow ◊, L$
$1 \rightarrow 0, L$
$◊ \rightarrow ◊, R$
q_0
q_1
q_2
q_3
q_4

Time 10

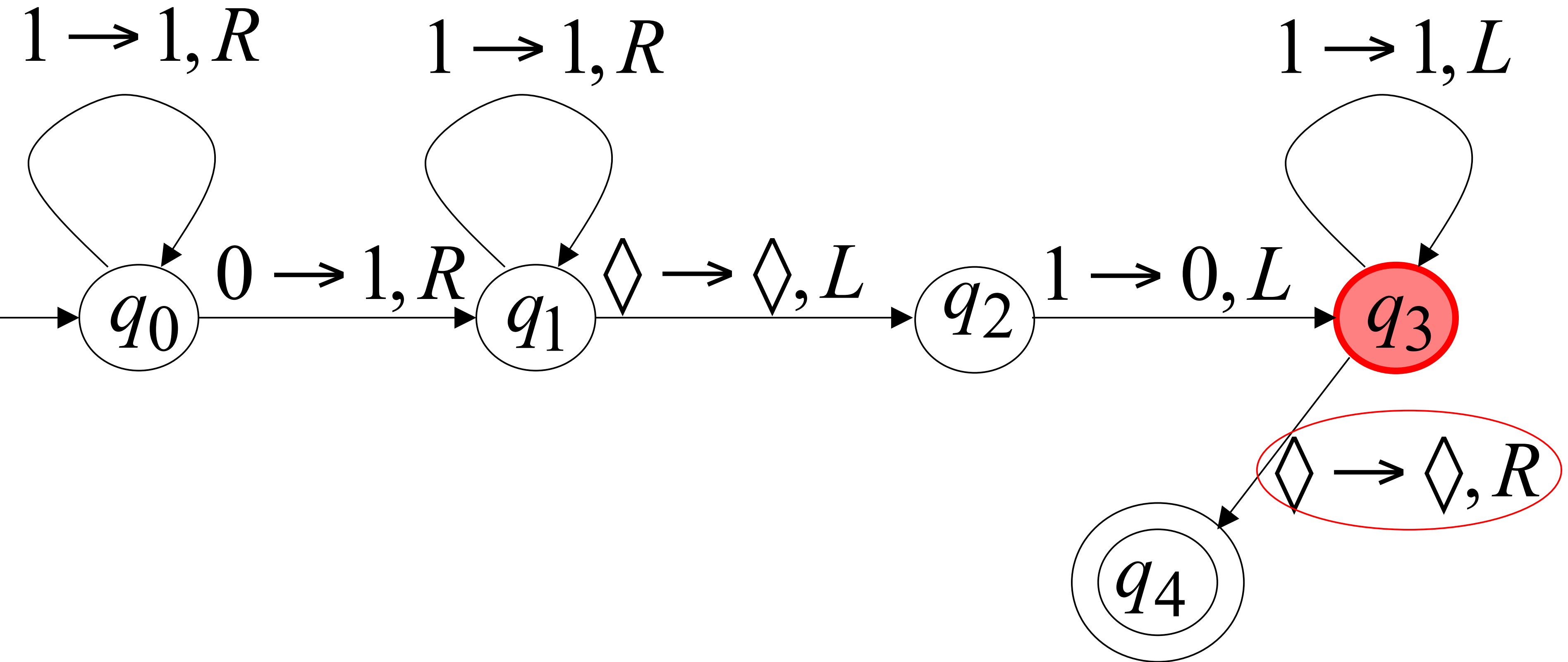

◊
1
1
1
1
0
◊
q_3


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

Time 11

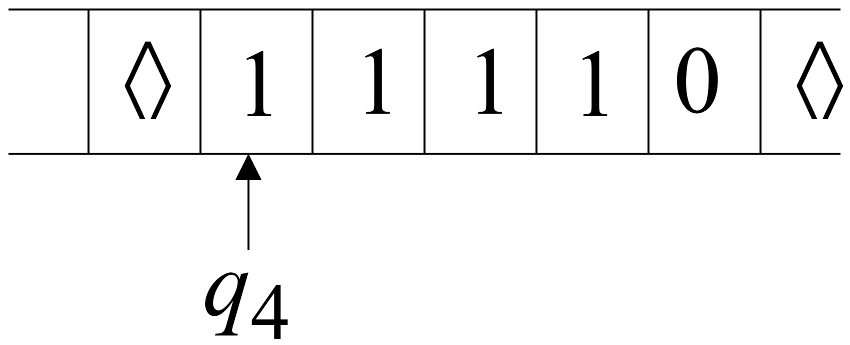


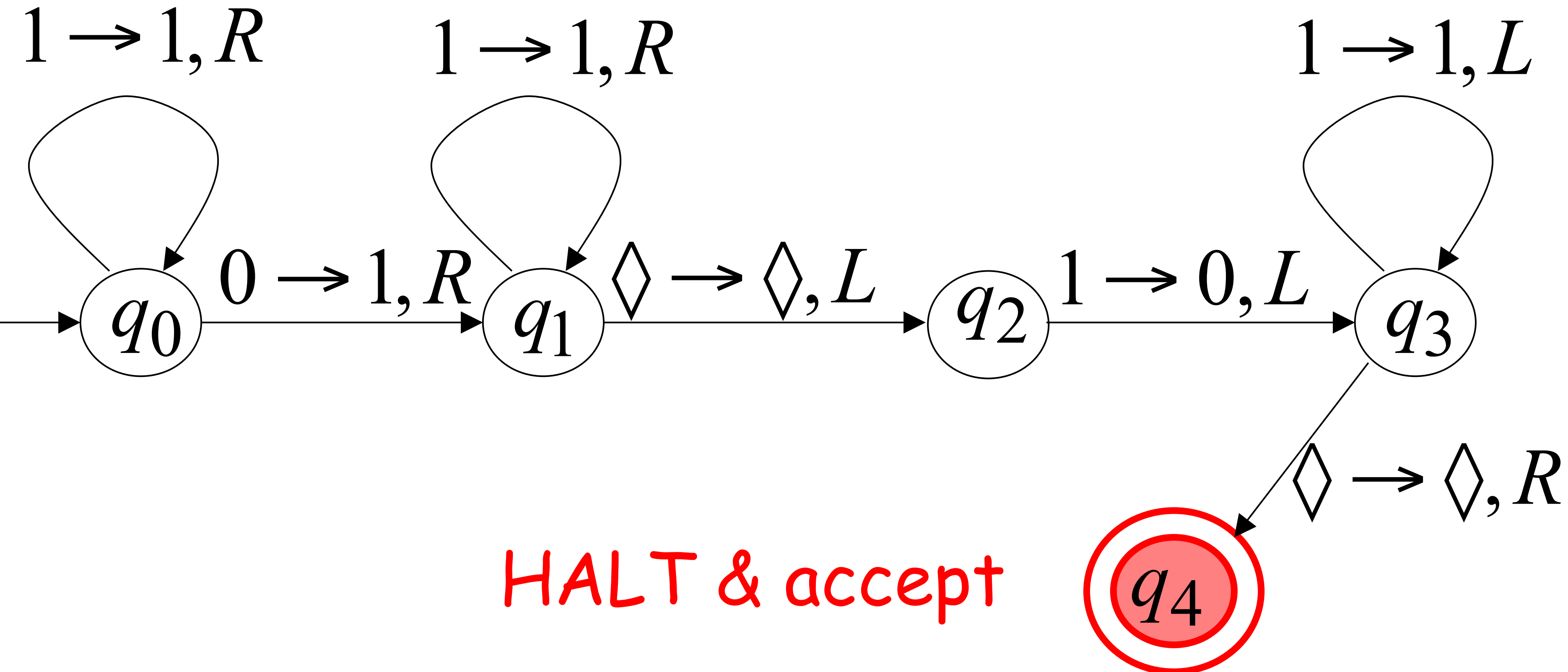
◊
1
1
1
1
0
◊
q_3



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

Time 12





HALT & accept

Another Example

The function $f(x) = 2x$ is computable

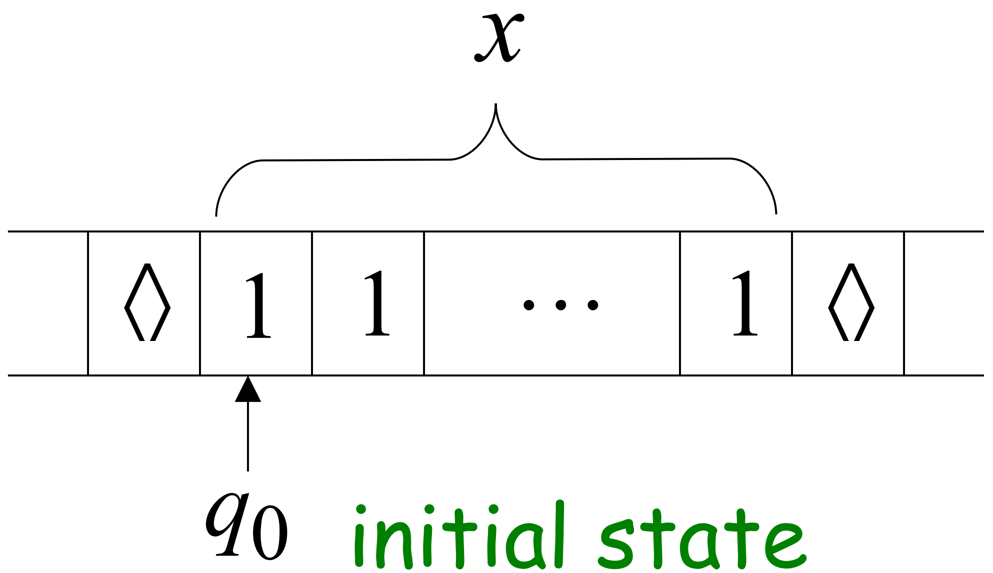
x is integer

Turing Machine:

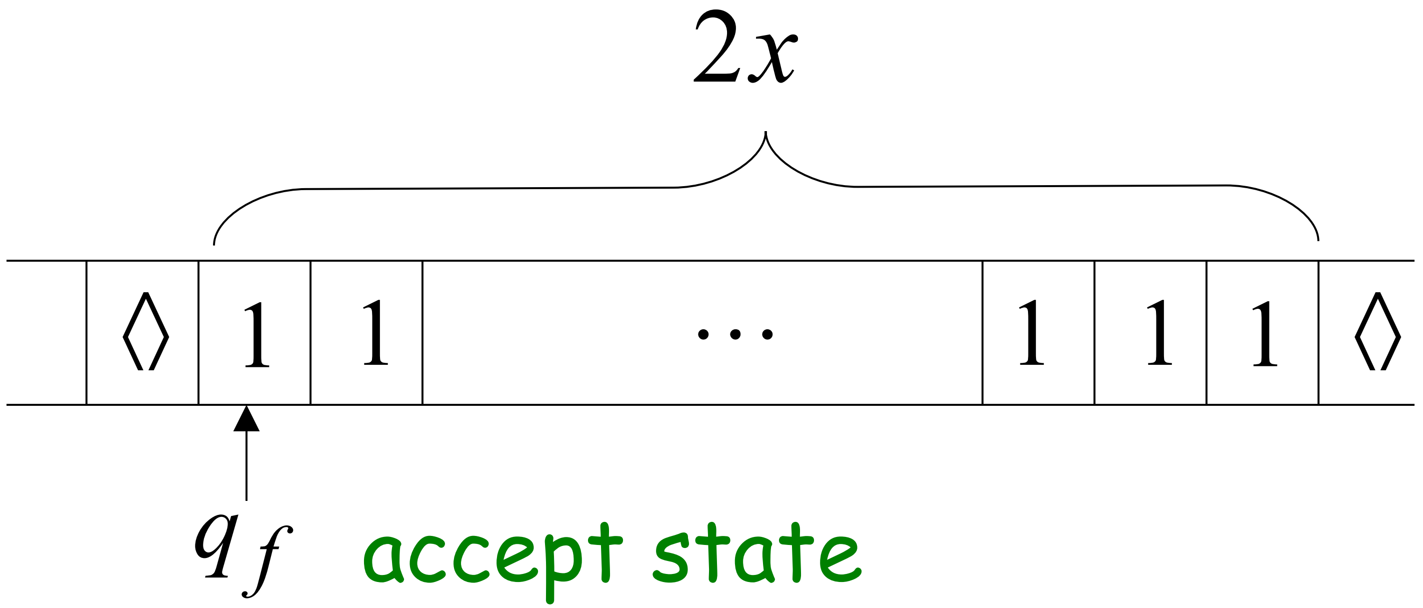
Input string: $x = 11$ unary

Output string: $xx = 1111$ unary

Start



Finish

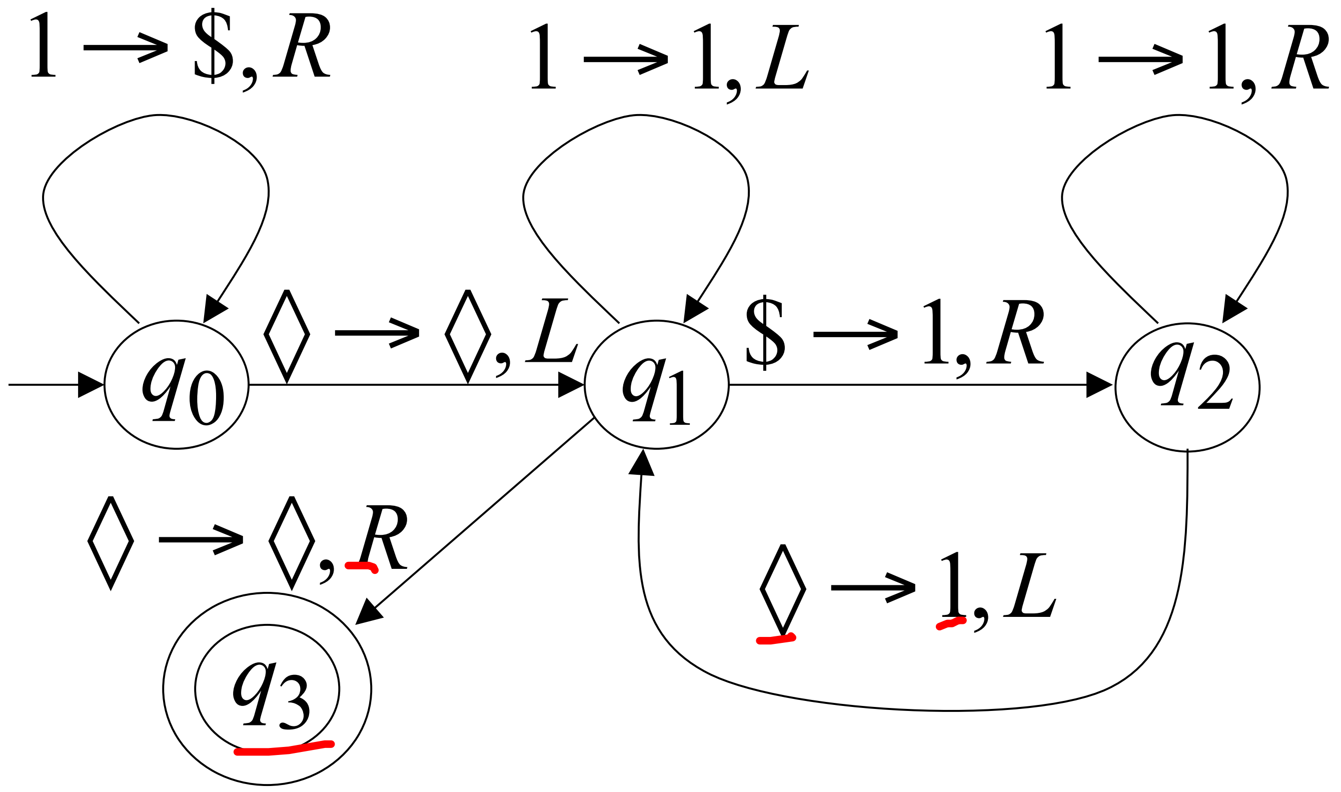


Turing Machine Pseudocode for $f(x) = 2x$

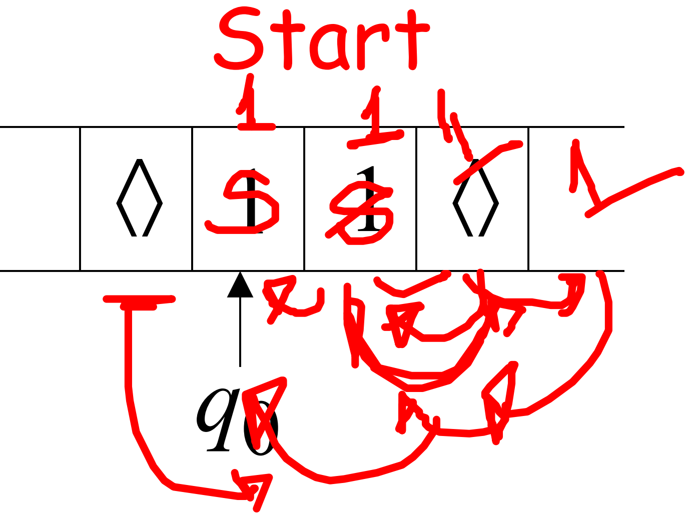
[illegible]

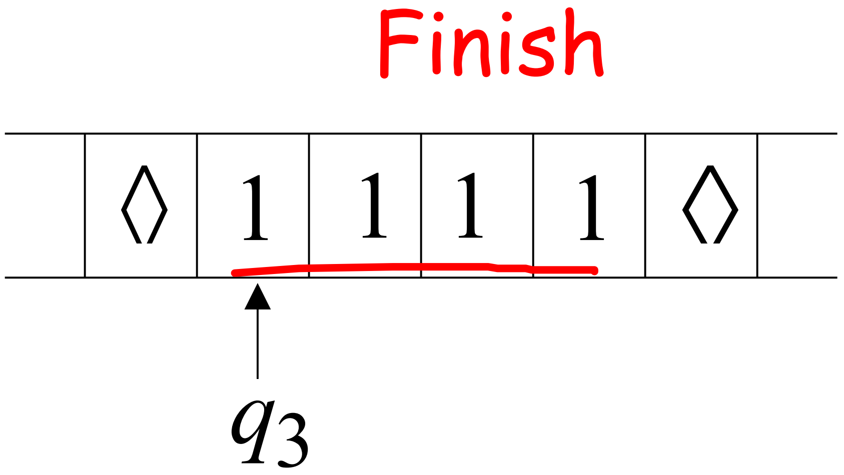
```
• Replace every 1 with $

• Repeat:
        • Find rightmost $, replace it with 1

        • Go to right end, insert 1

  Until no more $ remain
```

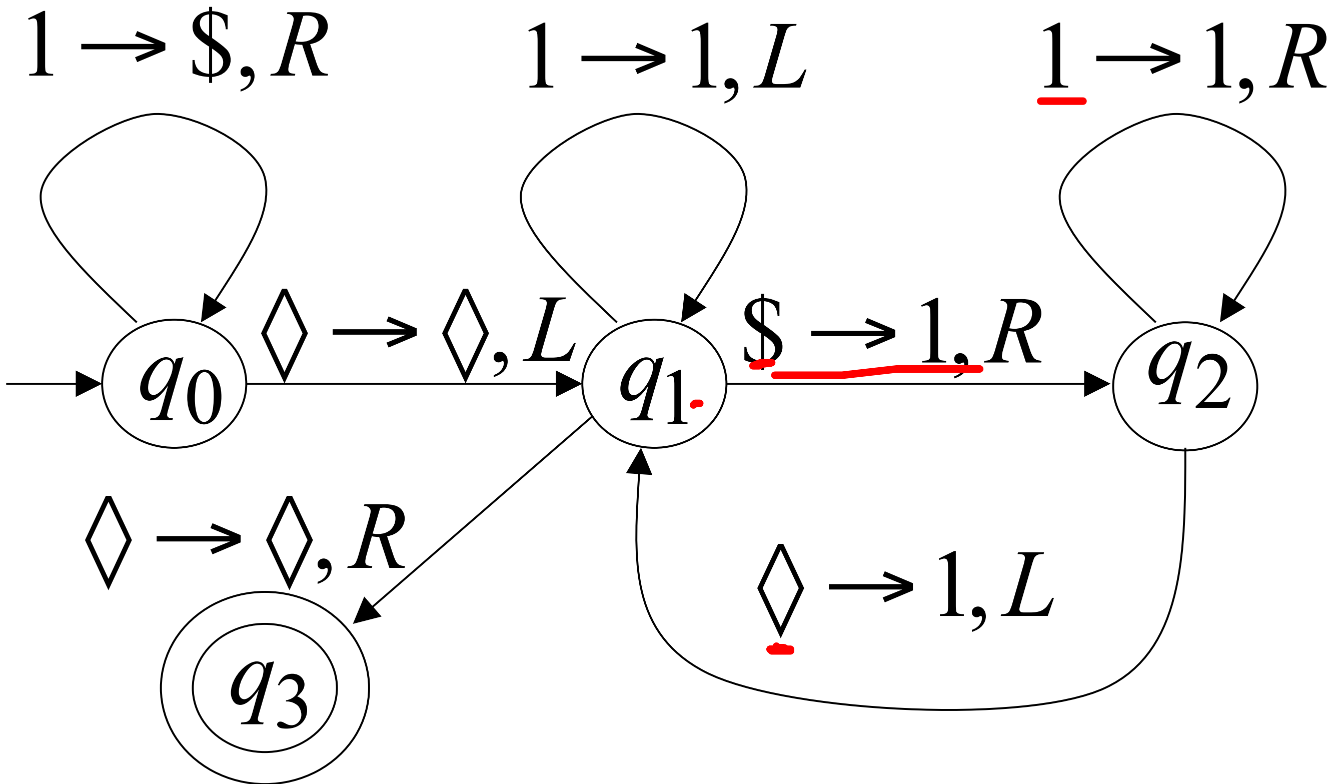
Turing Machine for $f(x) = 2x$



Example


Start
◊
~~1~~ $
~~1~~ $
◊
1
1
1
1
q_0


Finish
◊
1
1
1
1
◊
q_3


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

Another Example

The function is computable

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

Input: $\underline{x}\,\underset{\downarrow}{0}\,\underline{y}$

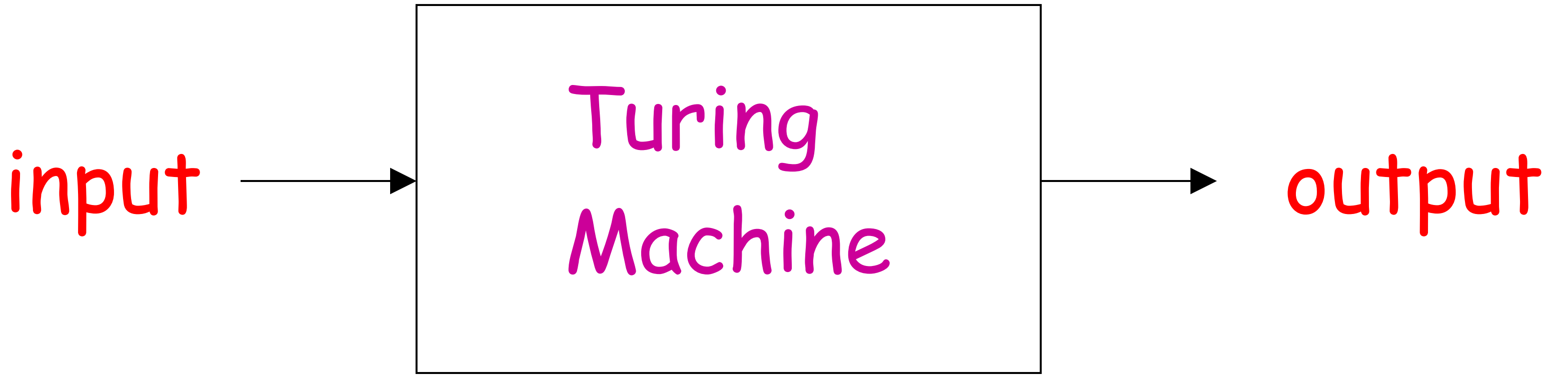
Output: $\underline{1}$ or 0

Turing Machine Pseudocode:

```
• Repeat

      Match a 1 from $x$ with a 1 from $y$

   Until all of $x$ or $y$ is matched

• If a 1 from $x$ is not matched
          erase tape, write 1        $(x > y)$
    else
          erase tape, write 0        $(x \leq y)$
```

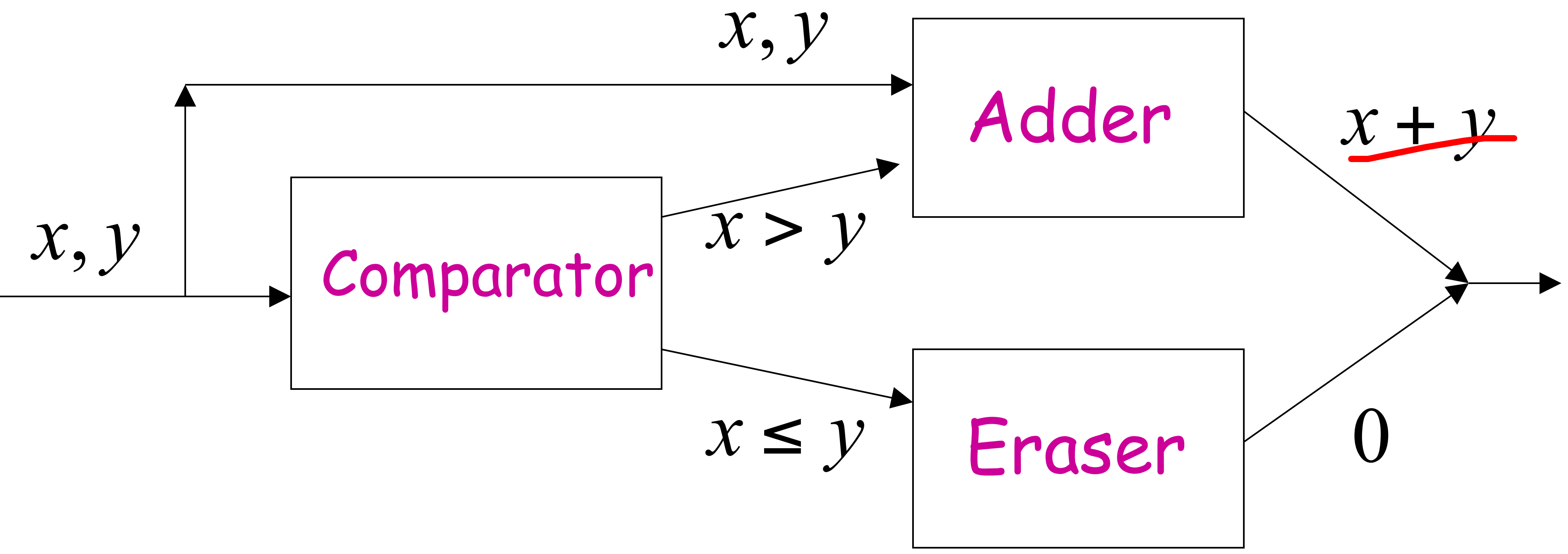
Combining Turing Machines

Block Diagram



Example:

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



x, y
x, y
Comparator
x > y
x ≤ y
Adder
Eraser
x + y
0