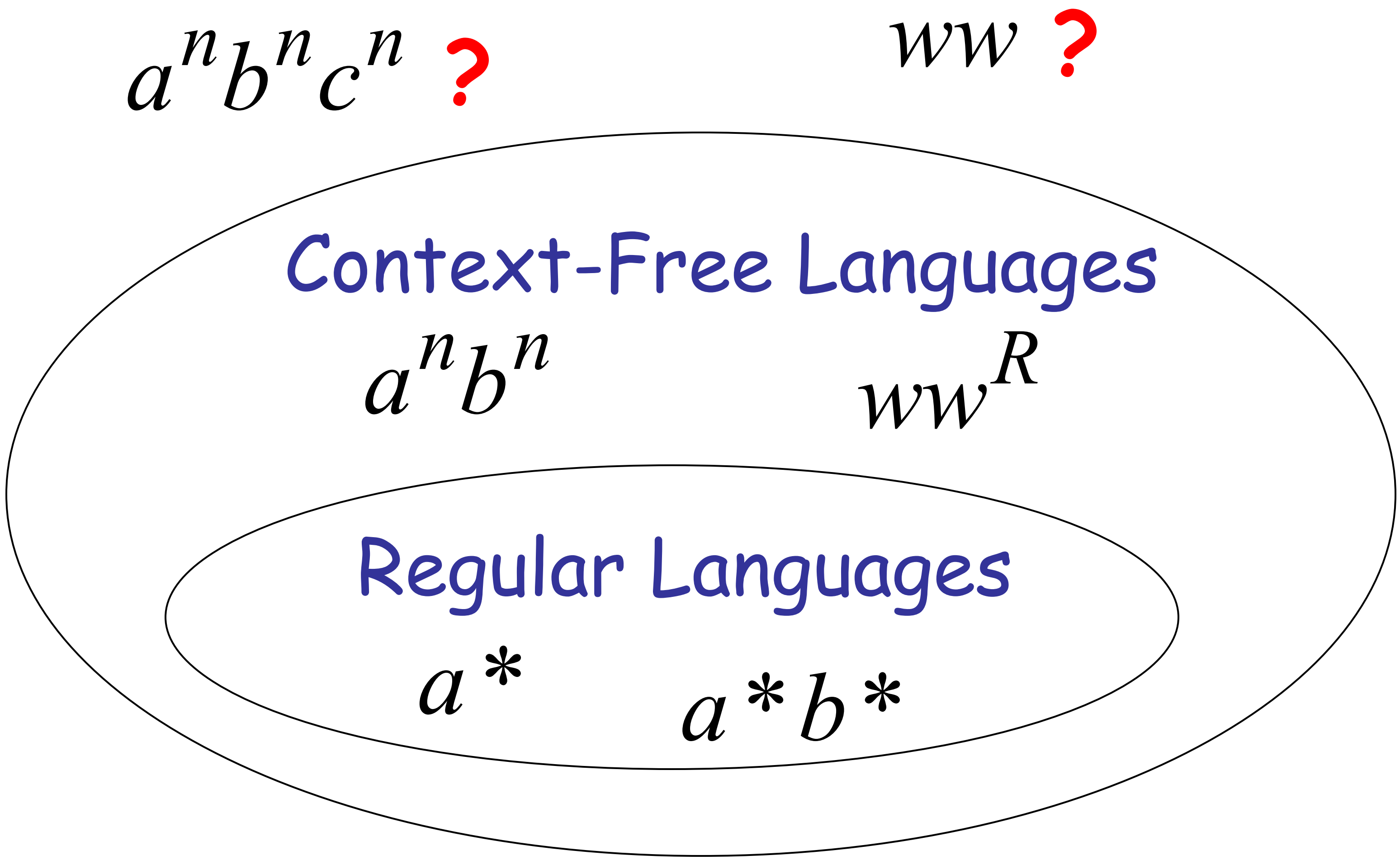
Turing Machines

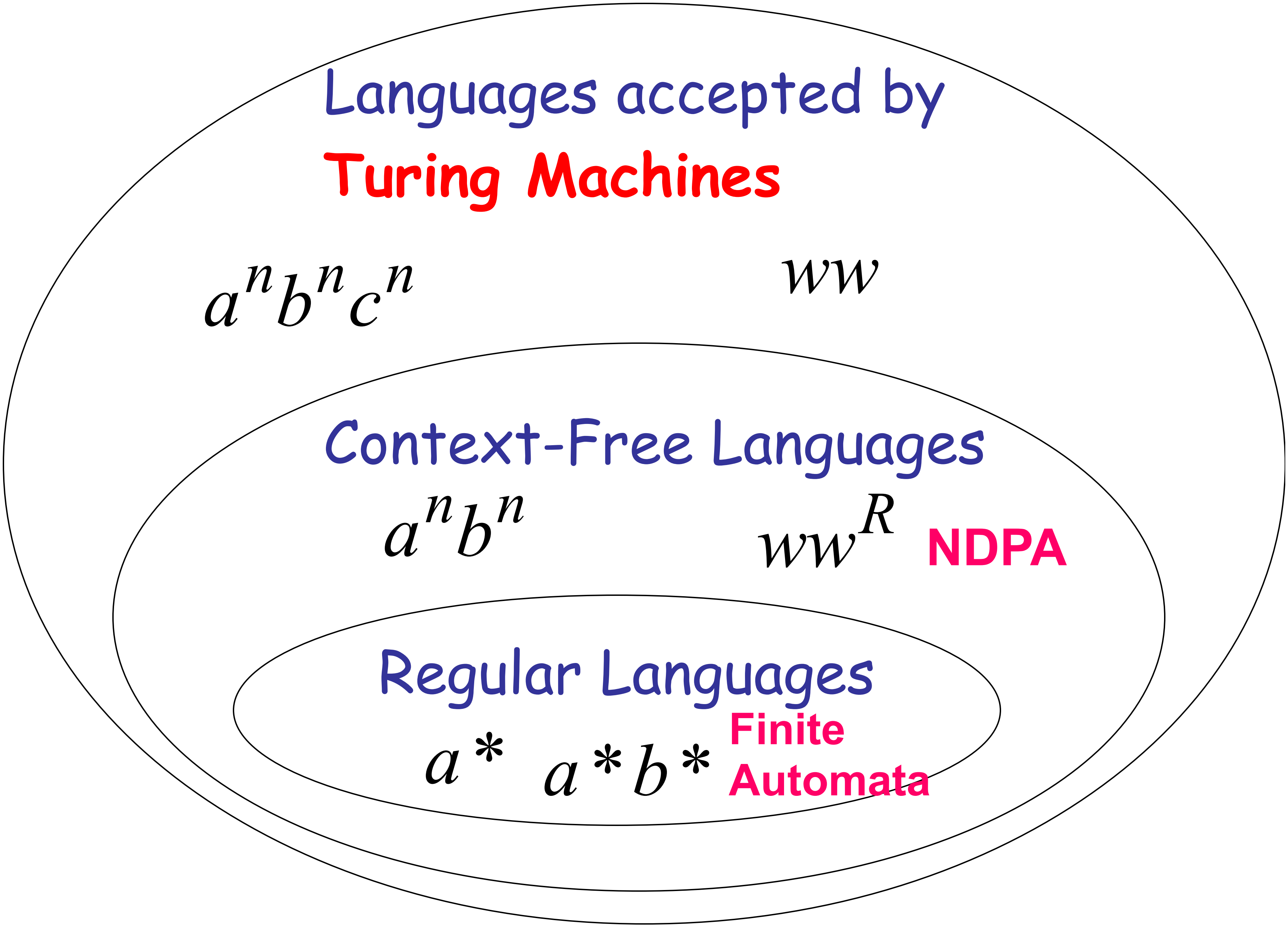
Invented by Alan Turing in 1936.

A simple mathematical model of a general purpose computer.

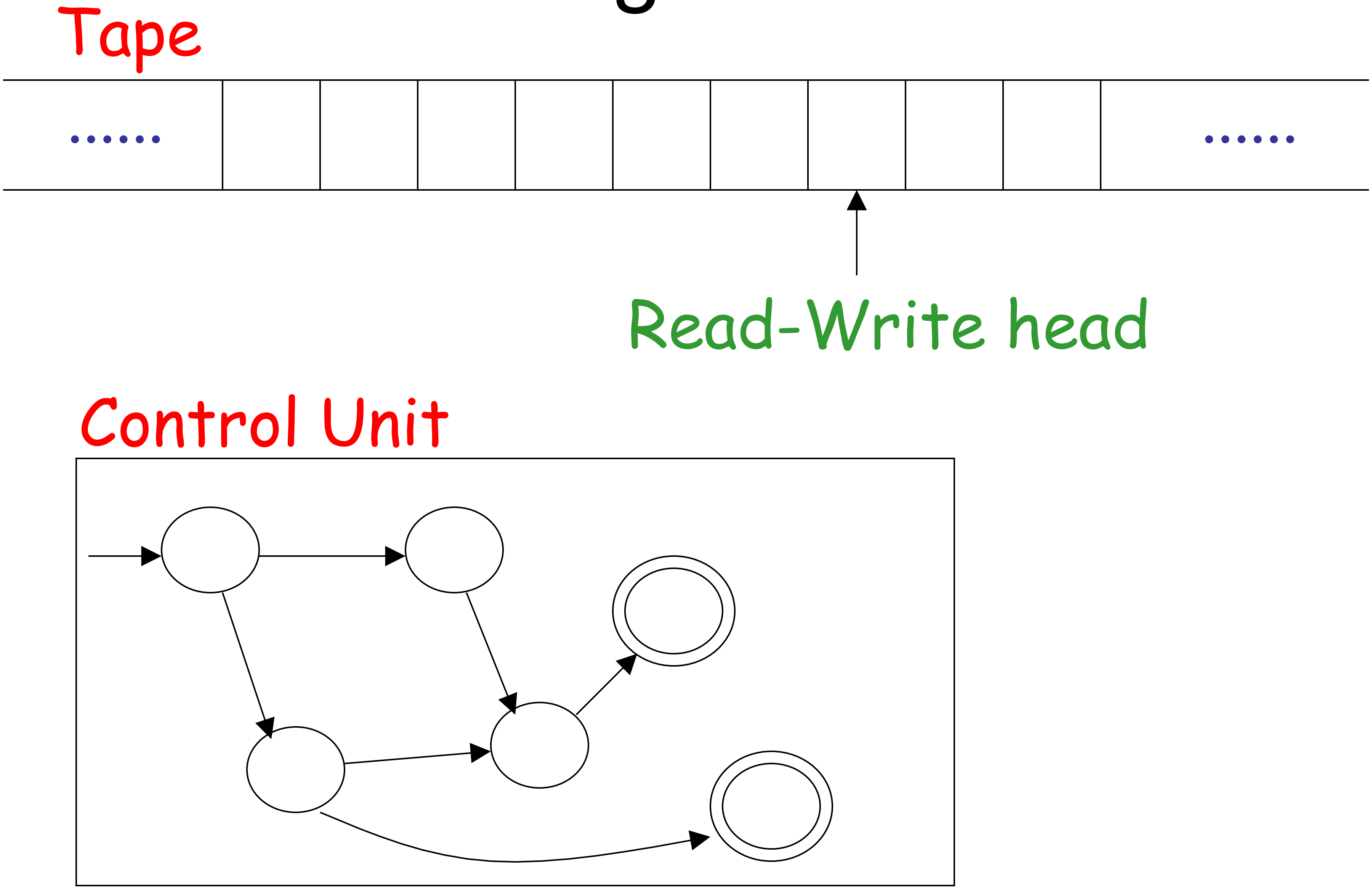
It is capable of performing any calculation which can be performed by any computing machine.

The Language Hierarchy



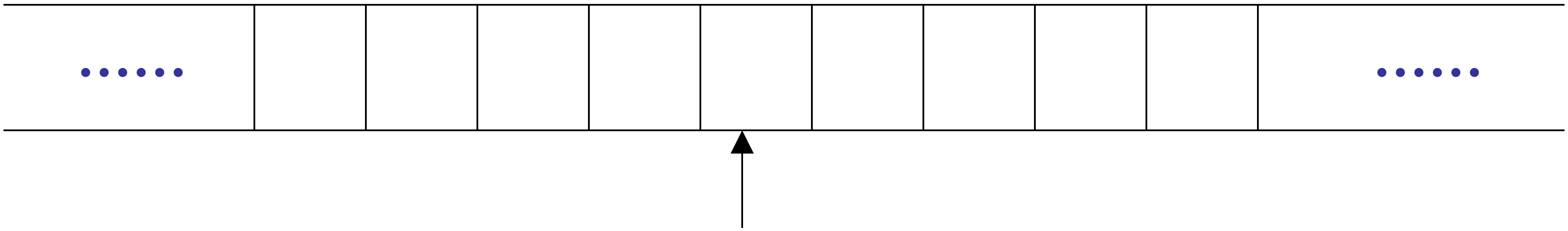

Languages accepted by
Turing Machines
$a^n b^n c^n$
ww
Context-Free Languages
$a^n b^n$
ww^R
NDPA
Regular Languages
a^*
$a^* b^*$
Finite
Automata

A Turing Machine



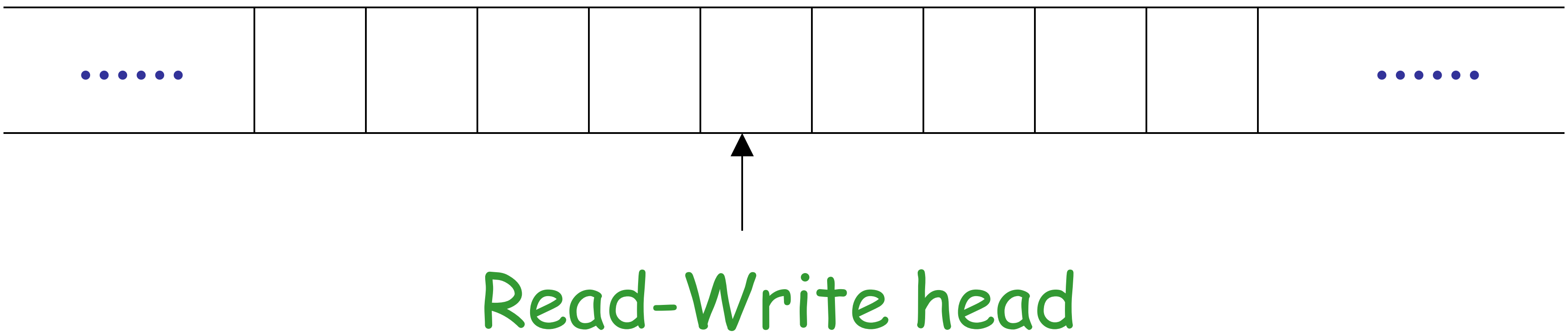
The Tape

No boundaries -- infinite length



Read-Write head

The head moves Left or Right

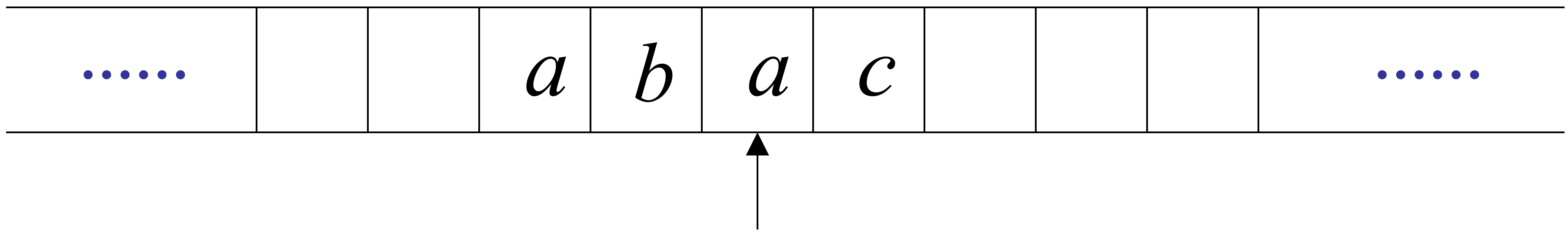


The head at each time step:

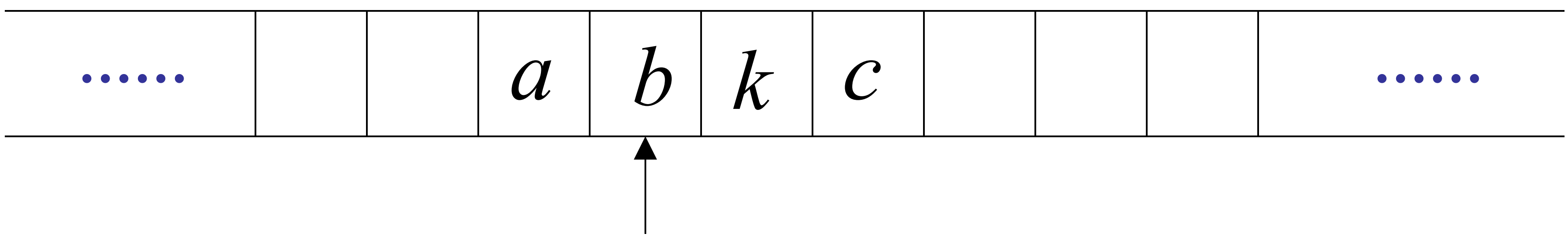
1. Reads a symbol
2. Writes a symbol
3. Moves Left or Right

Example:

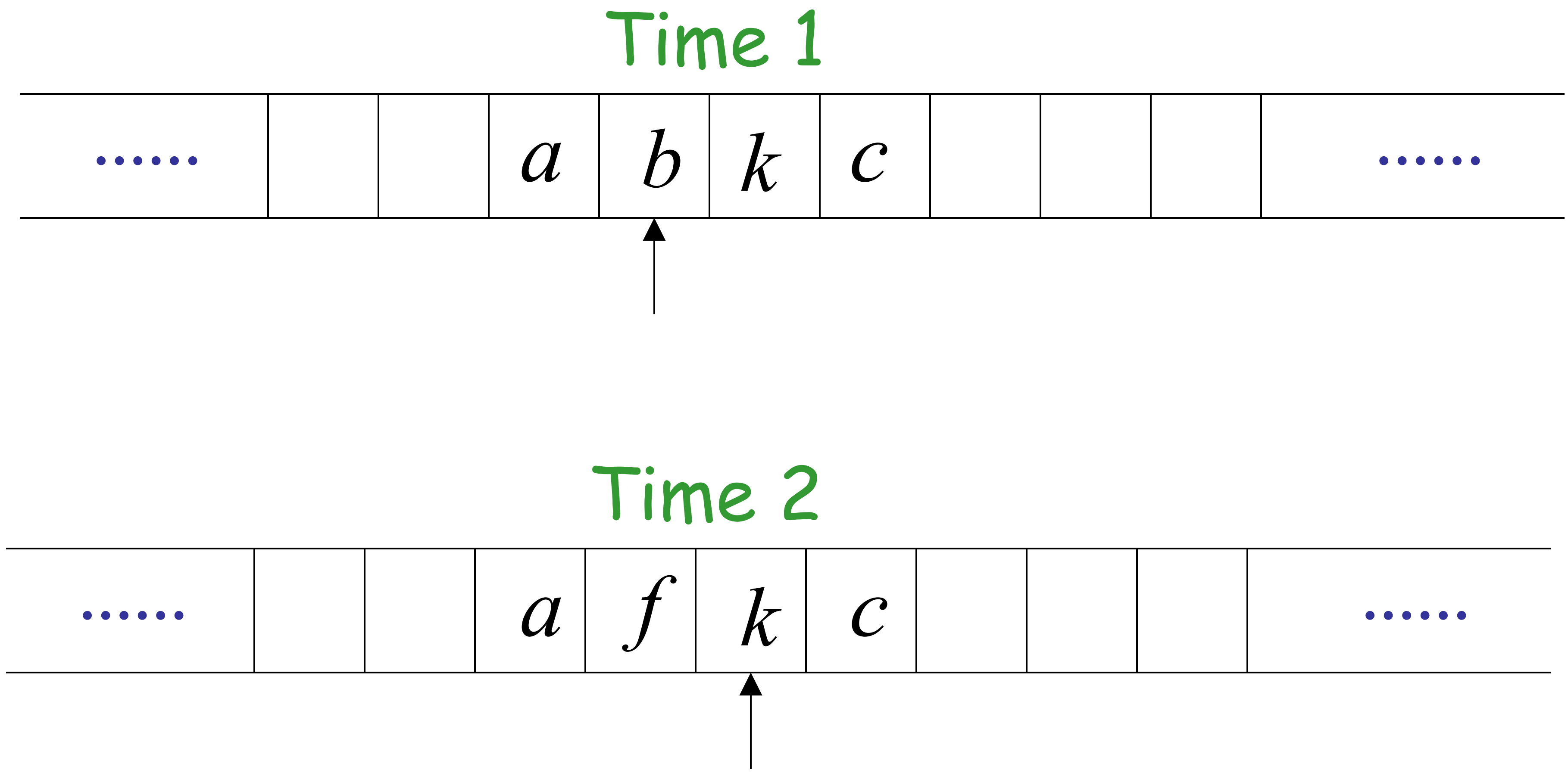
Time 0



Time 1

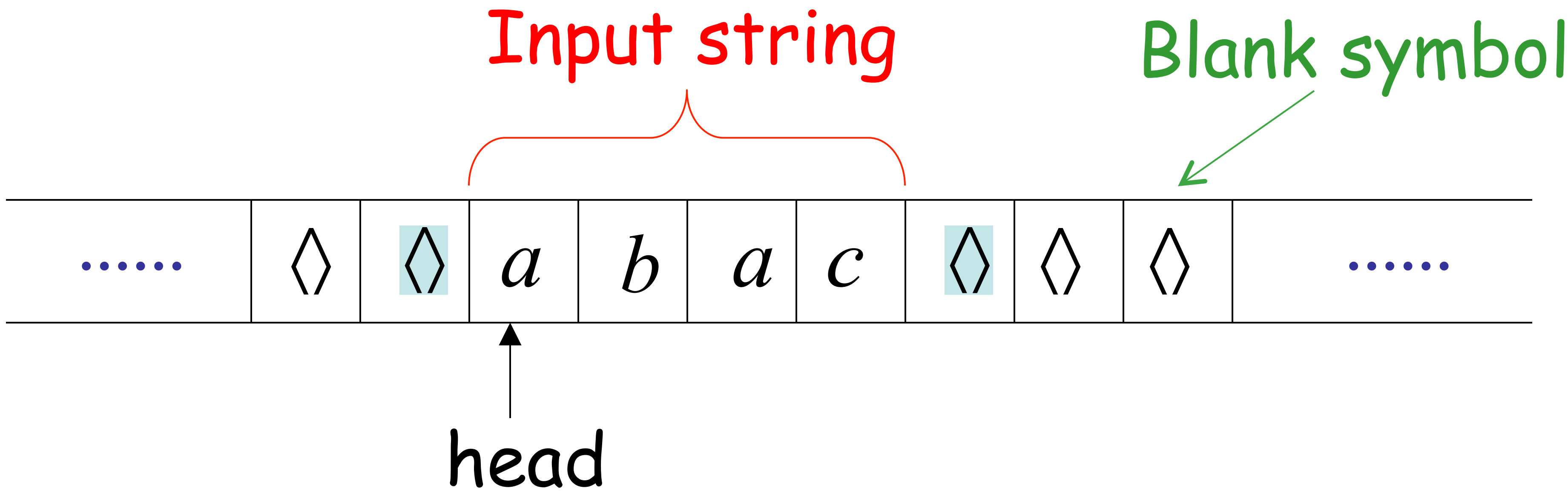


1. Reads a
2. Writes k
3. Moves Left



1. Reads b
2. Writes f
3. Moves Right

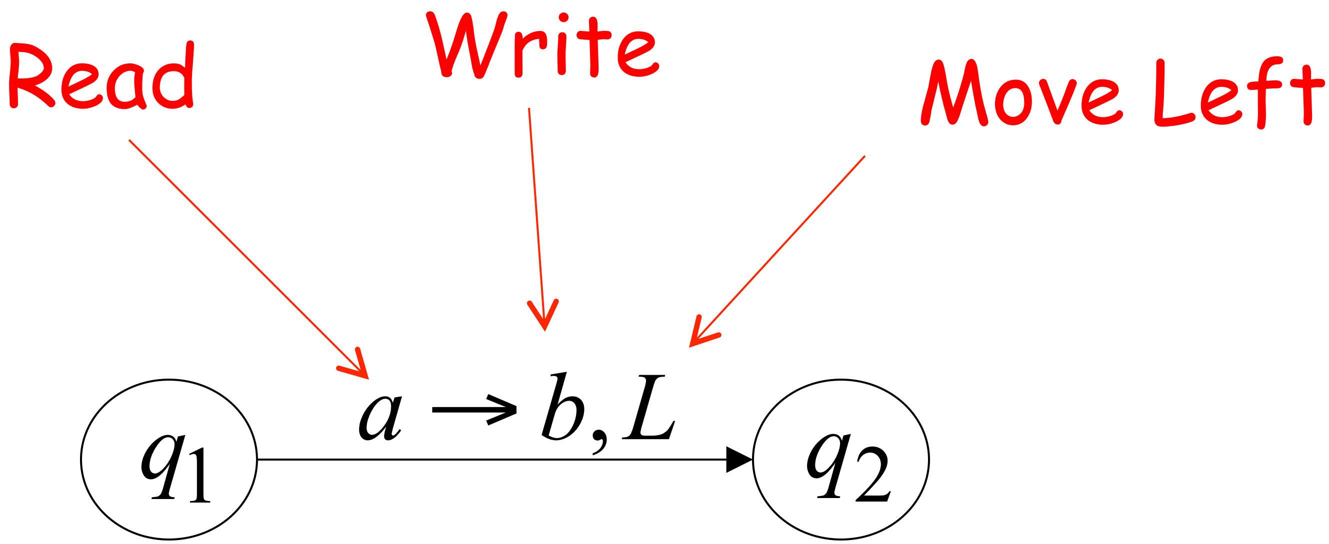
The Input String

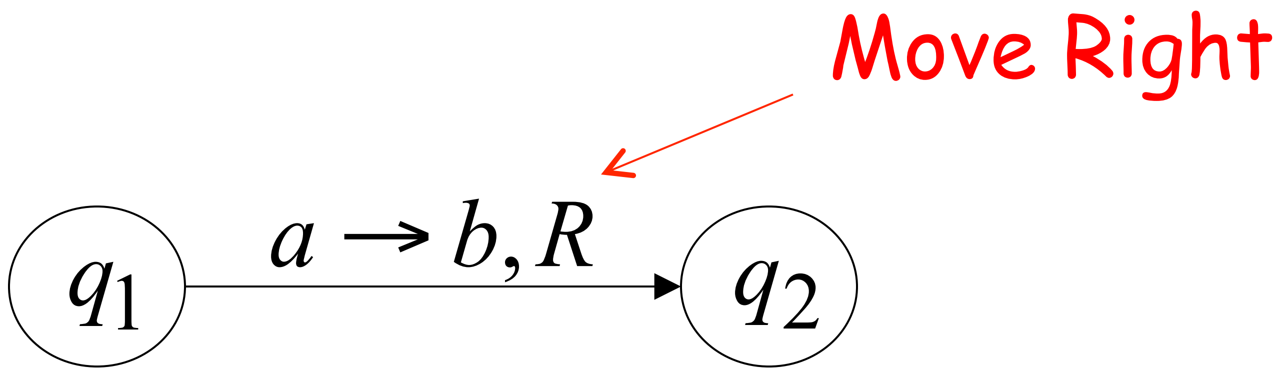


Head starts at the leftmost position of the input string

◊ **Are treated as left and right brackets for the input written on the tape.**

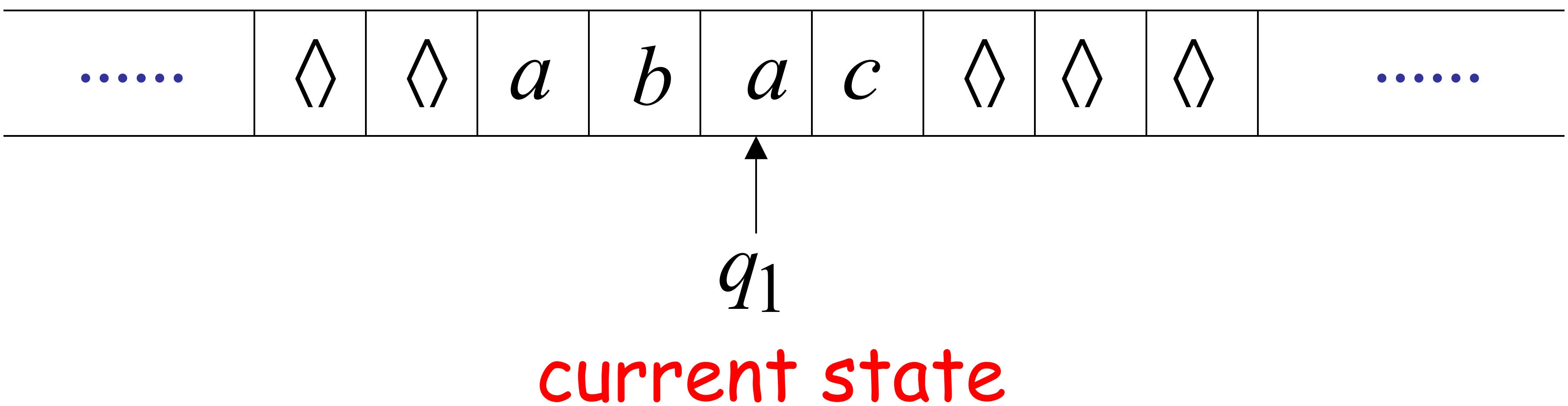
States & Transitions

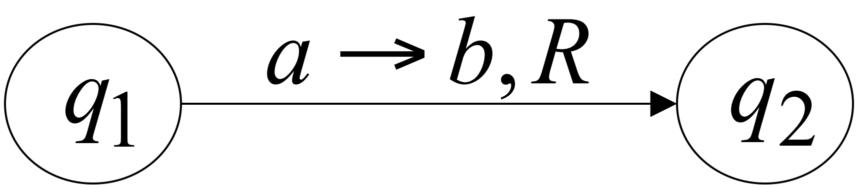




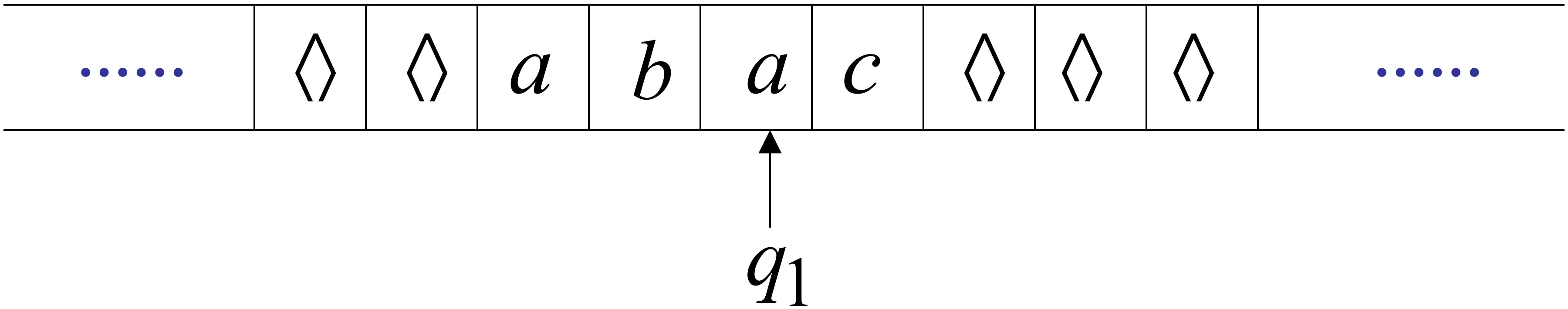
Example:

Time 1

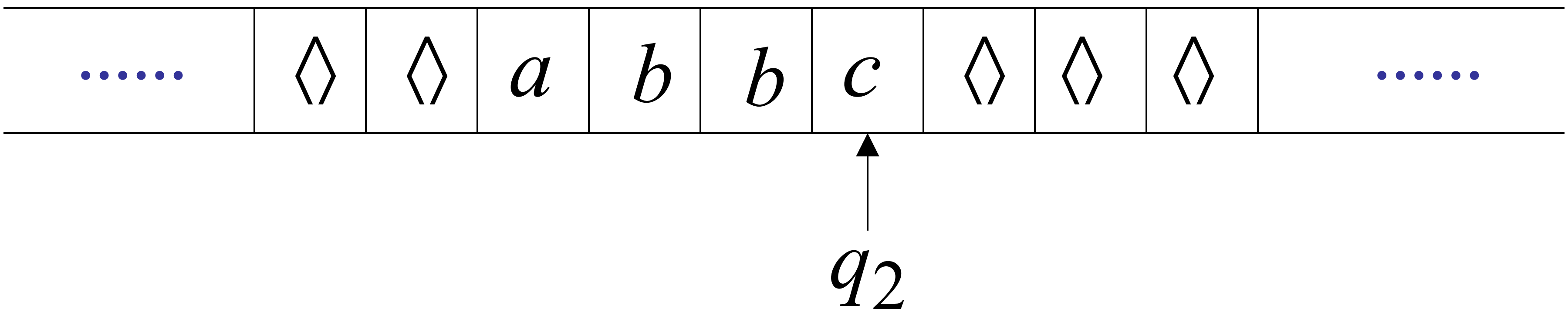


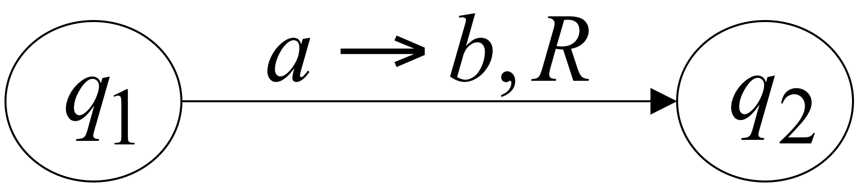


Time 1



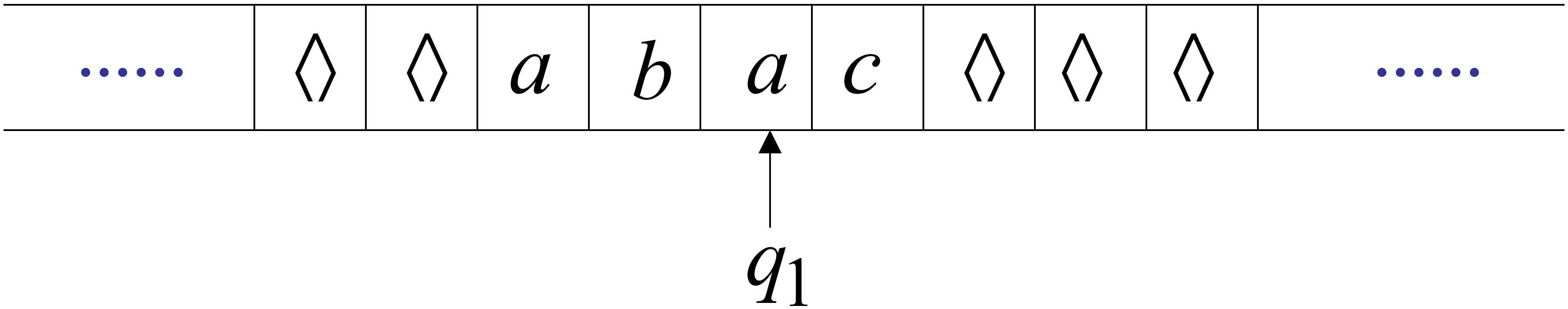
Time 2



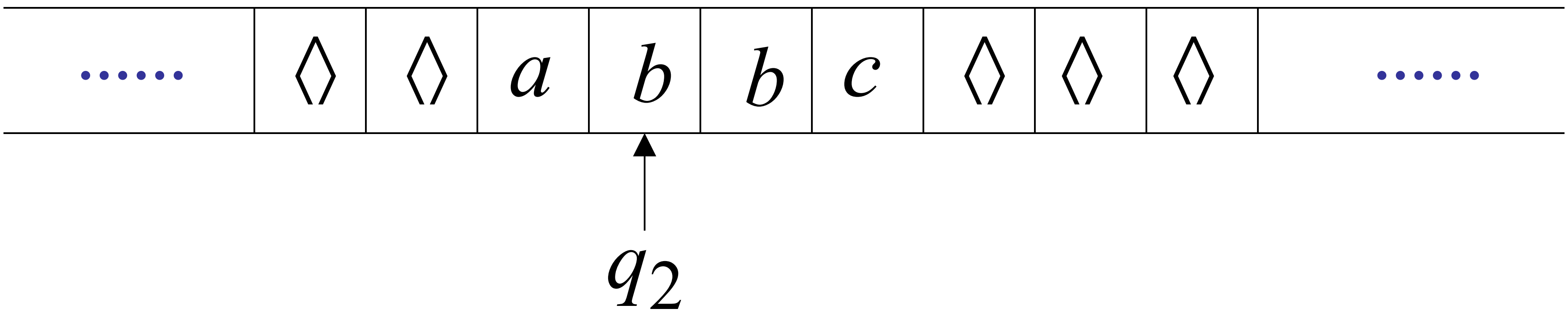


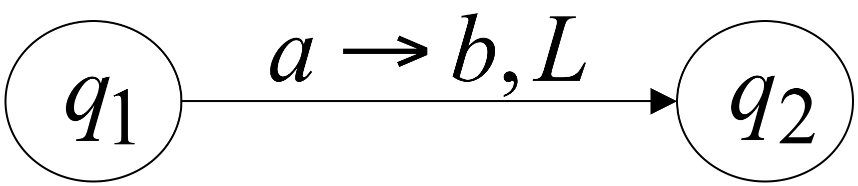
Example:

Time 1

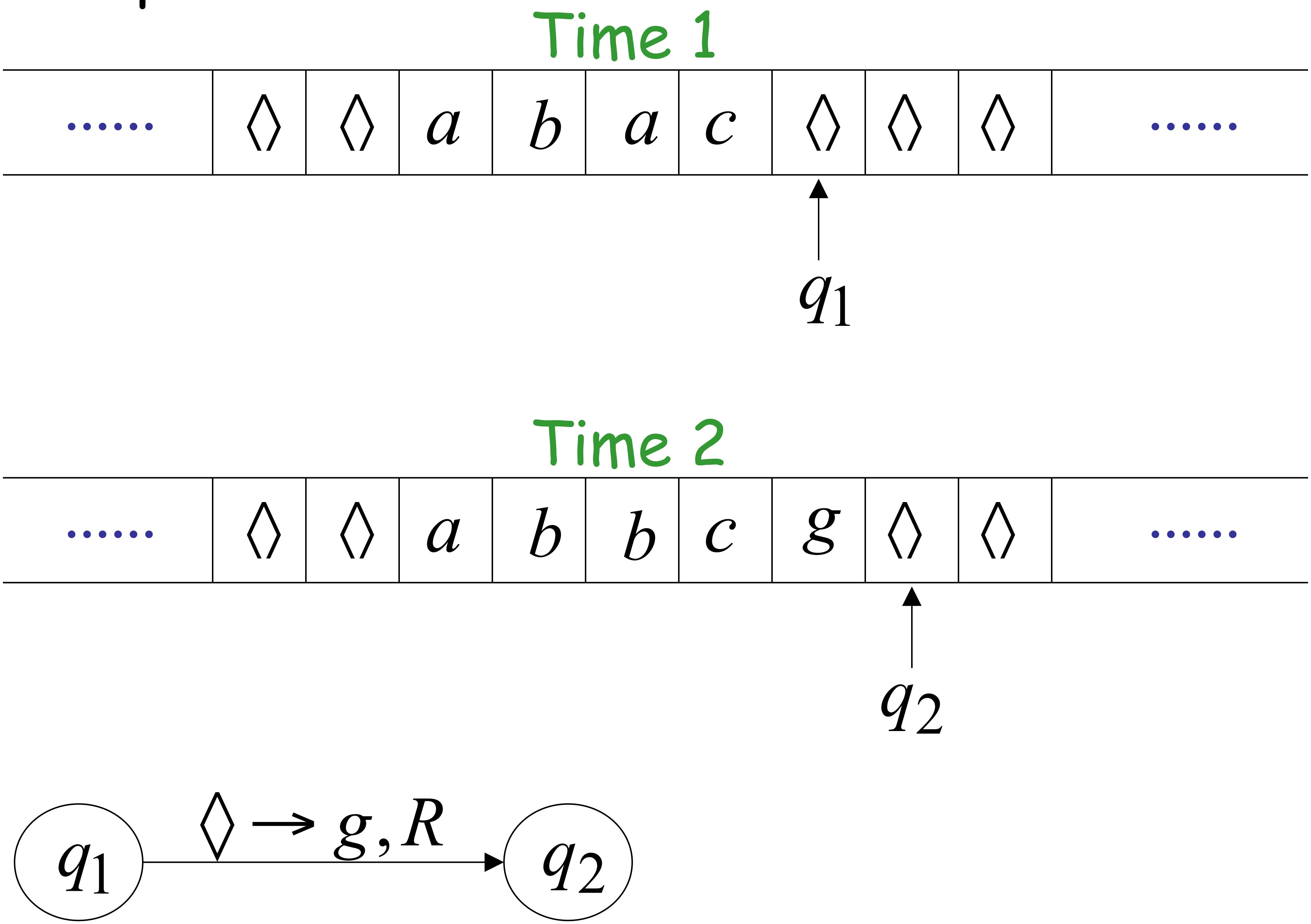


Time 2





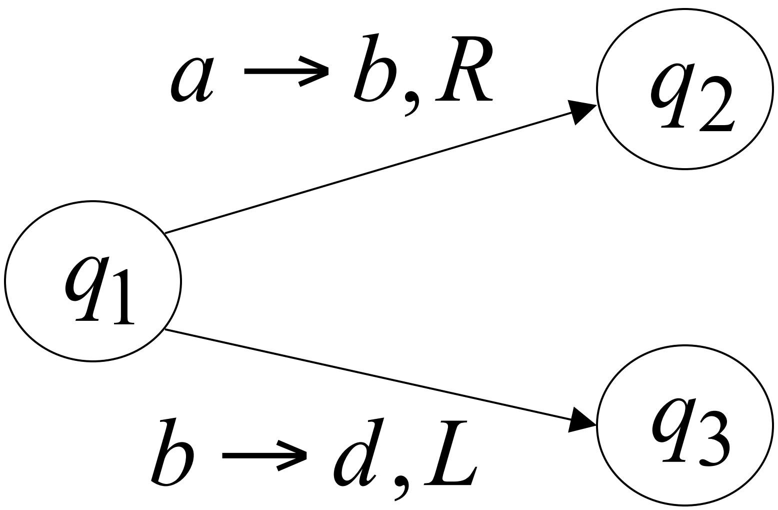
Example:



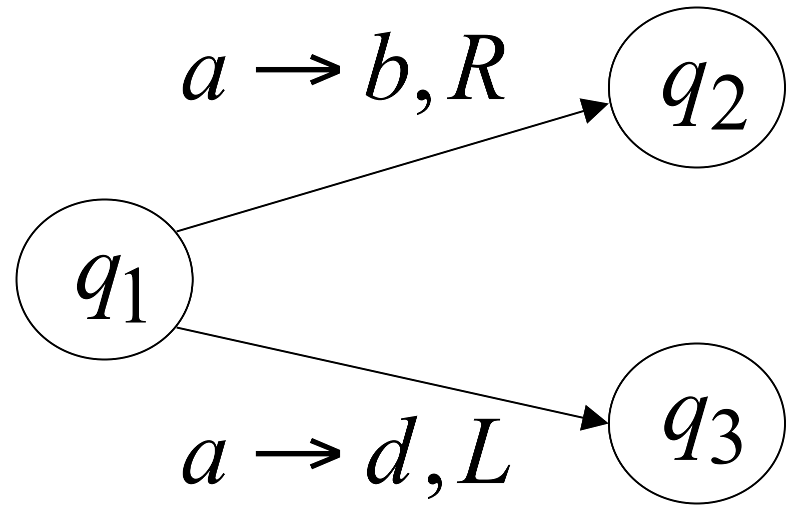
Determinism

Turing Machines are deterministic

Allowed



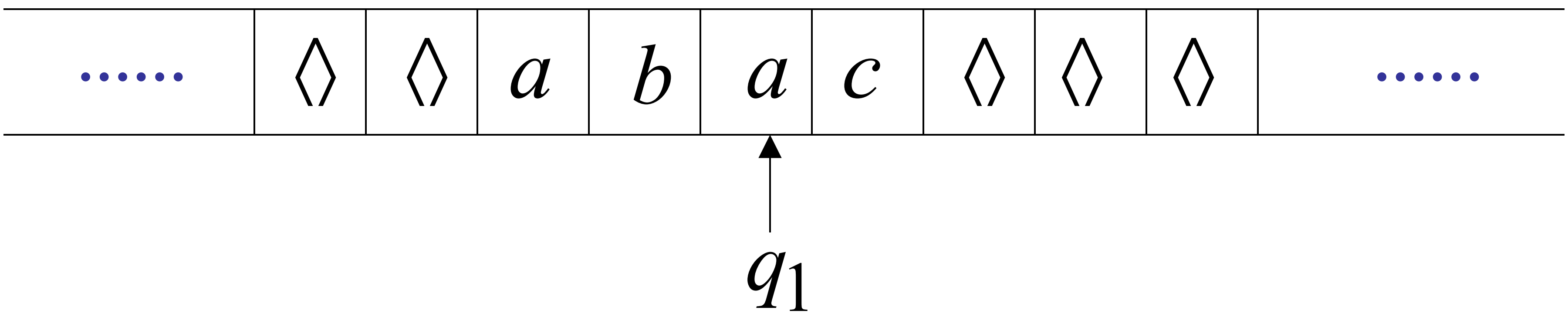
Not Allowed

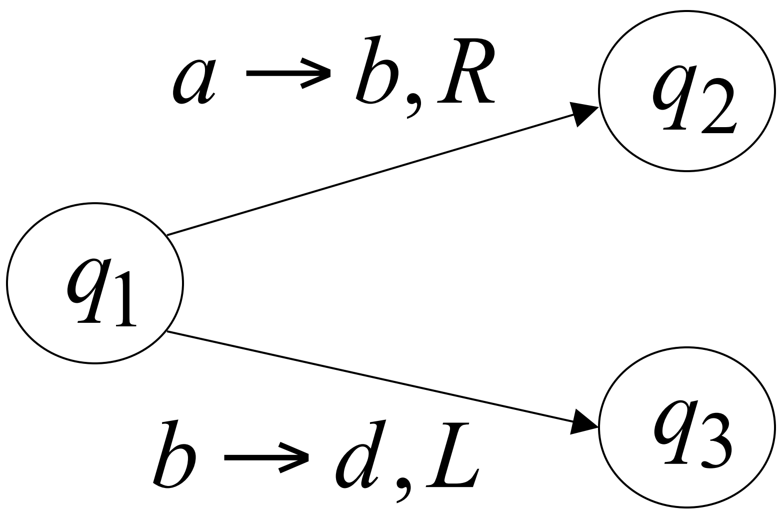


No lambda transitions allowed

Partial Transition Function

Example:





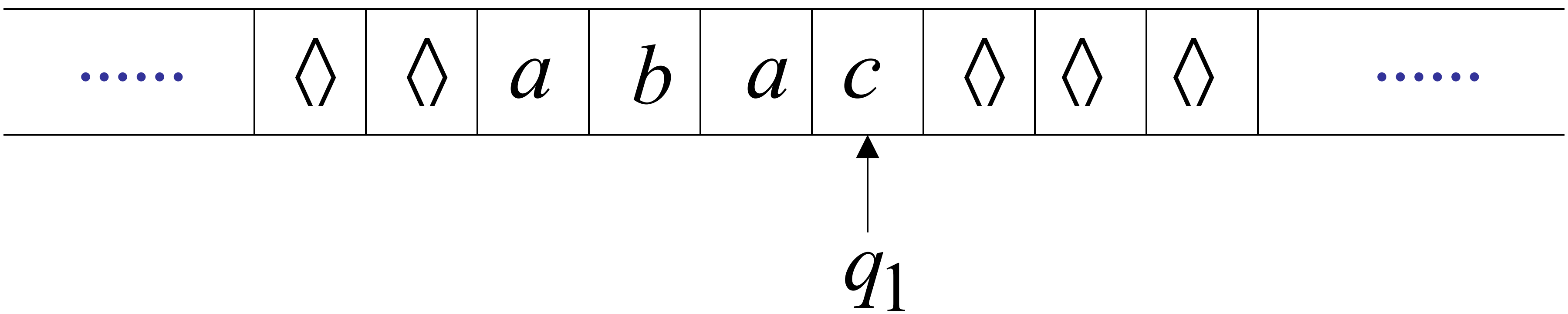
Allowed:

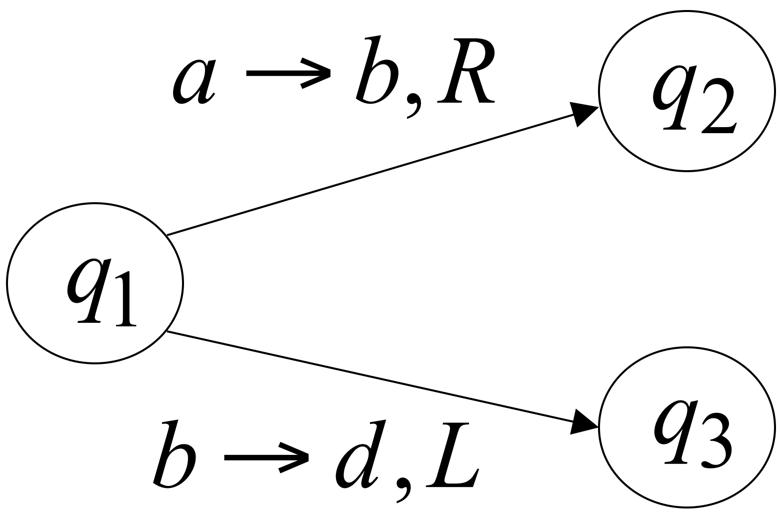
No transition for input symbol c

Halting

The machine ***halts*** if there are no possible transitions to follow

Example:

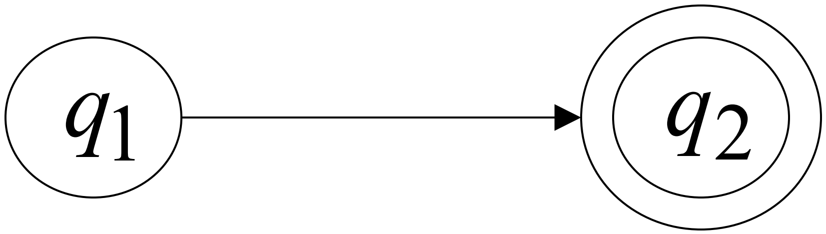




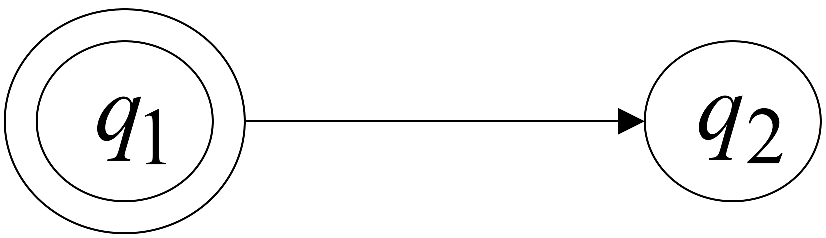
No possible transition

HALT!!!

Final States



Allowed



Not Allowed

- Final states have no outgoing transitions

- In a final state the machine halts

Acceptance

Accept Input → If machine halts in a final state

Reject Input → If machine halts in a non-final state

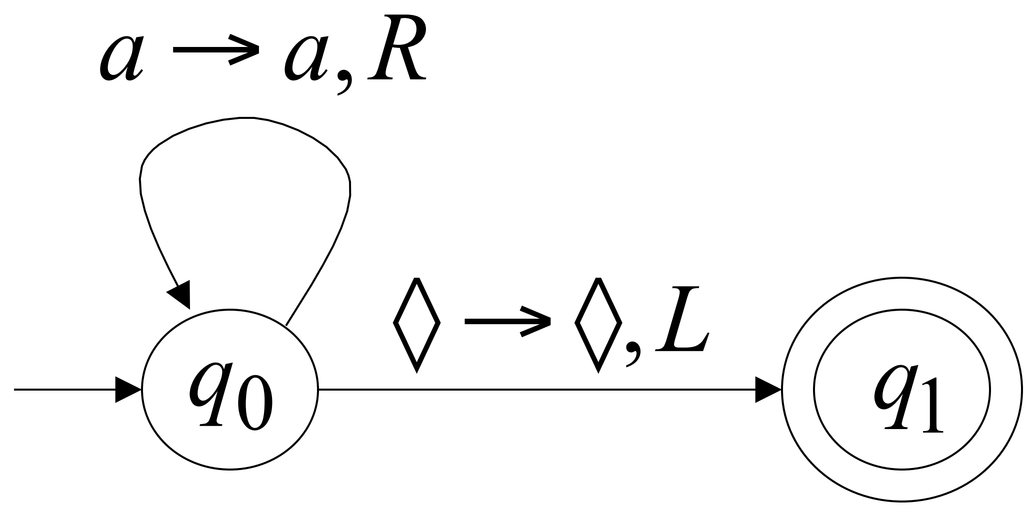
or

If machine enters an *infinite loop*

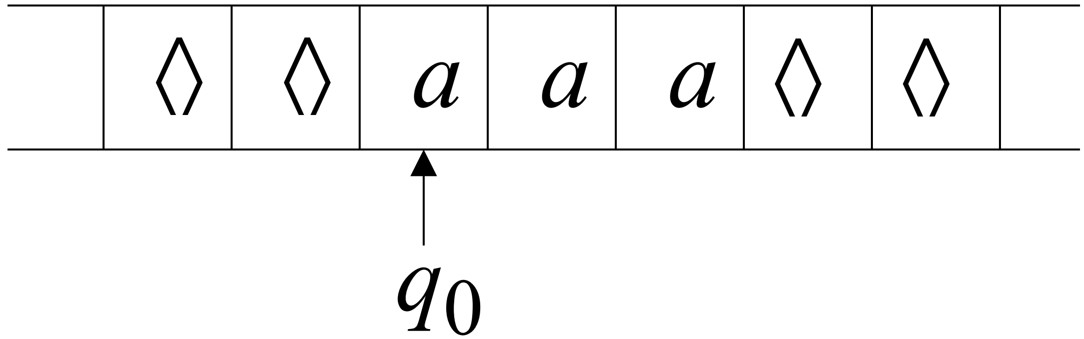
Turing Machine Example

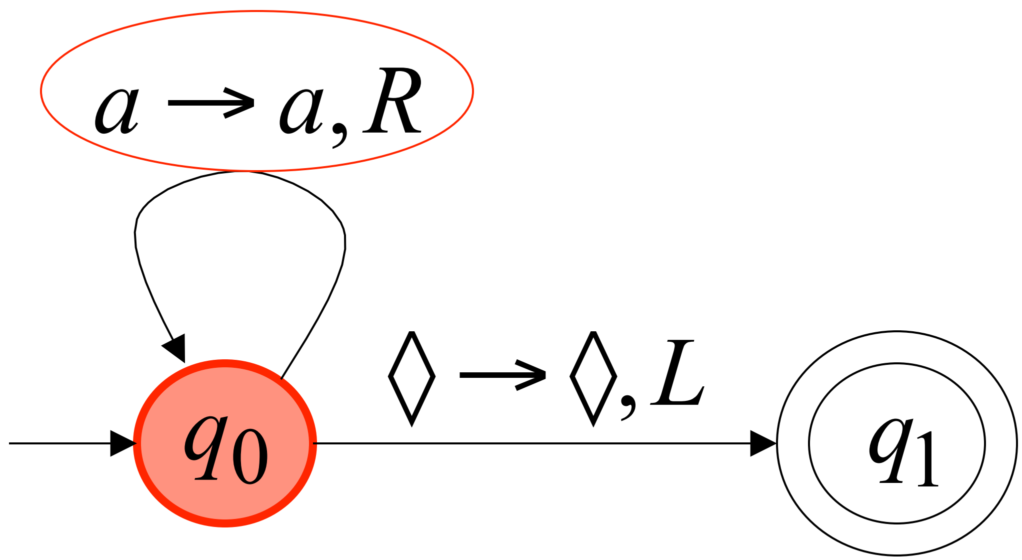
A Turing machine that accepts the language:

$$aa^*$$

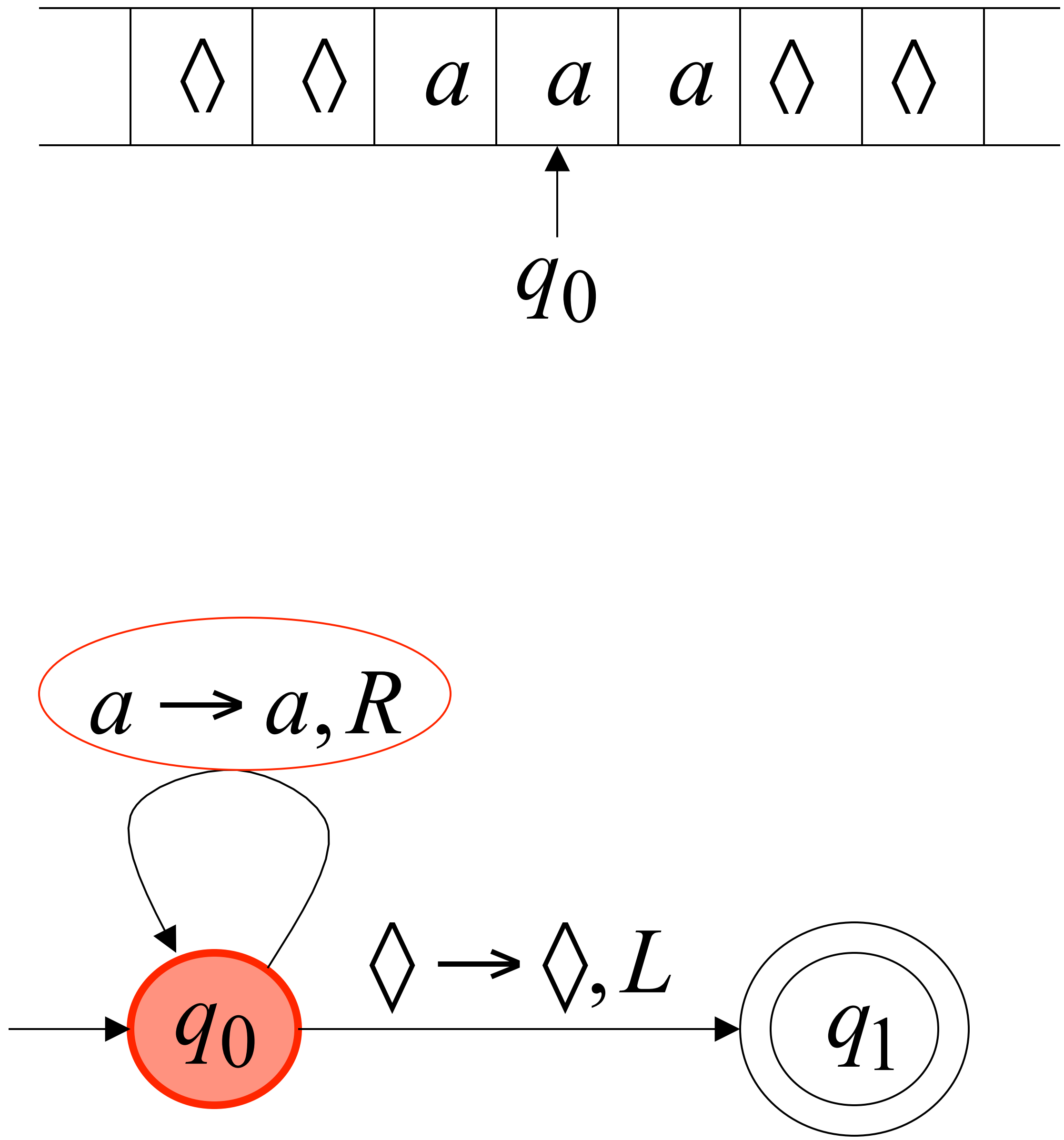


Time 0

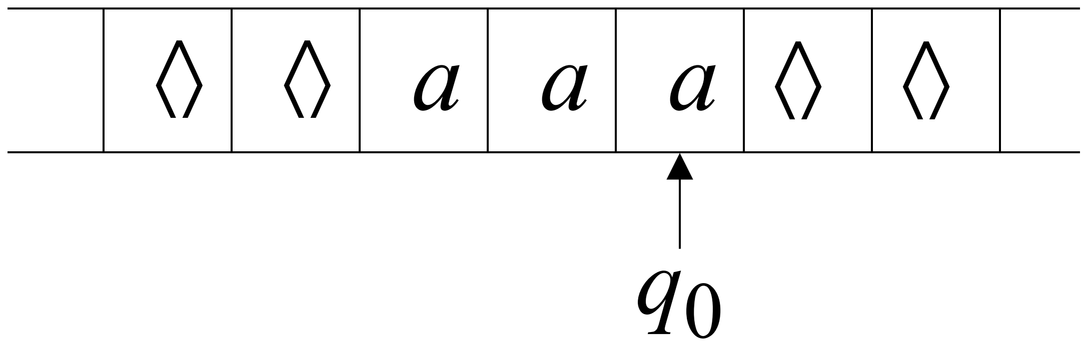


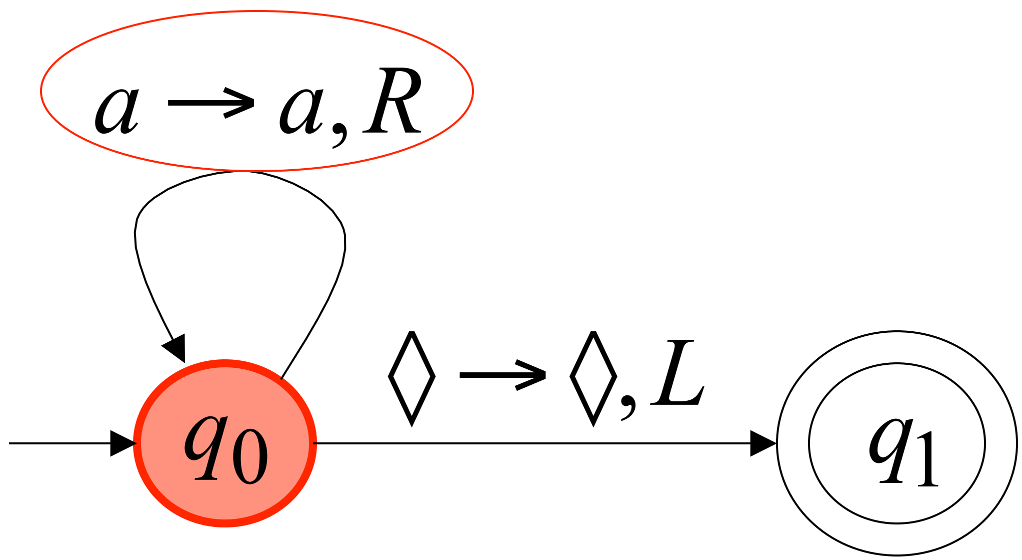


Time 1

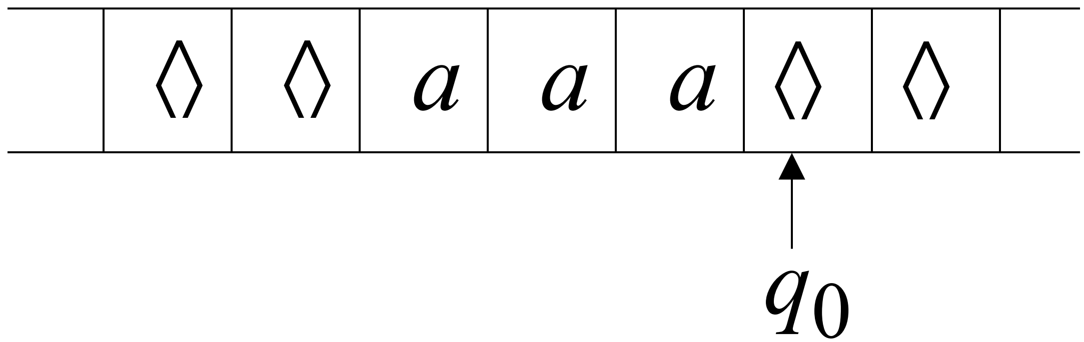


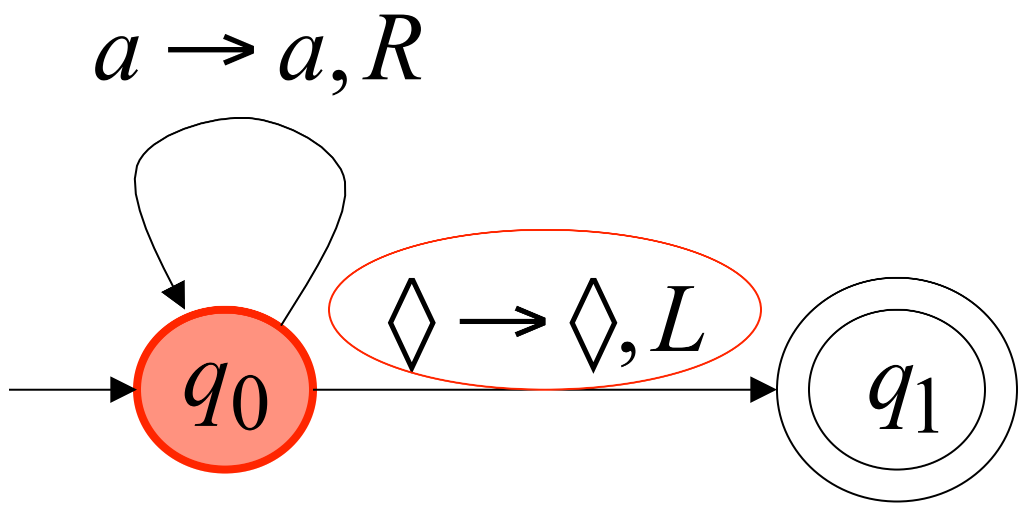
Time 2



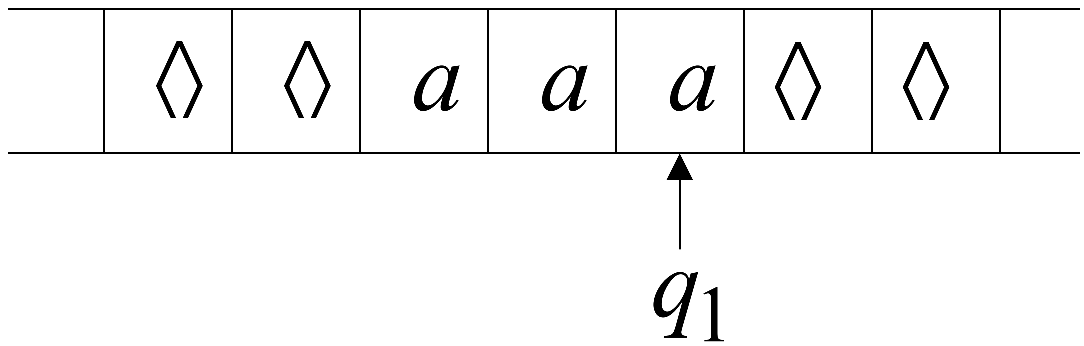


Time 3

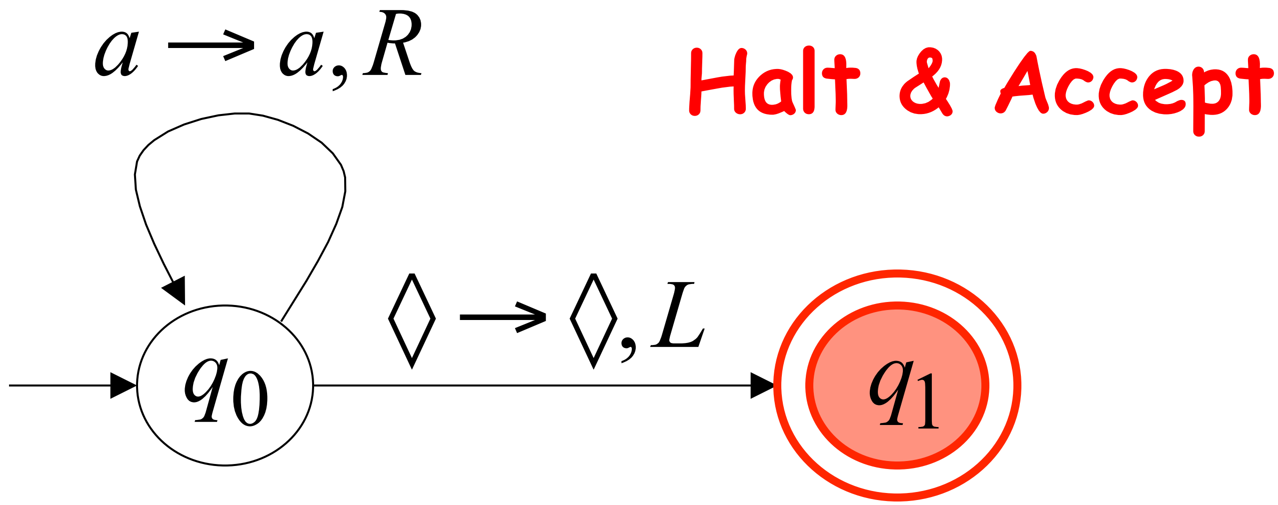




Time 4



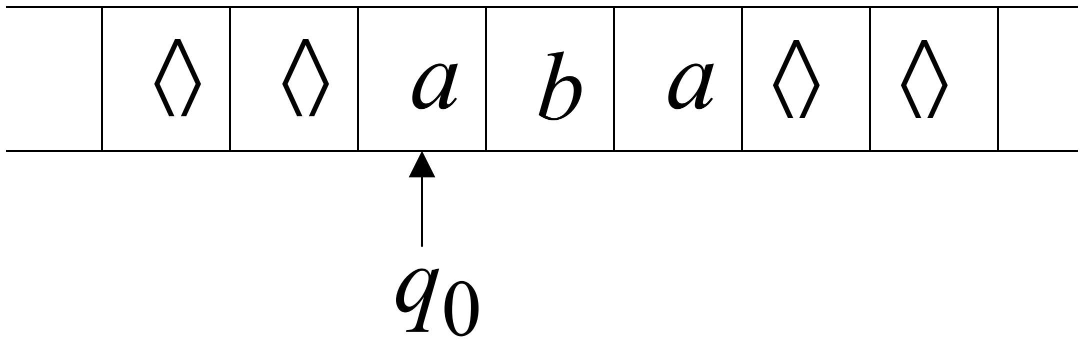
◊
◊
a
a
a
◊
◊
q_1

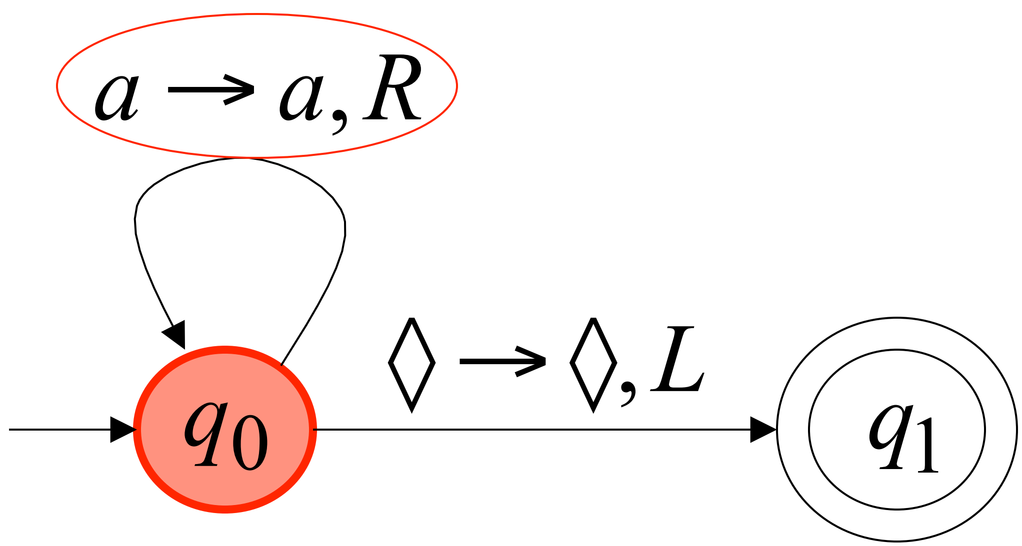


$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1
Halt & Accept

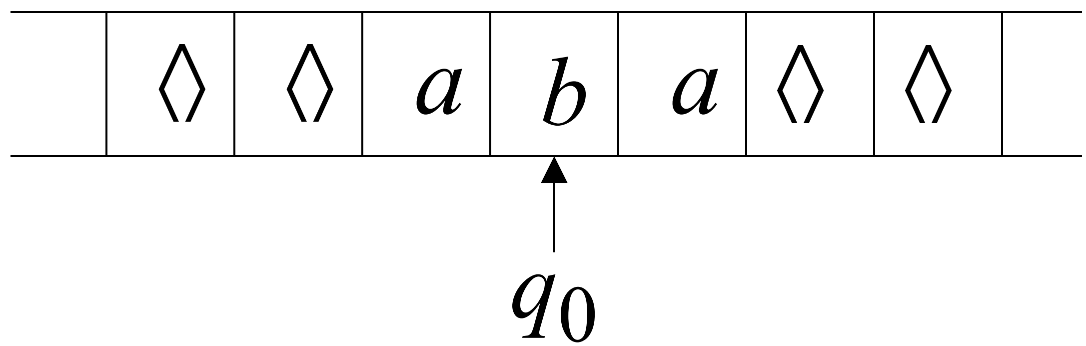
Rejection Example

Time 0



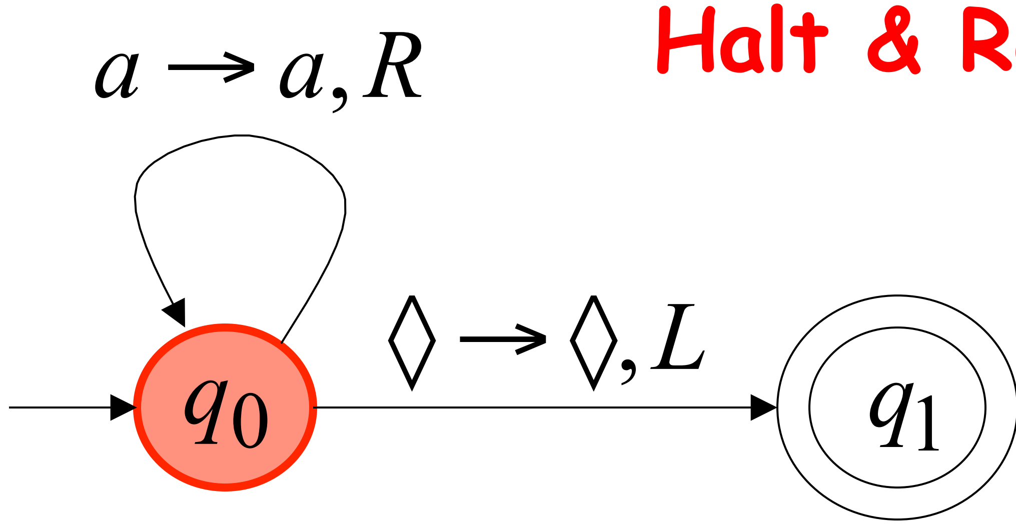


Time 1

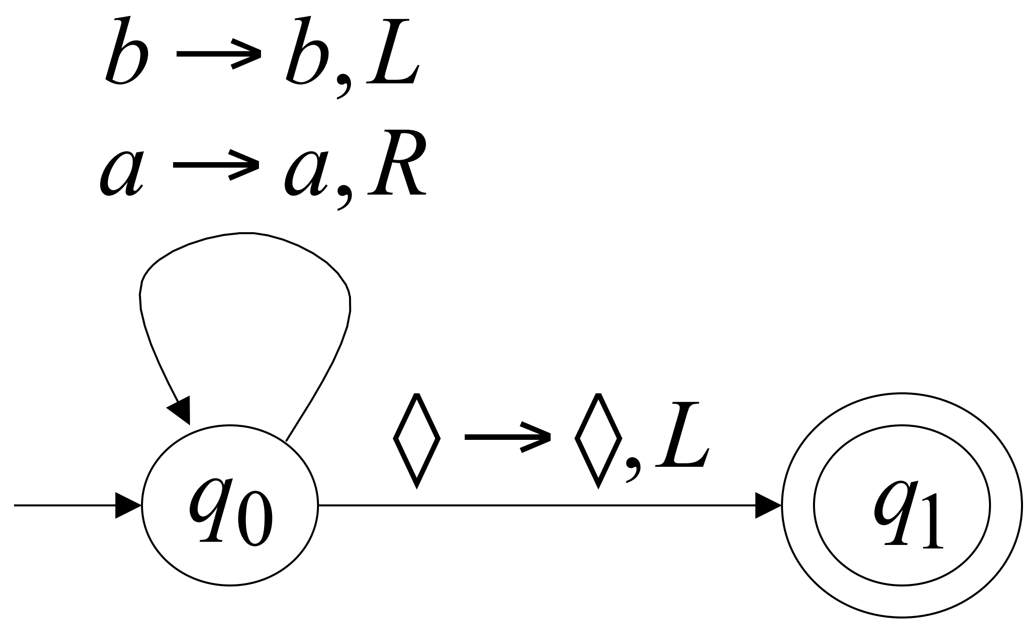


No possible Transition

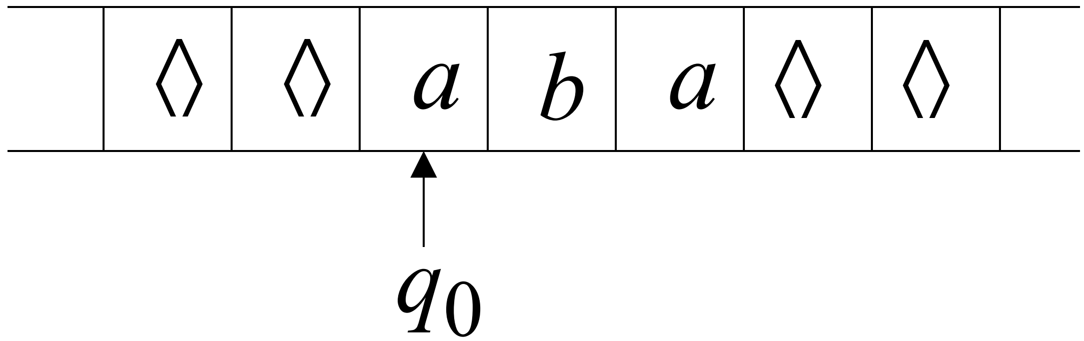
Halt & Reject



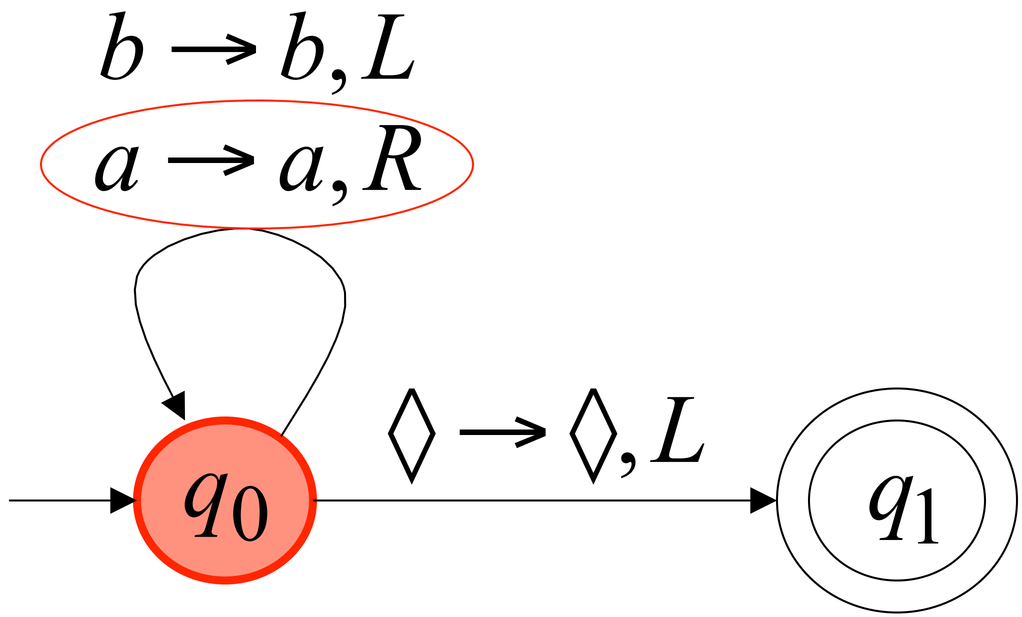
Infinite Loop Example



Time 0

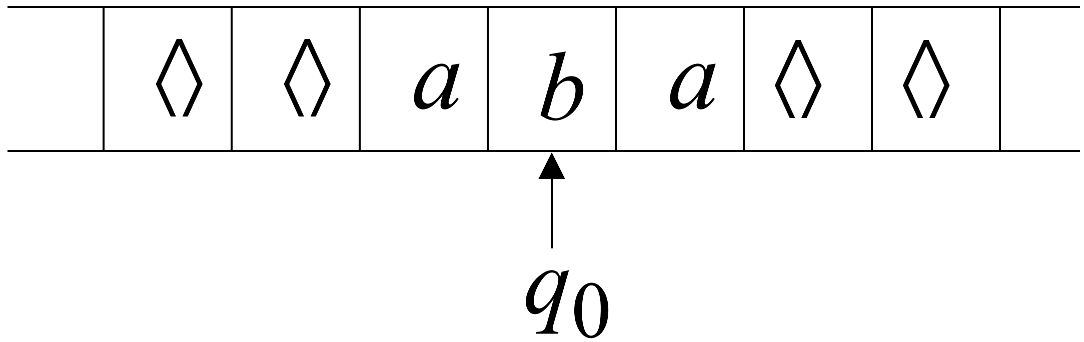


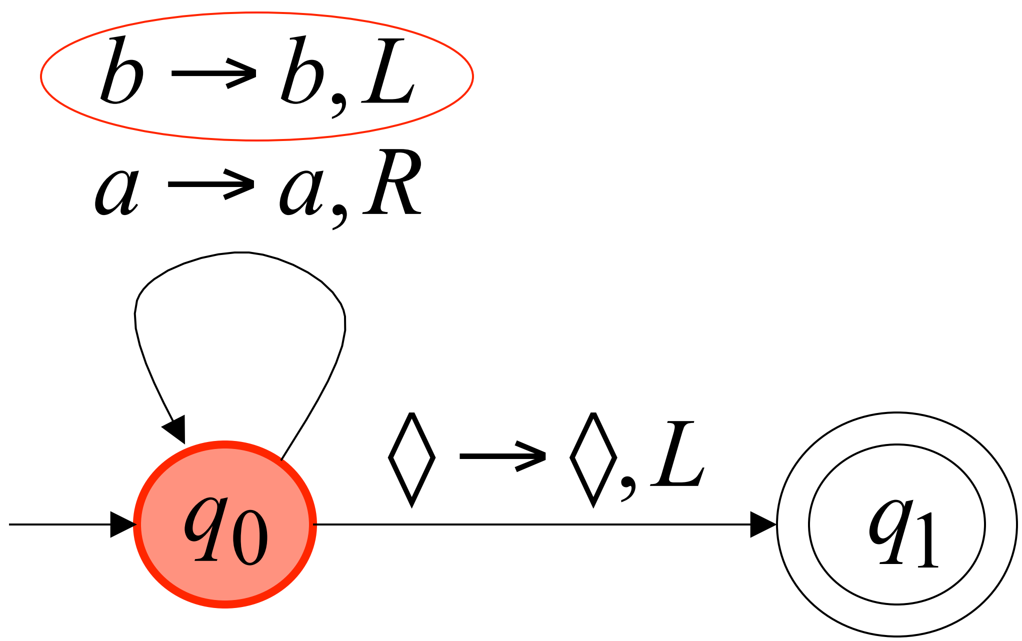
◊
◊
a
b
a
◊
◊
q_0



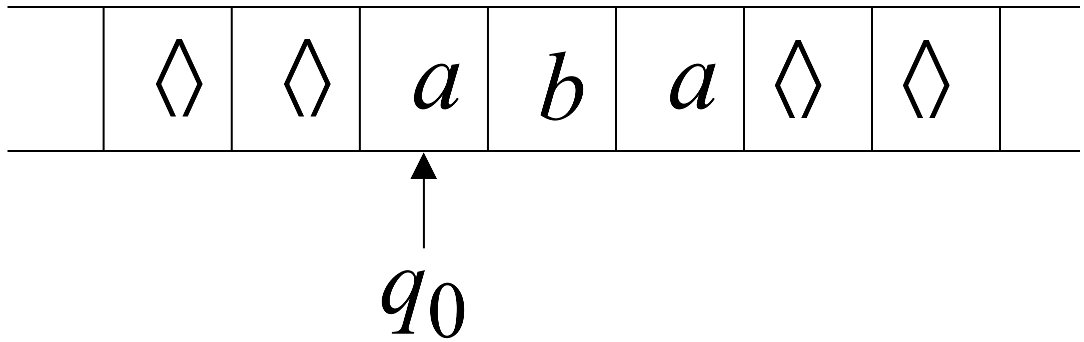
$b \rightarrow b, L$
$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

Time 1

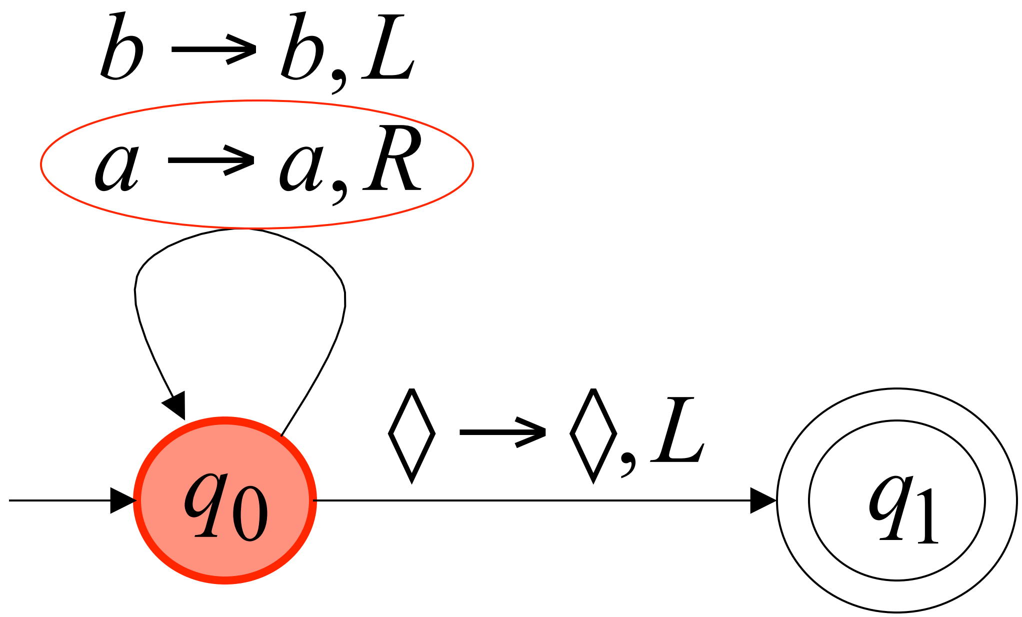

◊ ◊ a b a ◊ ◊
q_0


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

Time 2

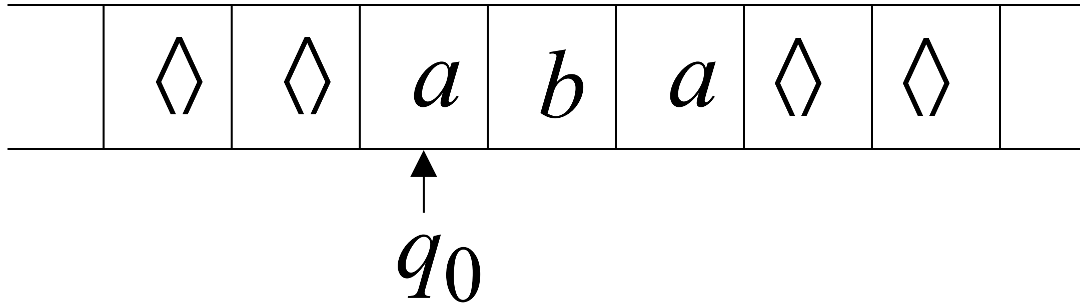


◊
◊
a
b
a
◊
◊
q_0

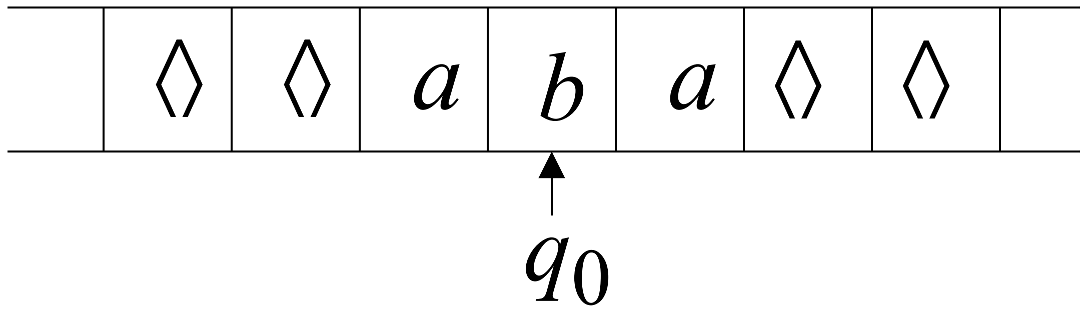


$b \rightarrow b, L$
$a \rightarrow a, R$
$\lozenge \rightarrow \lozenge, L$
q_0
q_1

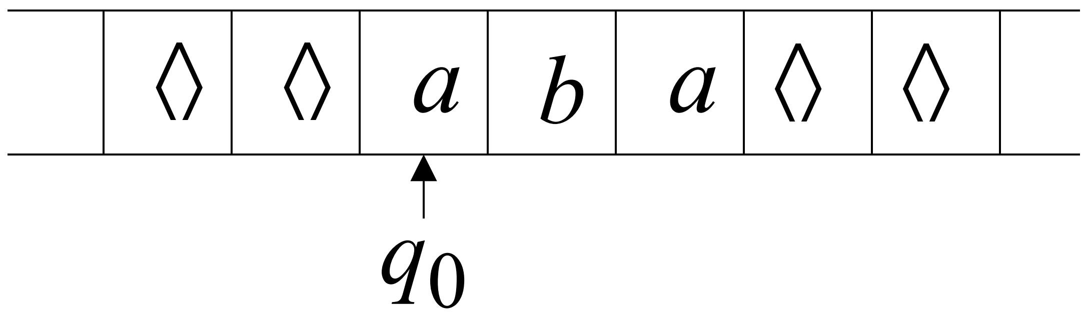
Time 2



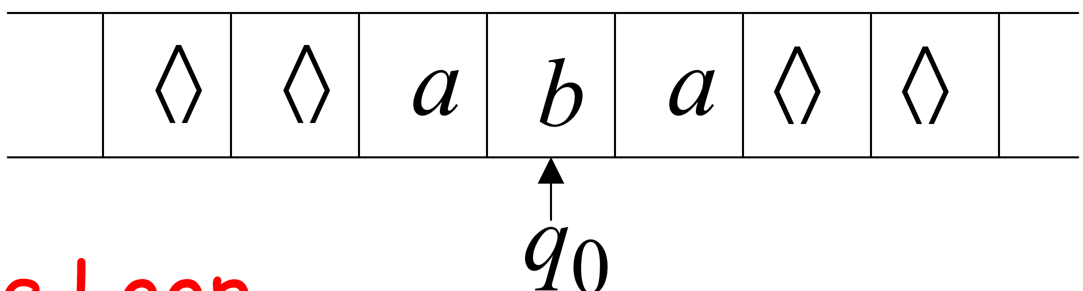
Time 3



Time 4



Time 5



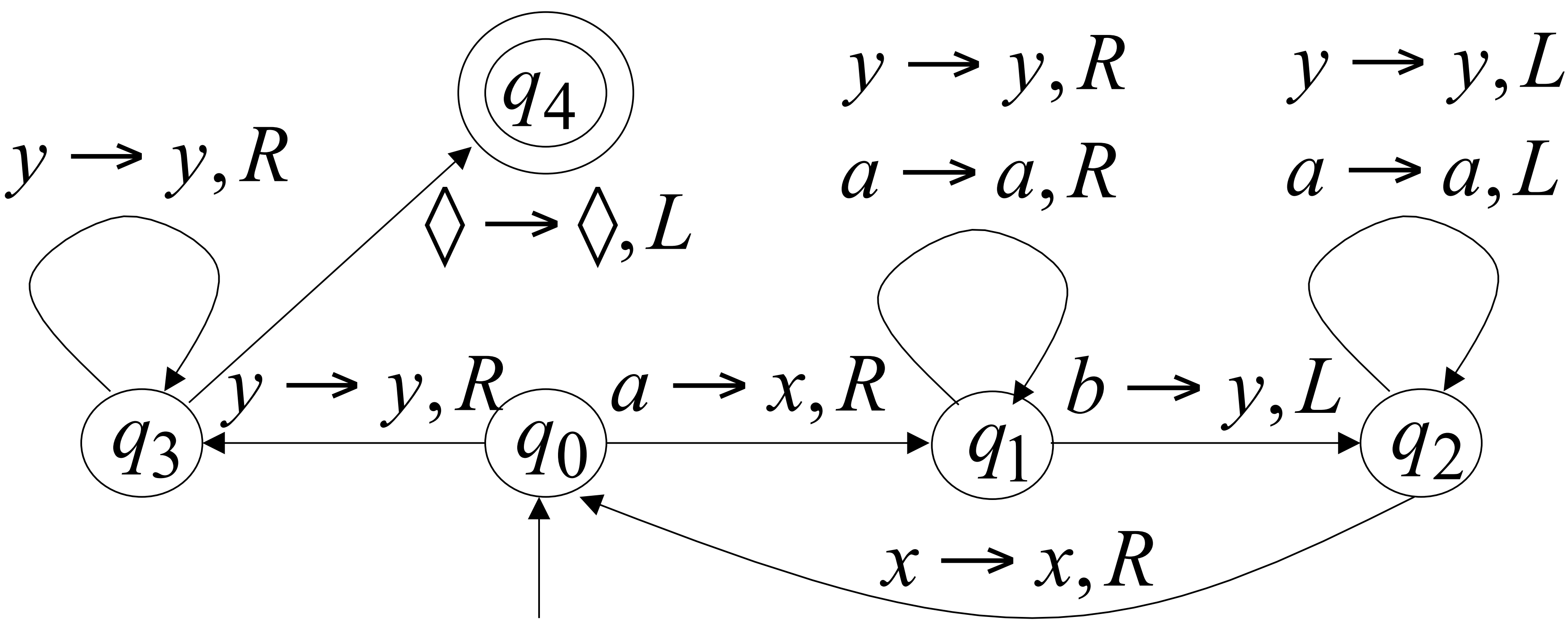
... Infinite Loop

Because of the infinite loop:

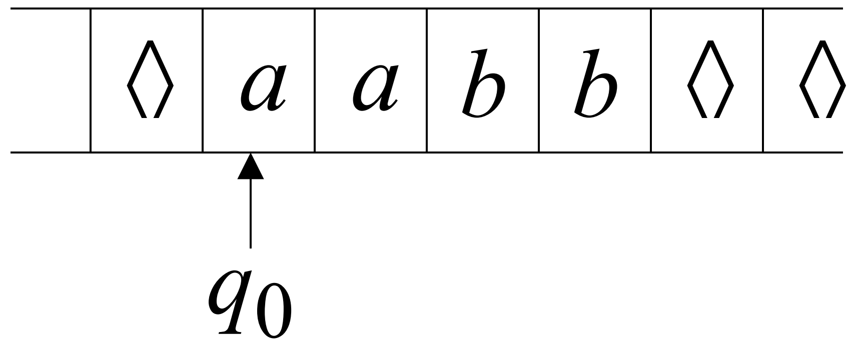
- The final state cannot be reached
- The machine never halts
- The input is not accepted

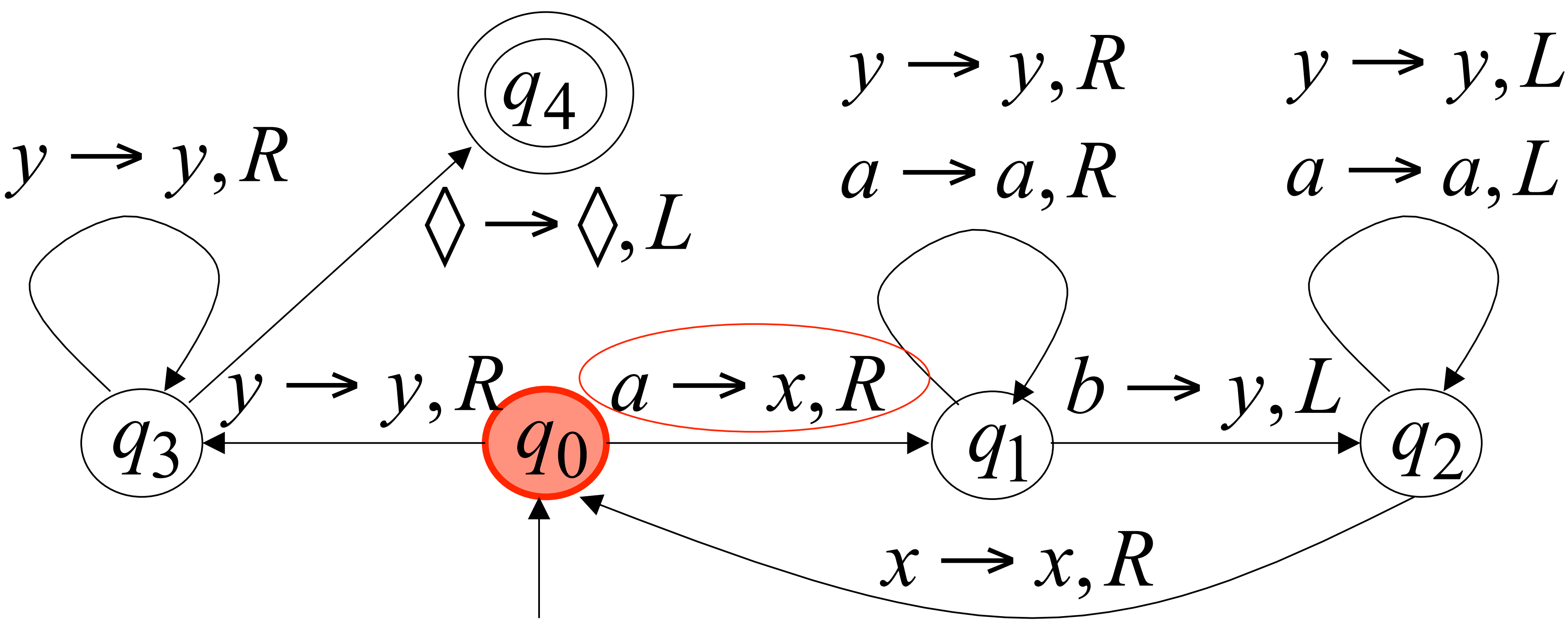
Another Turing Machine Example

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

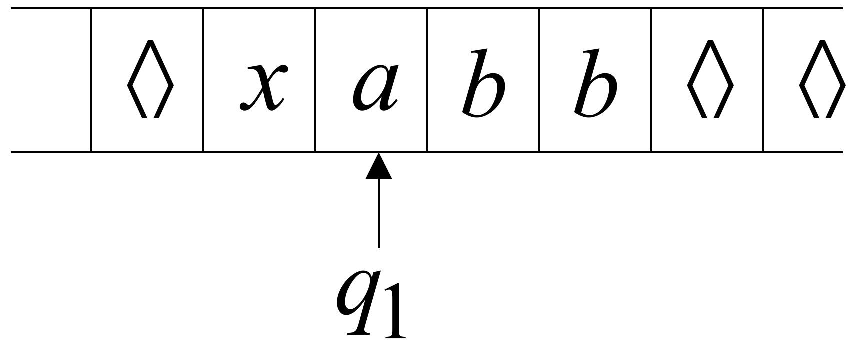


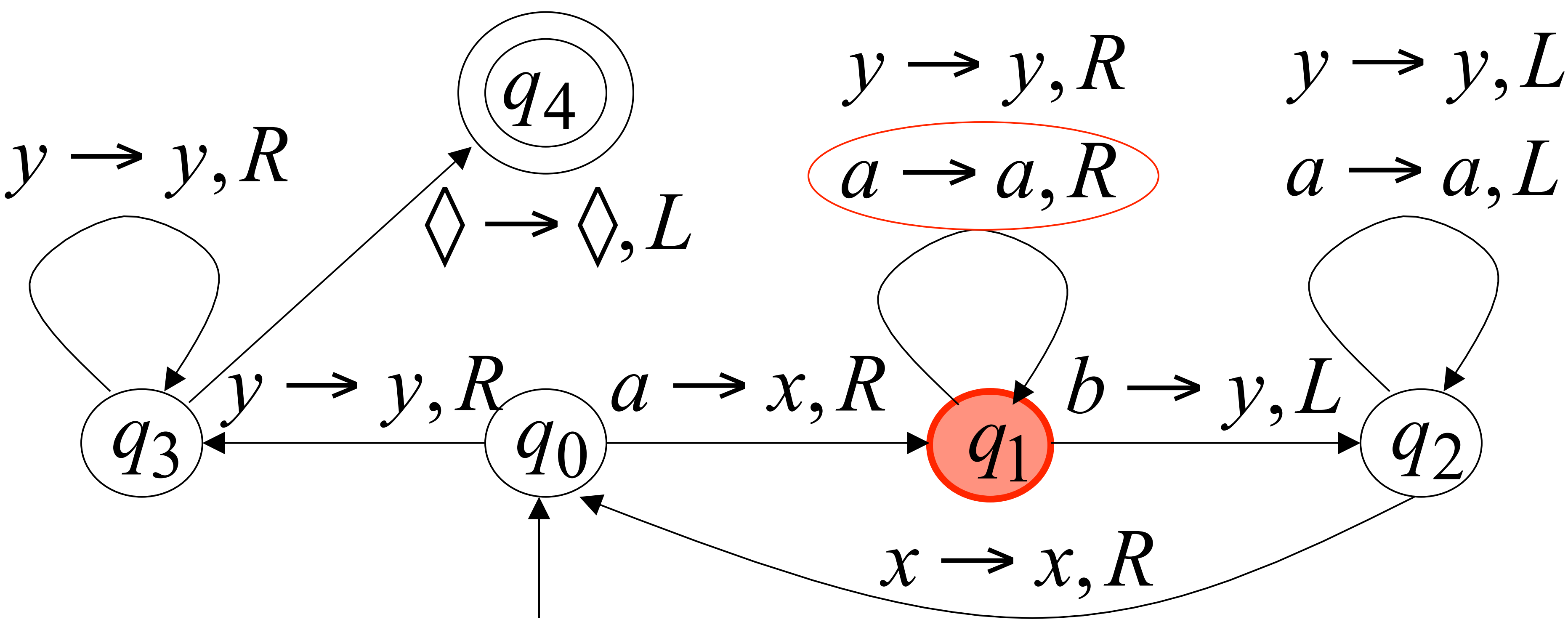
Time 0


◊ a a b b ◊ ◊
q_0

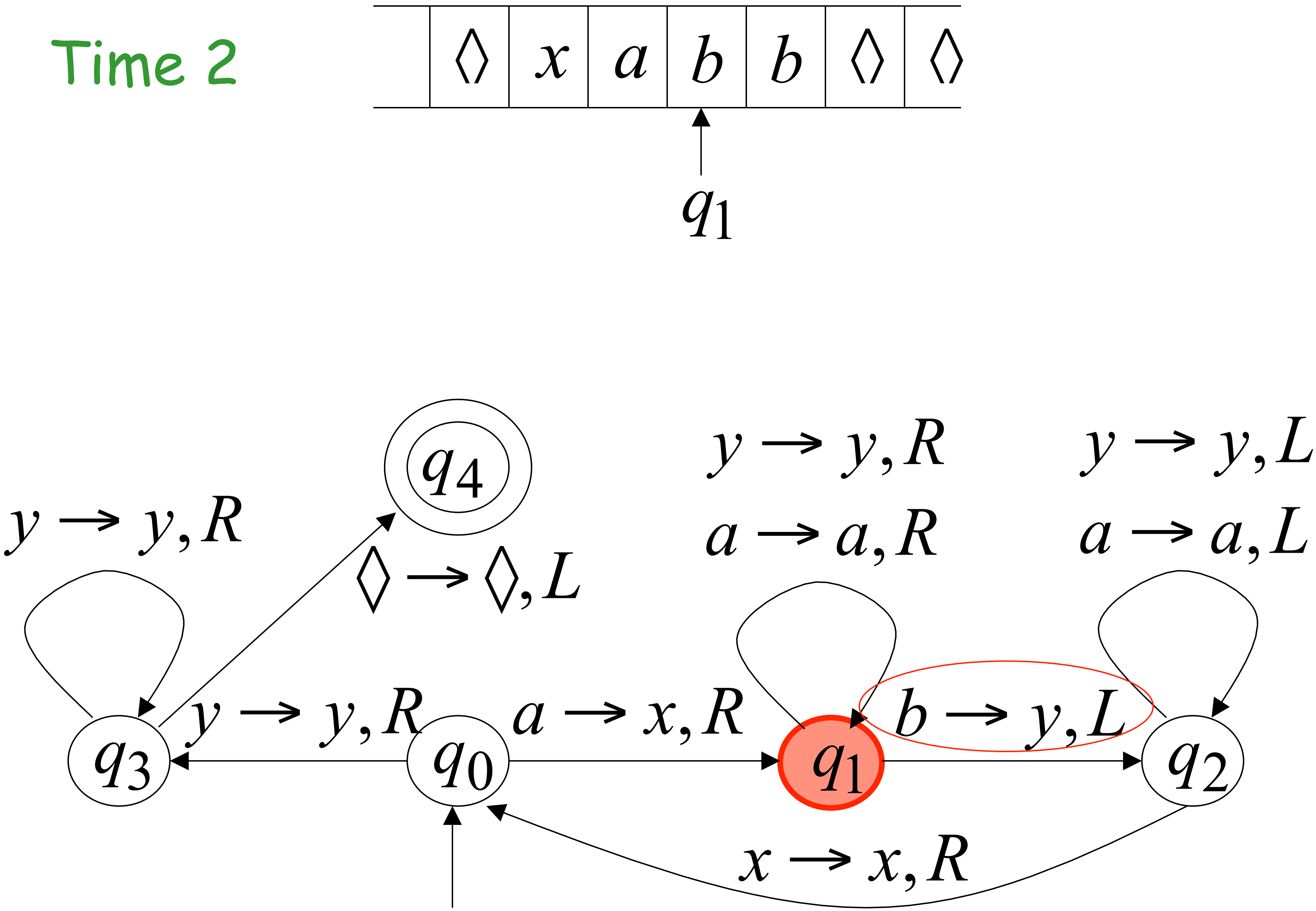

q_0
q_1
q_2
q_3
q_4
$a \rightarrow x, R$
$y \rightarrow y, R$
$a \rightarrow a, R$
$b \rightarrow y, L$
$y \rightarrow y, L$
$a \rightarrow a, L$
$x \rightarrow x, R$
$y \rightarrow y, R$
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$

Time 1

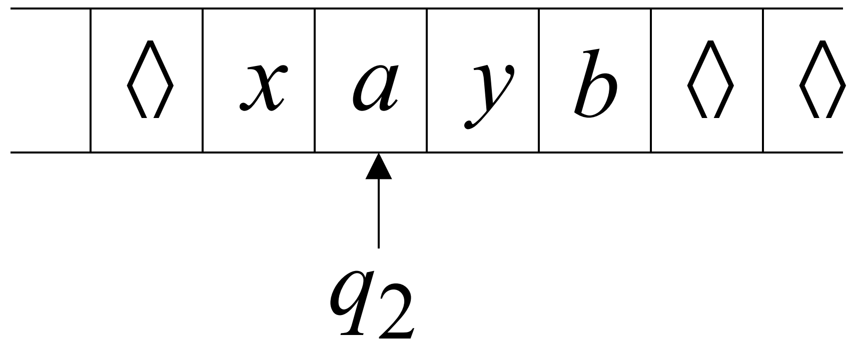

◊ x a b b ◊ ◊
q_1

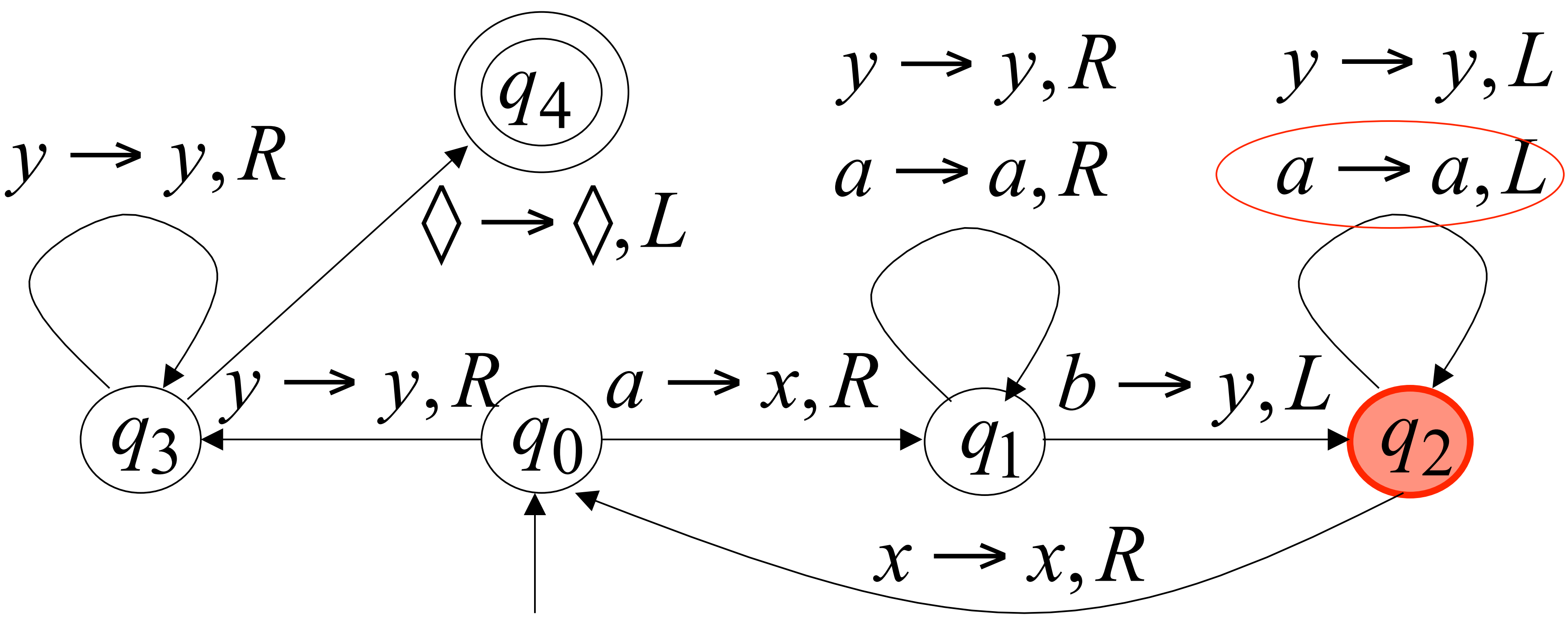

q_4
$y \to y, R$
$\lozenge \to \lozenge, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_0
q_1
q_2
$x \to x, R$

Time 2

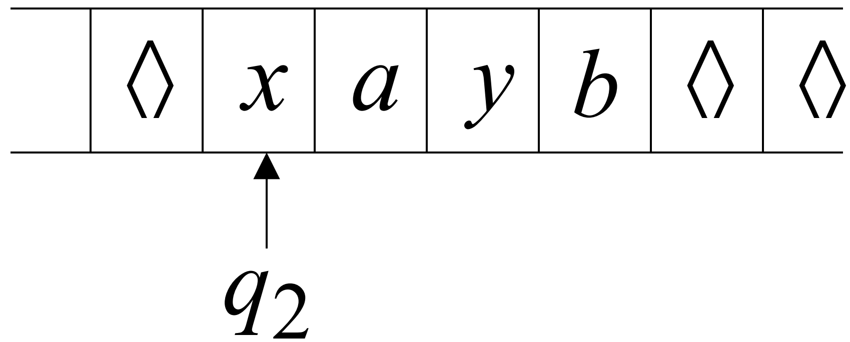

◊ x a b b ◊ ◊
q_1
q_4
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

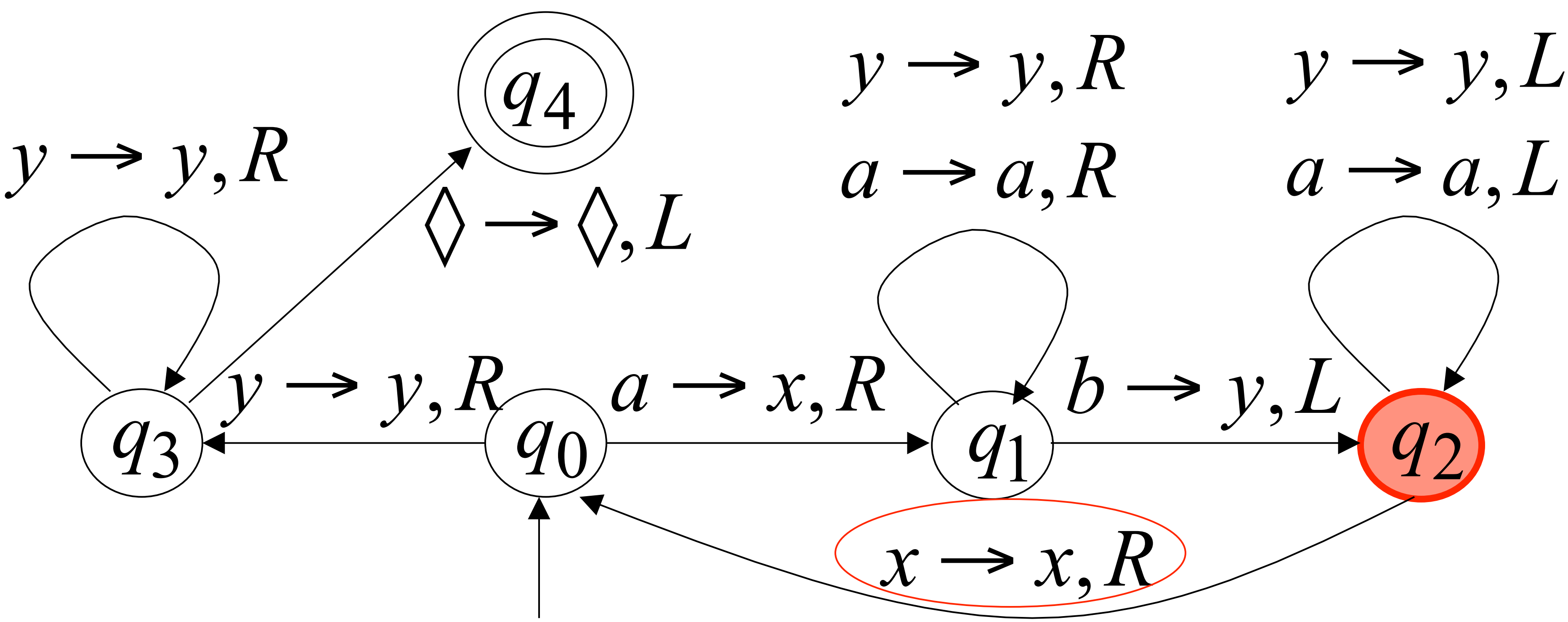
Time 3



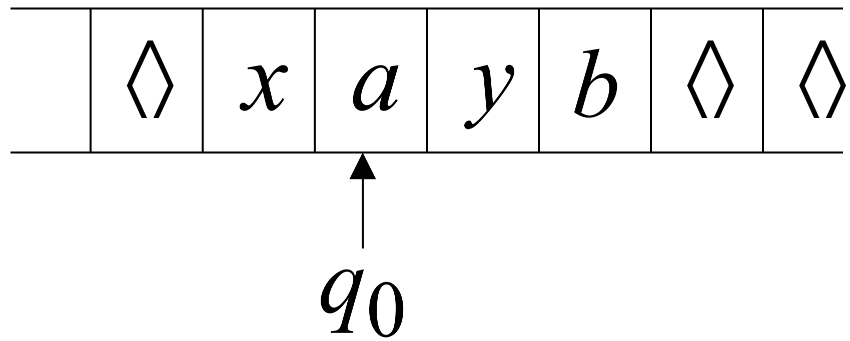


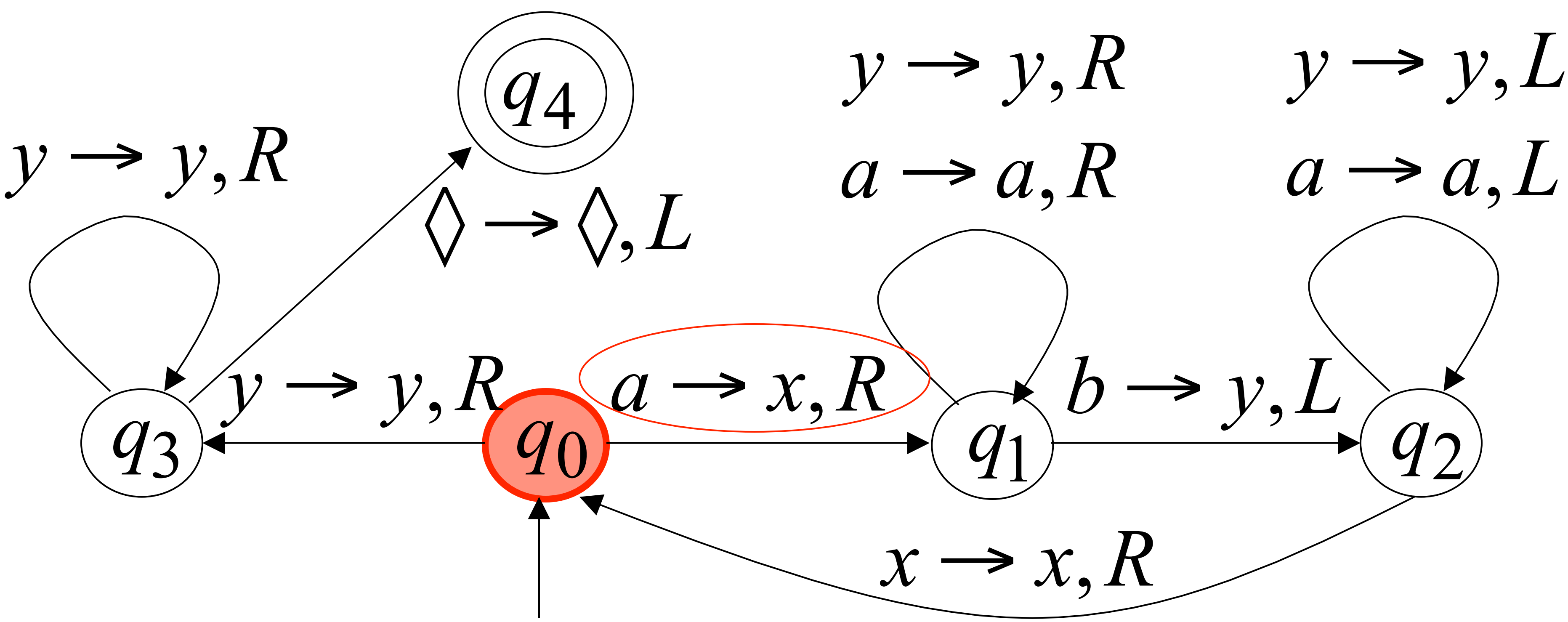
Time 4


◊
x
a
y
b
◊
◊
q_2

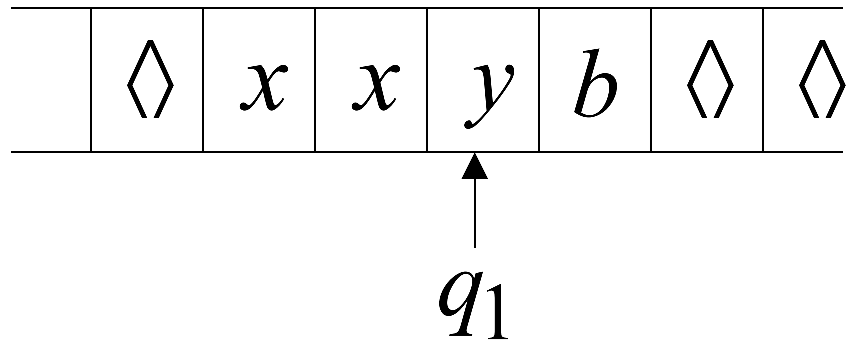

q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

Time 5

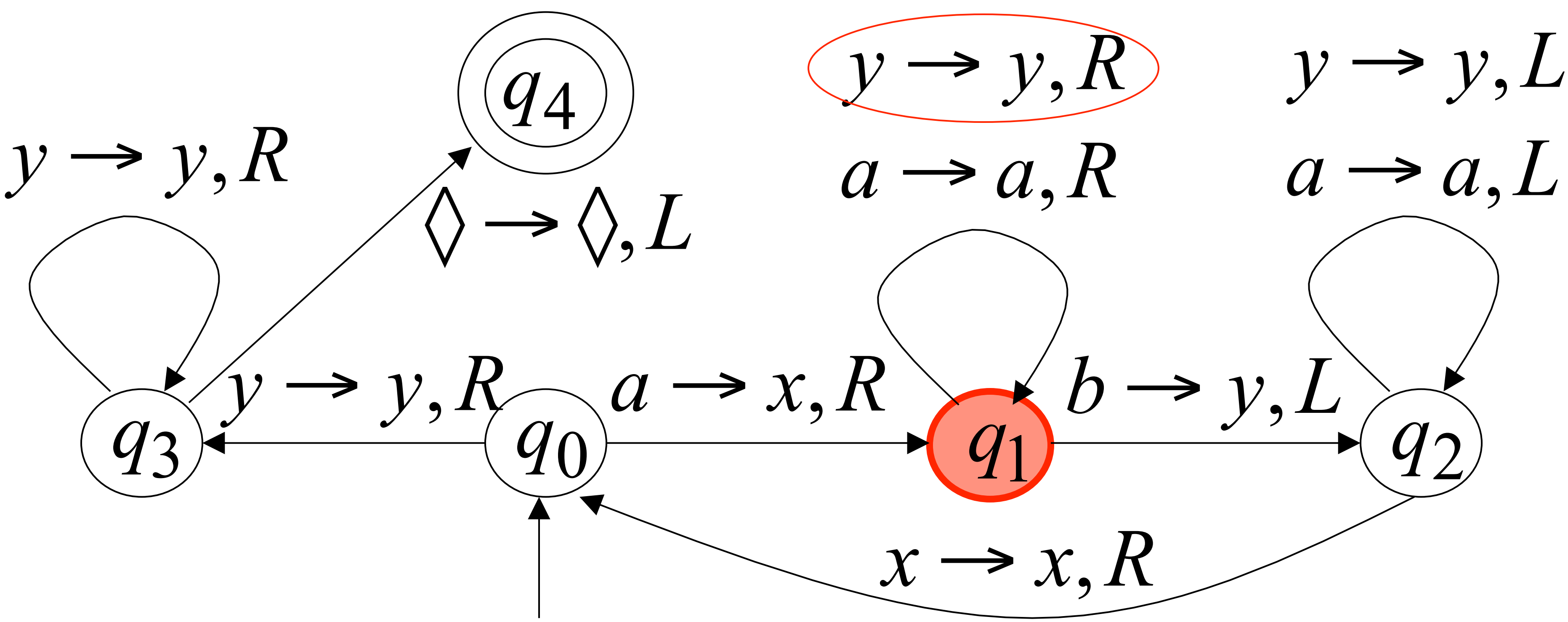

◊ x a y b ◊ ◊
q_0


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 6

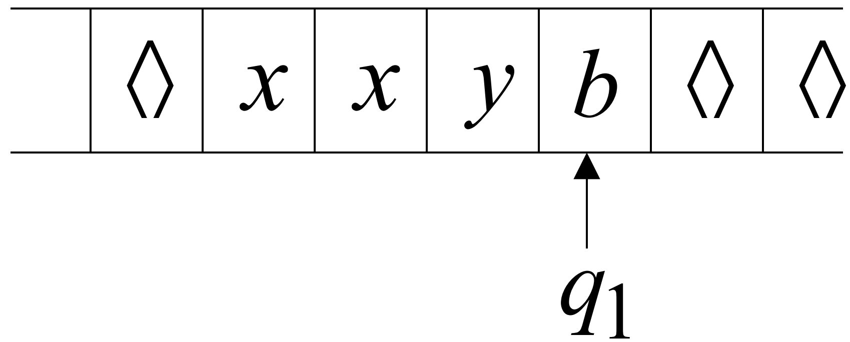


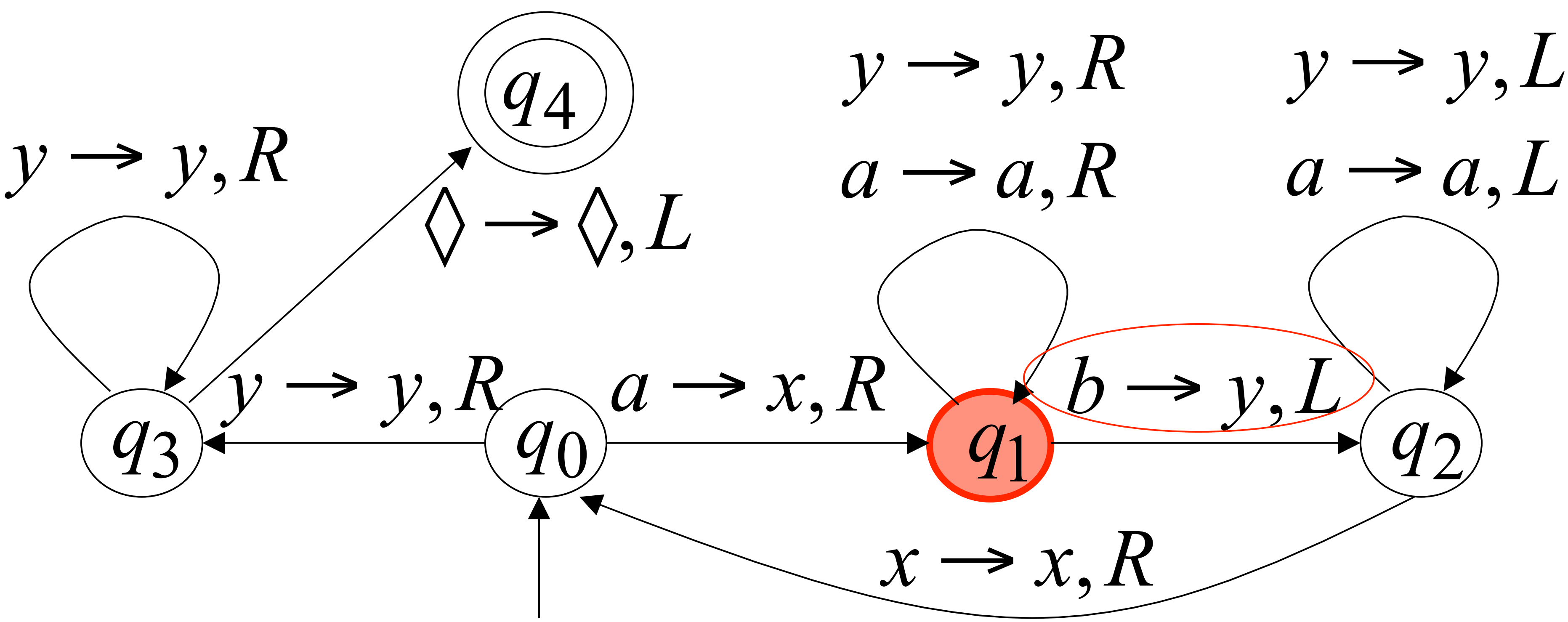
◊
x
x
y
b
◊
◊
q_1



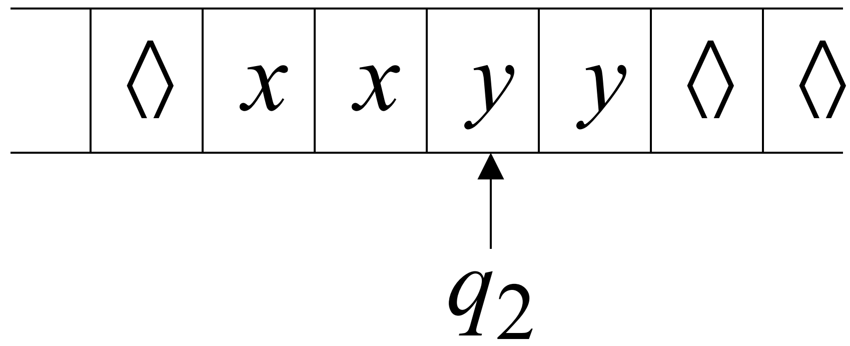
q_4
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

Time 7

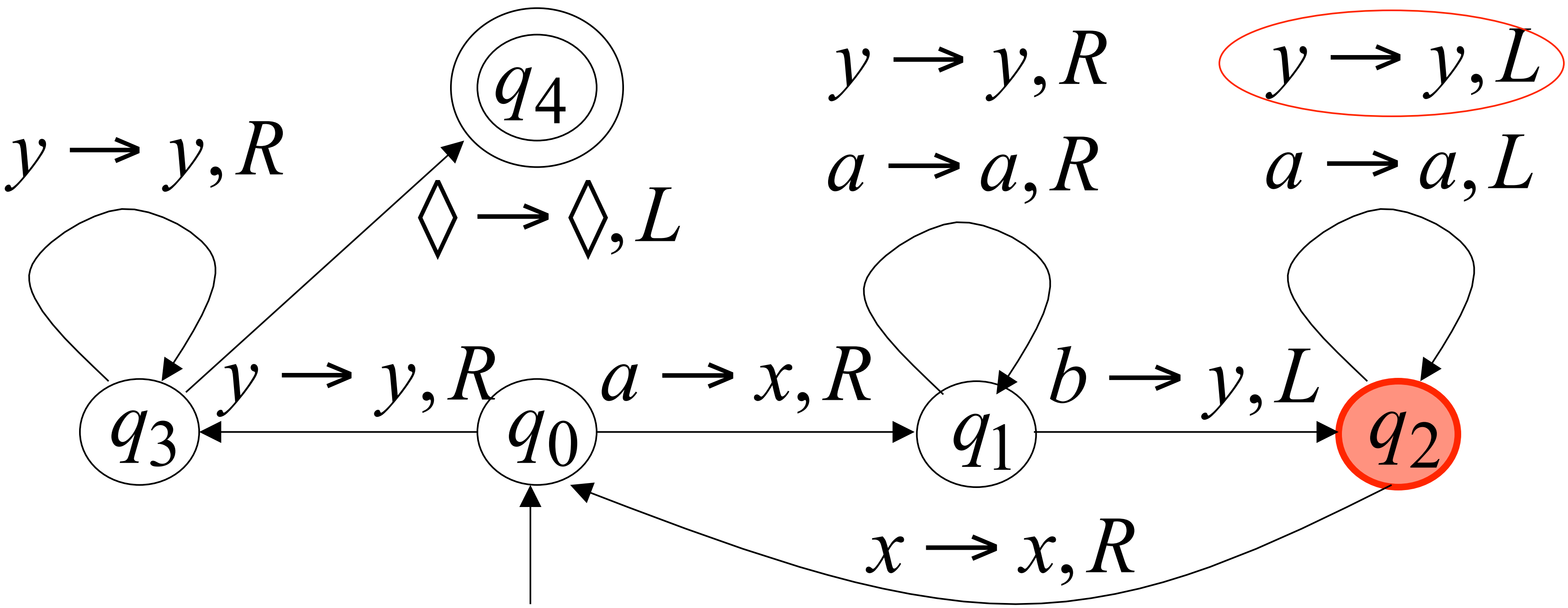

◊
x
x
y
b
◊
◊
q_1


q_4
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
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 8

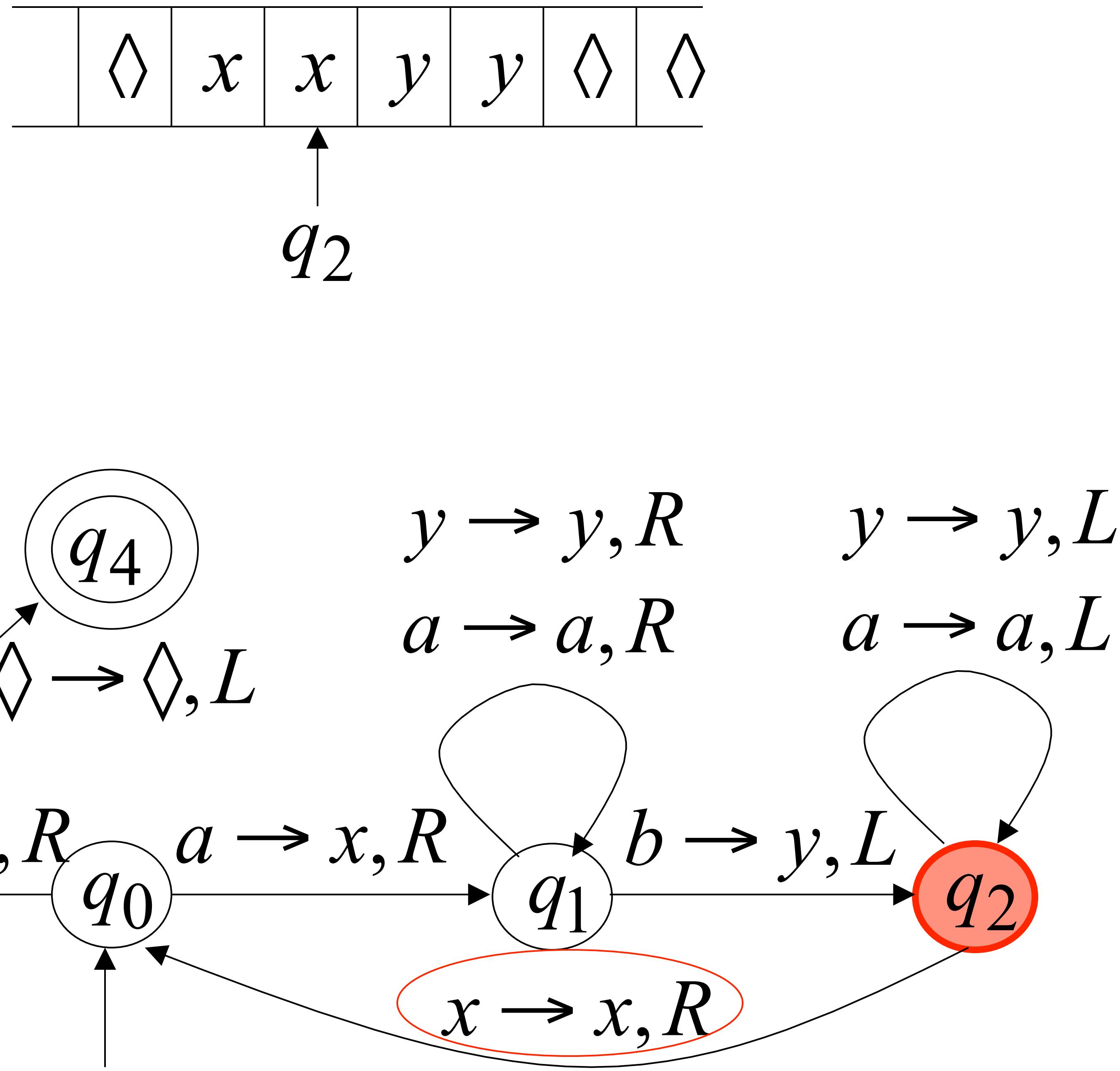


◊
x
x
y
y
◊
◊
q_2

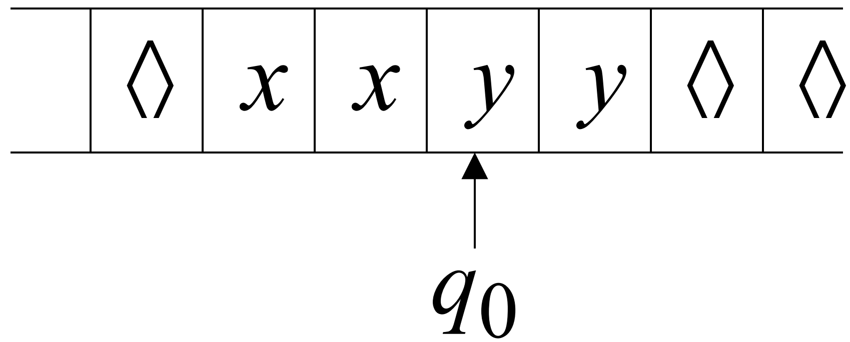


q_4
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

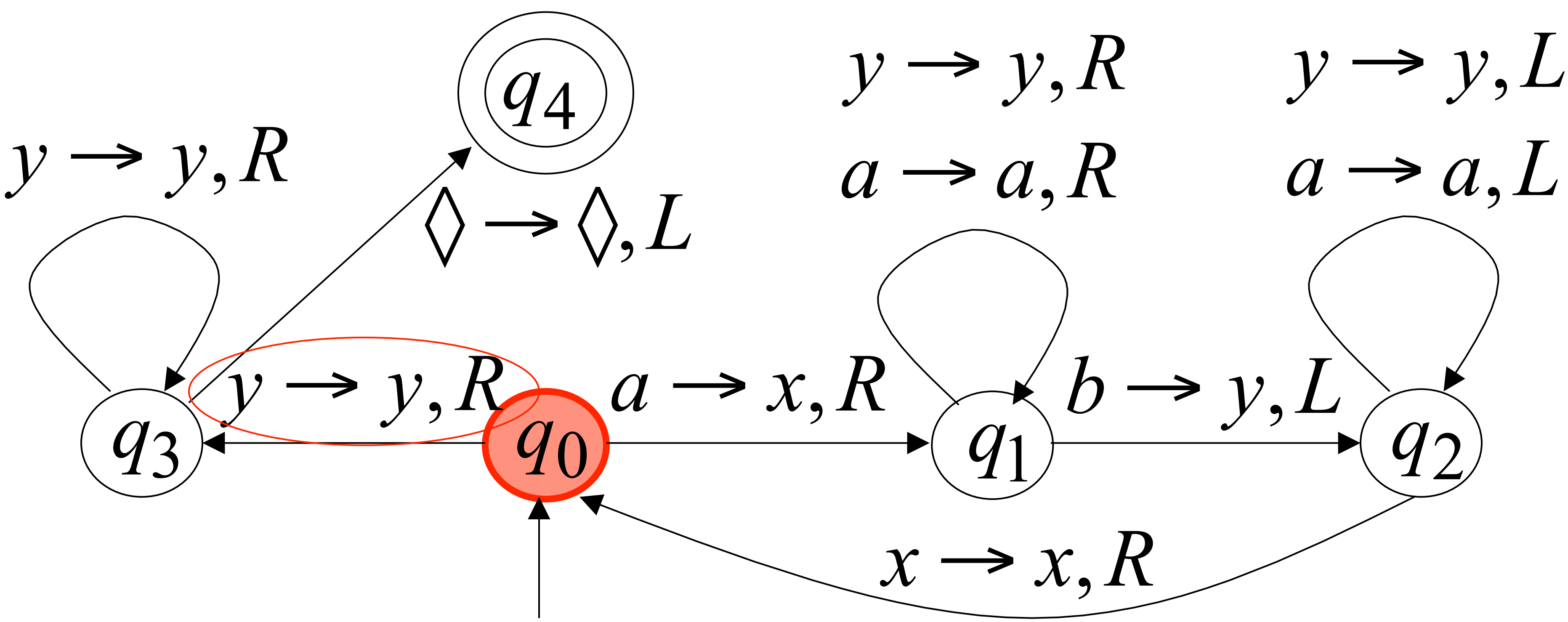
Time 9


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

Time 10

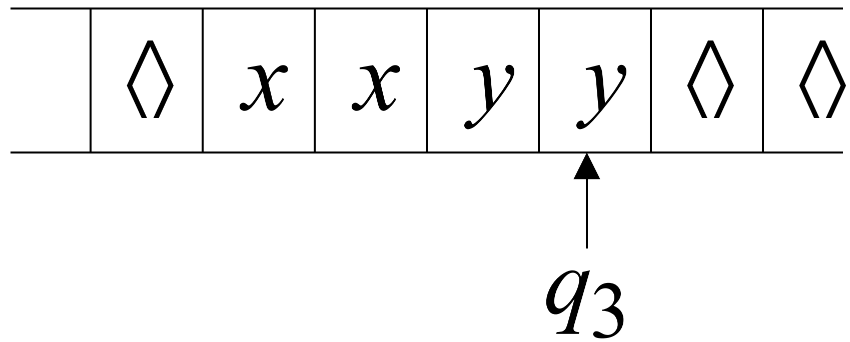


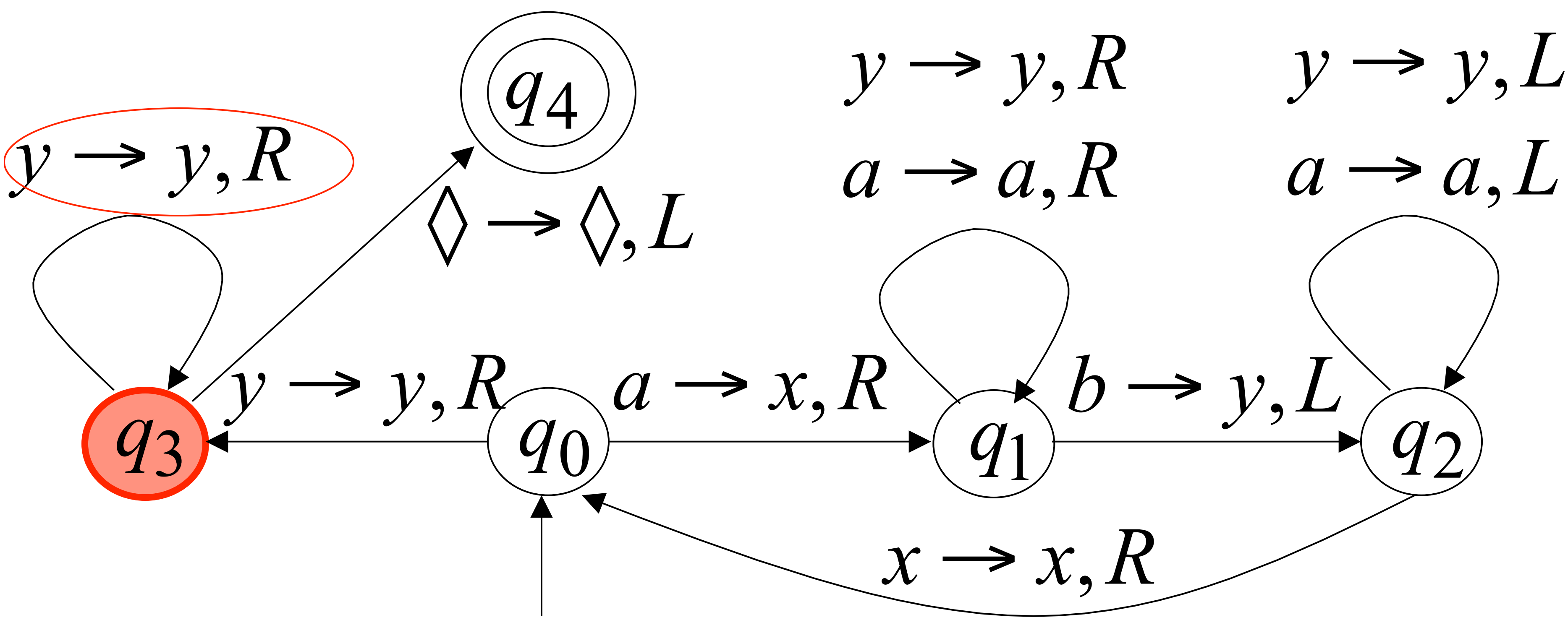
◊
x
x
y
y
◊
◊
q_0



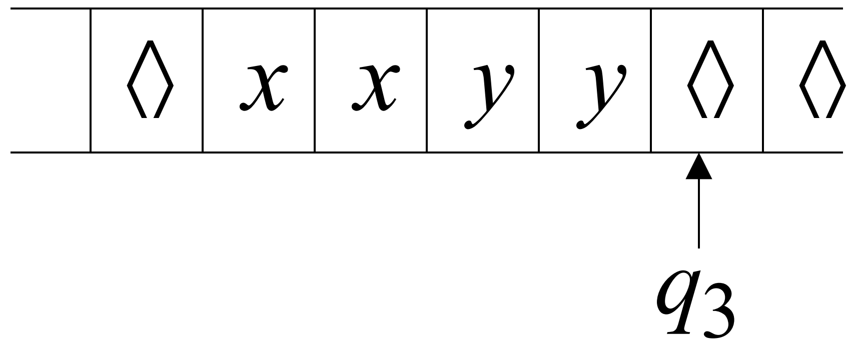
q_4
$y \rightarrow y, R$
◊ → ◊, L
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
q_3
$y \rightarrow y, R$
q_0
$a \rightarrow x, R$
q_1
$b \rightarrow y, L$
q_2
$x \rightarrow x, R$

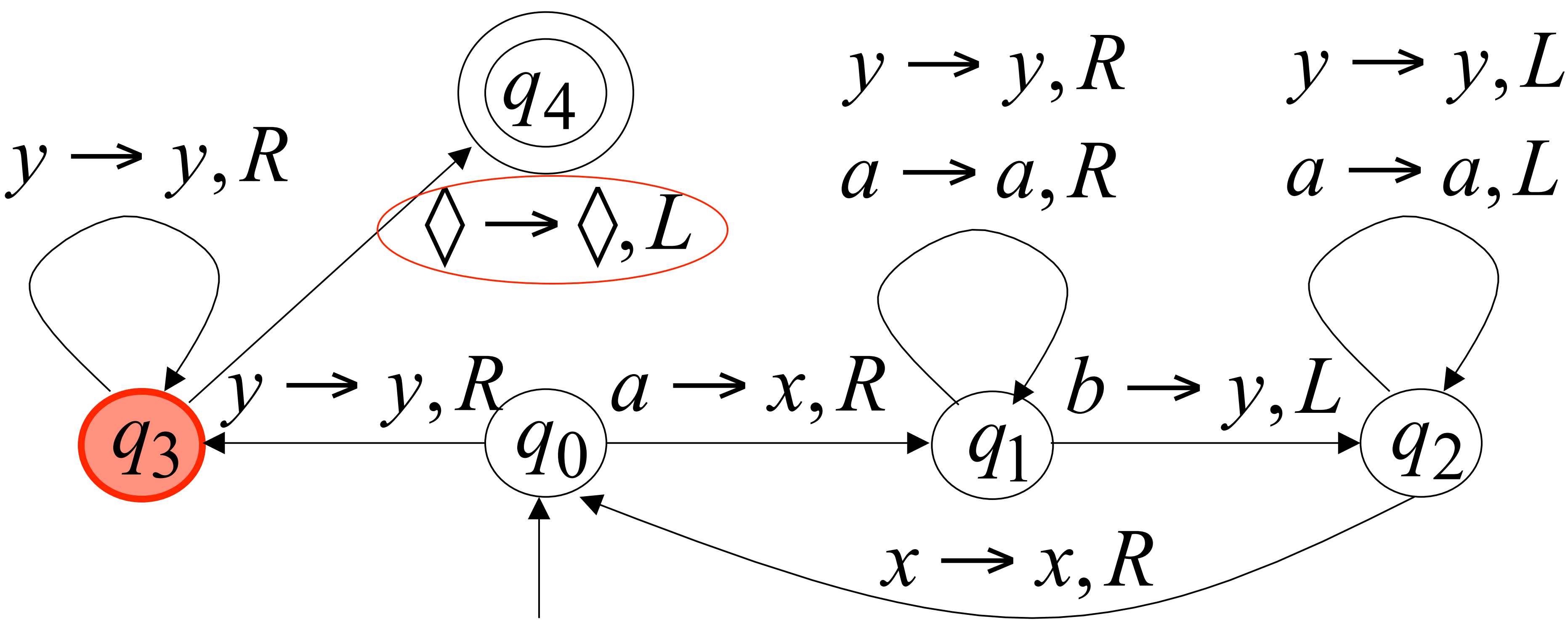
Time 11


◊
x
x
y
y
◊
◊
q_3

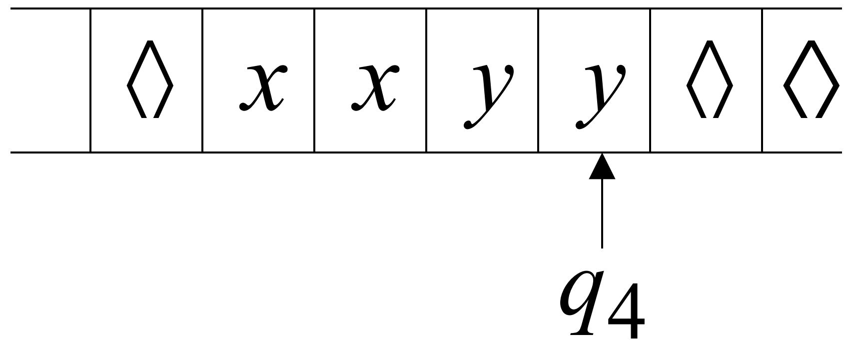

$y \to y, R$
q_4
$\lozenge \to \lozenge, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_0
q_1
q_2
$x \to x, R$

Time 12

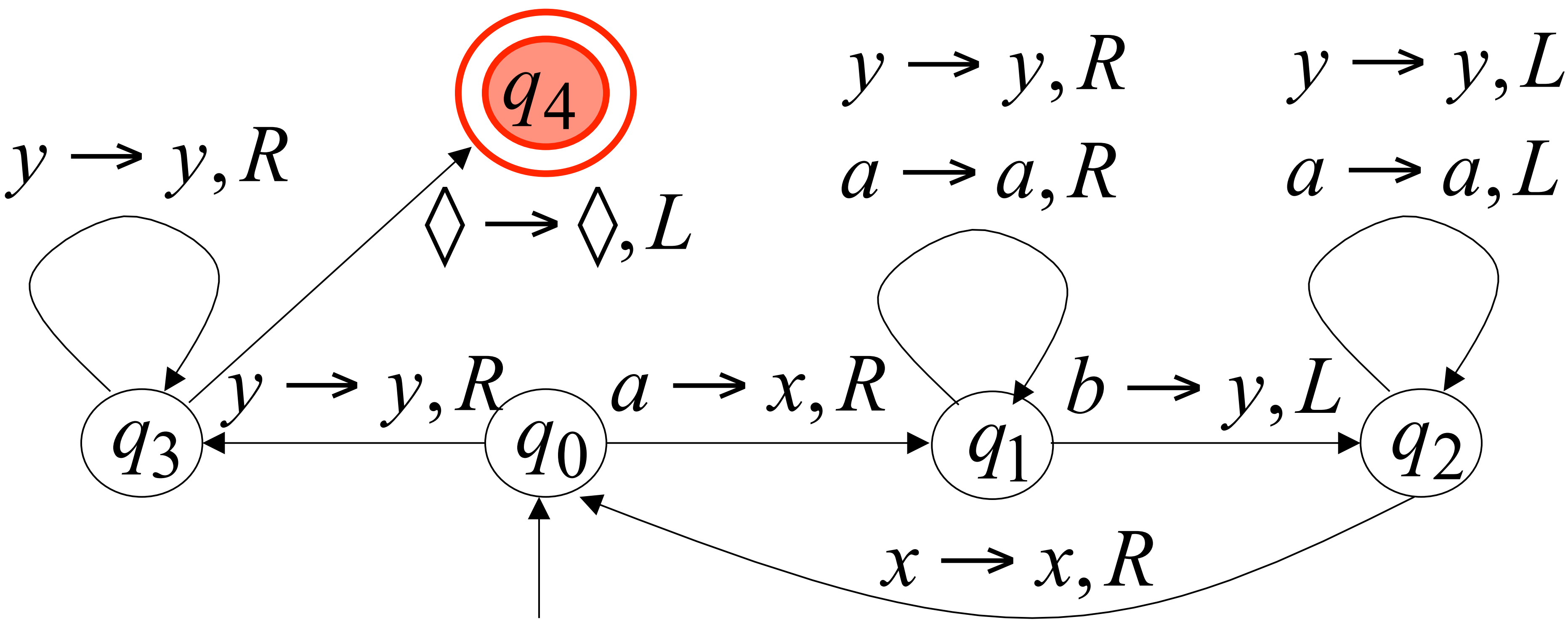

◊
x
x
y
y
◊
◊
q_3


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 13



Halt & Accept



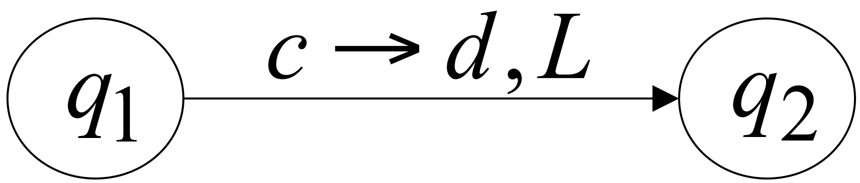
Observation:

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

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

Formal Definitions for Turing Machines

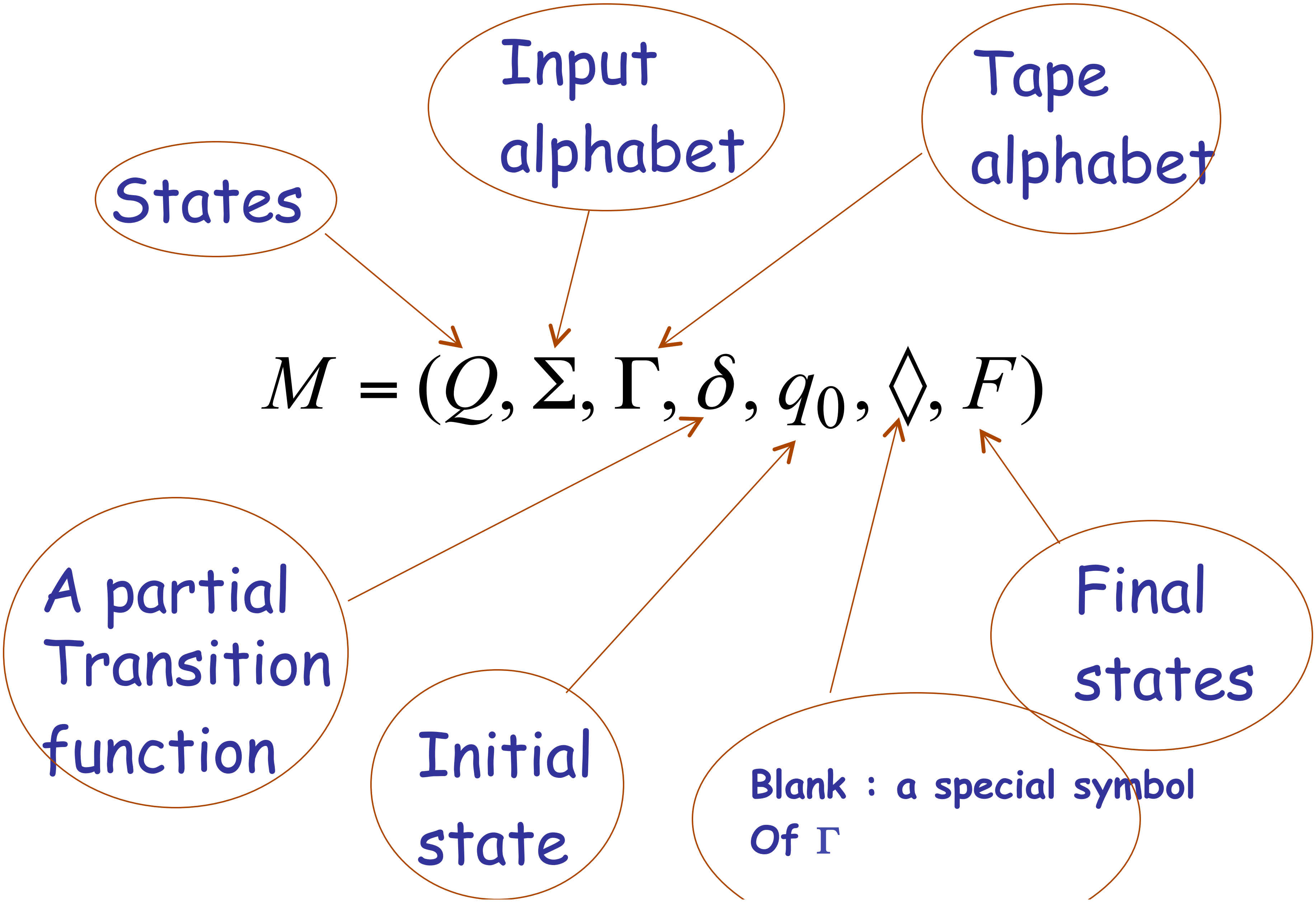
Transition Function



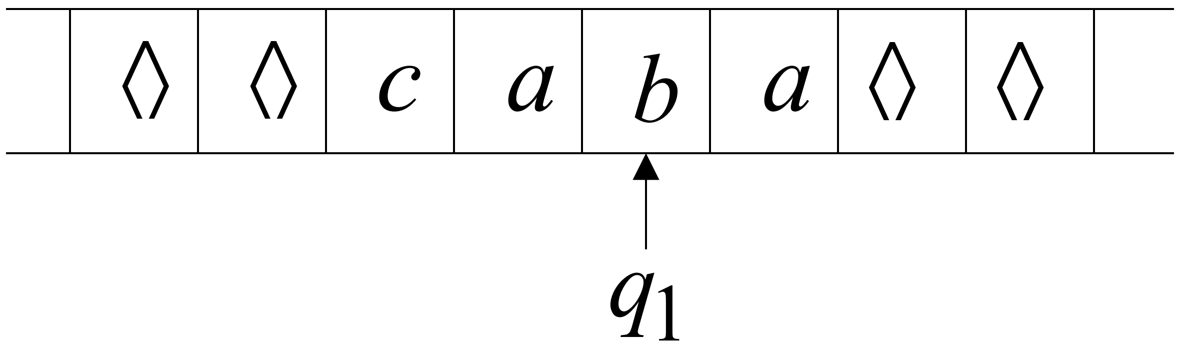
$$\delta(q_1, c) = (q_2, d, L)$$

Turing Machine:

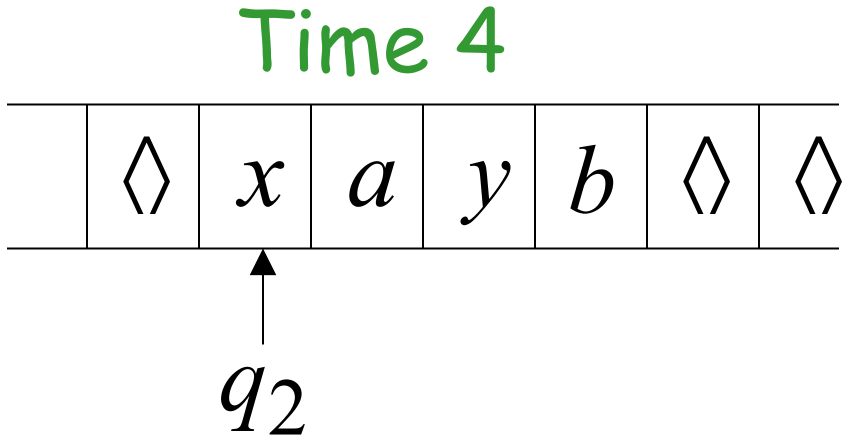
$$M = (Q, \Sigma, \Gamma, \delta, q_0, \Diamond, F)$$

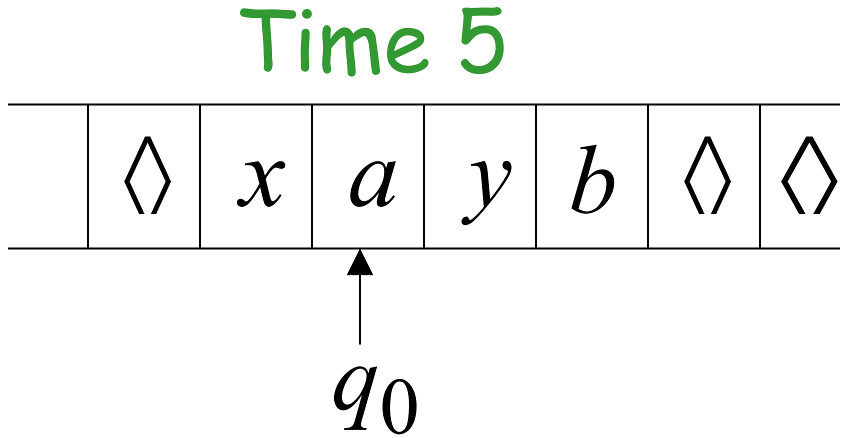


Configuration

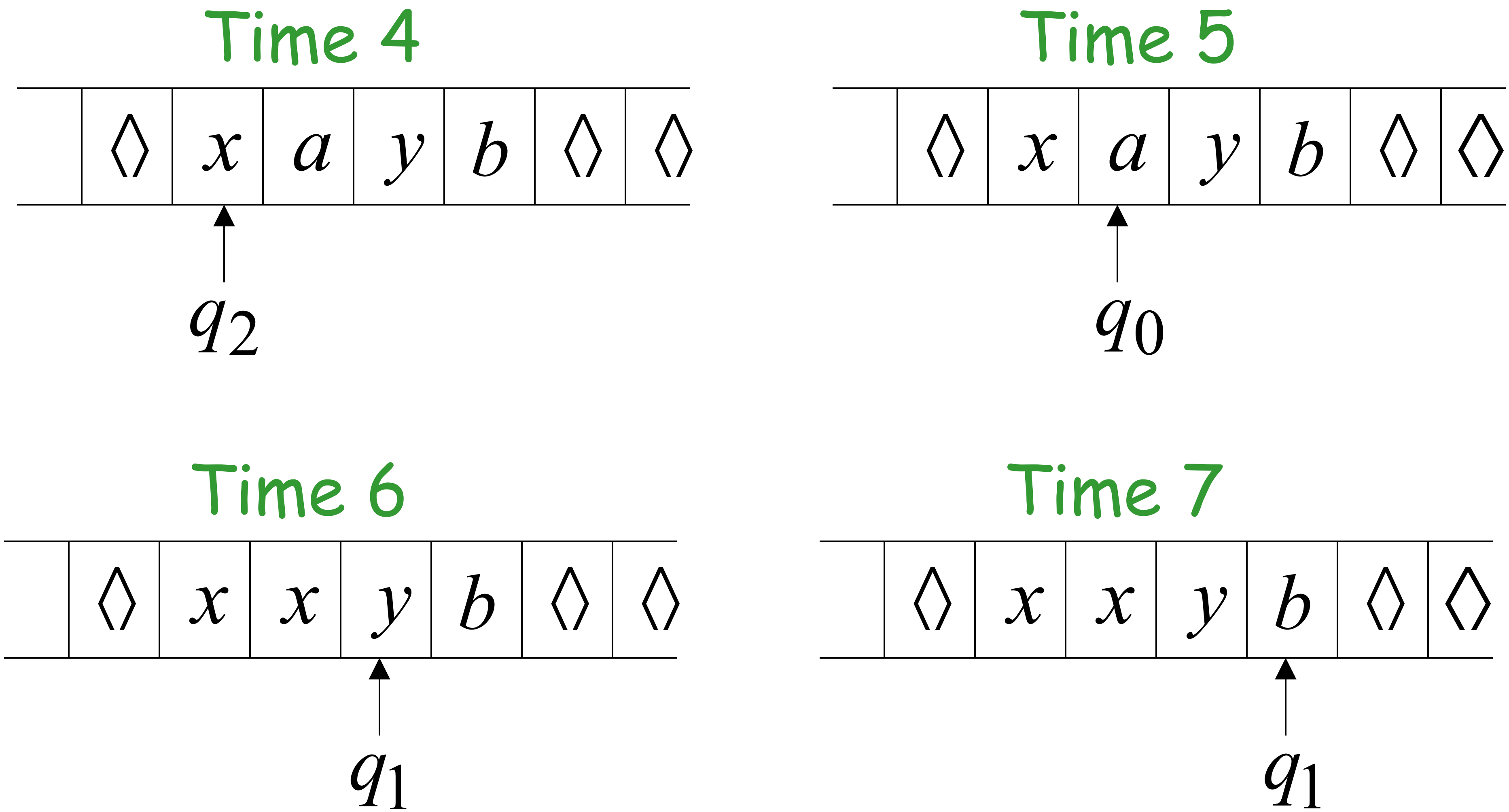


Instantaneous description: $ca\ q_1\ ba$





A Move: $q_2\ xayb \succ x\ q_0\ ayb$



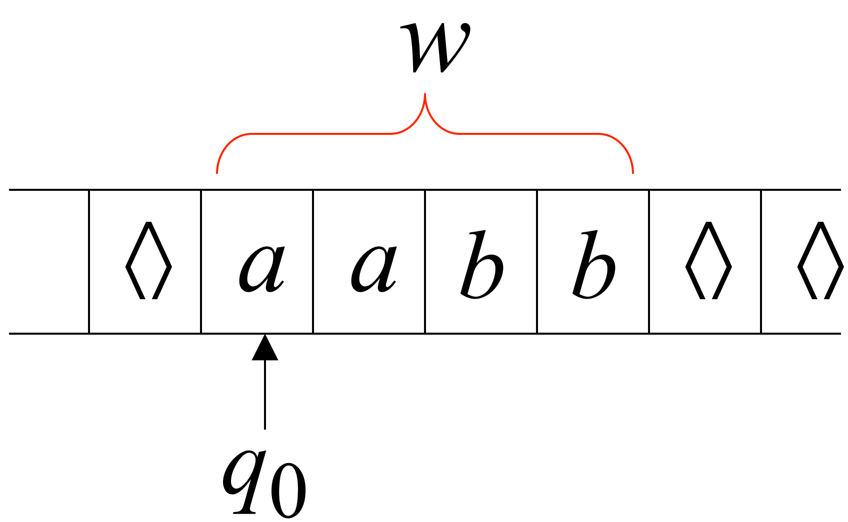
$$q_2\, xayb \succ x\, q_0\, ayb \succ xx\, q_1\, yb \succ xxy\, q_1\, b$$

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

Equivalent notation: $$q_2\,xayb \overset{*}{\succ} xxy\,q_1\,b$$

Initial configuration: $q_0 \, w$

Input string



The Accepted Language

For any Turing Machine M

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

Initial state

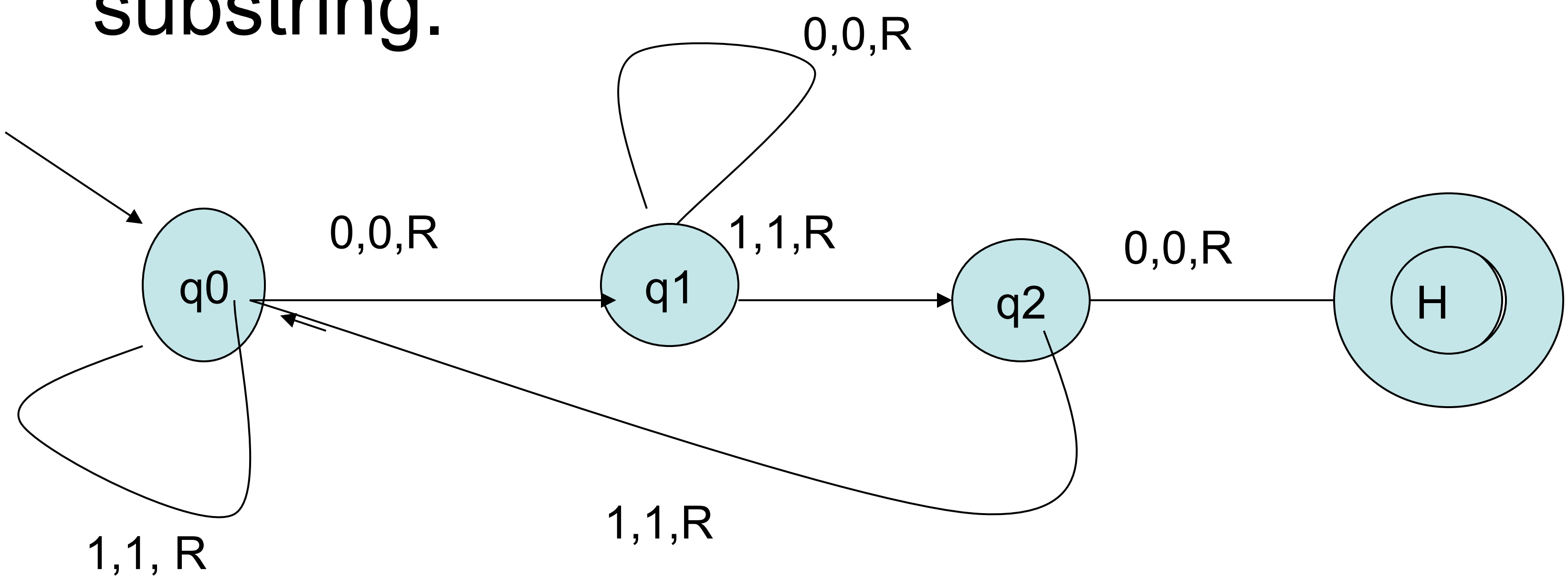
Final state

Standard Turing Machine

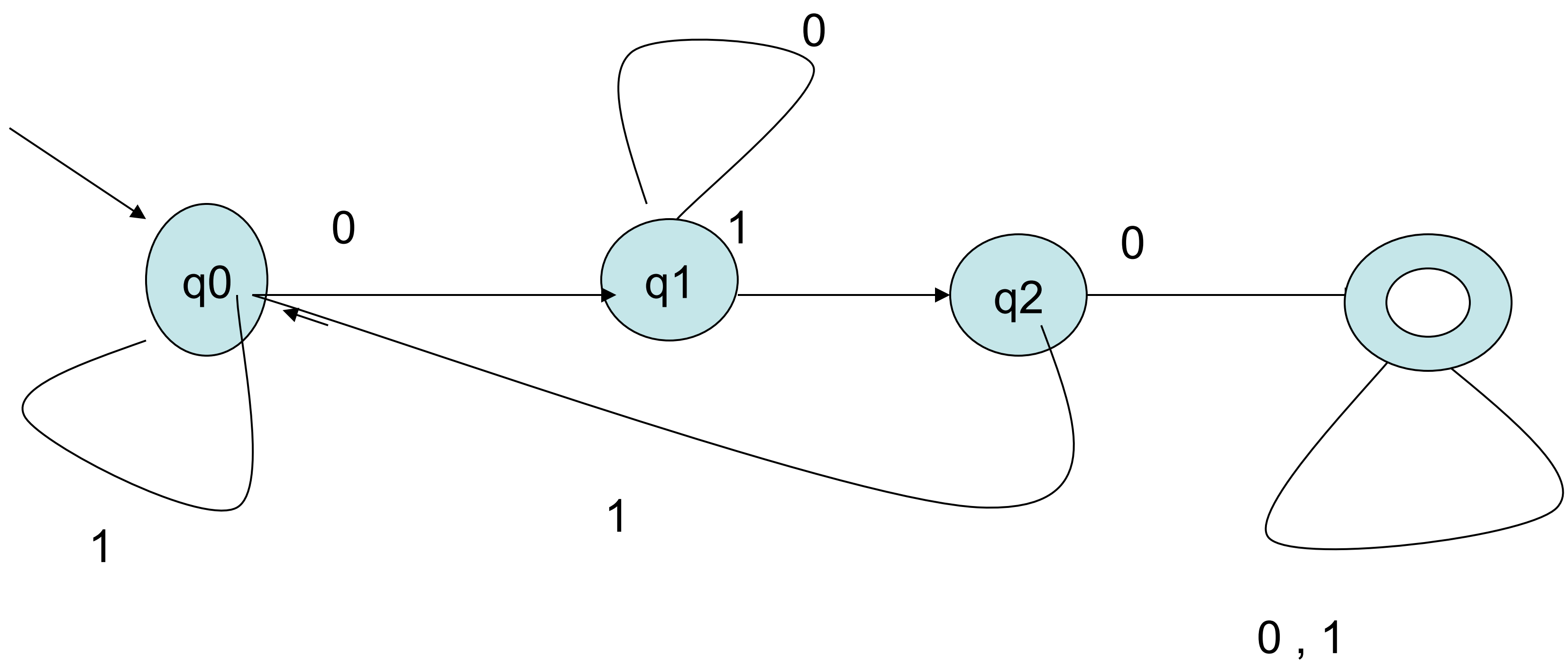
The machine we described is the standard:

- Deterministic
- Infinite tape in both directions
- Tape is the input/output file

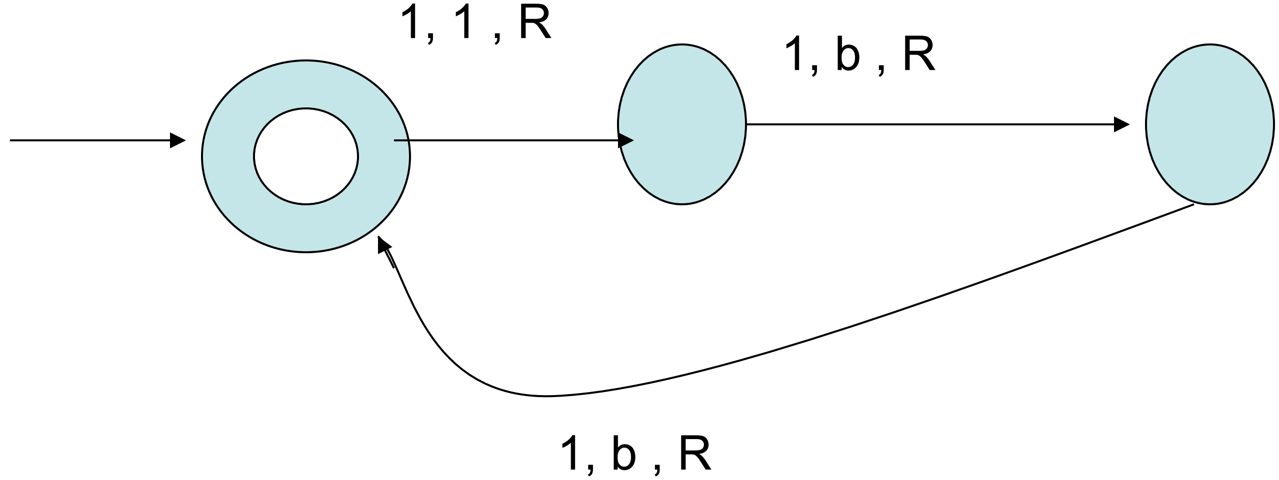
Design a Turing machine to recognize all strings in which 010 is present as a substring.



DFA for the previous language



Turing machine for odd no of 1's



Recursively Enumerable
and
Recursive

Languages

Definition:

A language is **recursively enumerable** if some Turing machine accepts it

Let L be a recursively enumerable language

and M the Turing Machine that accepts it

For string w :

if $w \in L$ then M halts in a final state

if $w \notin L$ then M halts in a non-final state

or loops forever

Definition:

A language is **recursive (decidable)** if some Turing machine accepts it and halts on any input string

In other words:

A language is recursive if there is a membership algorithm for it

Let L be a recursive language

and M the Turing Machine that accepts it

For string w:

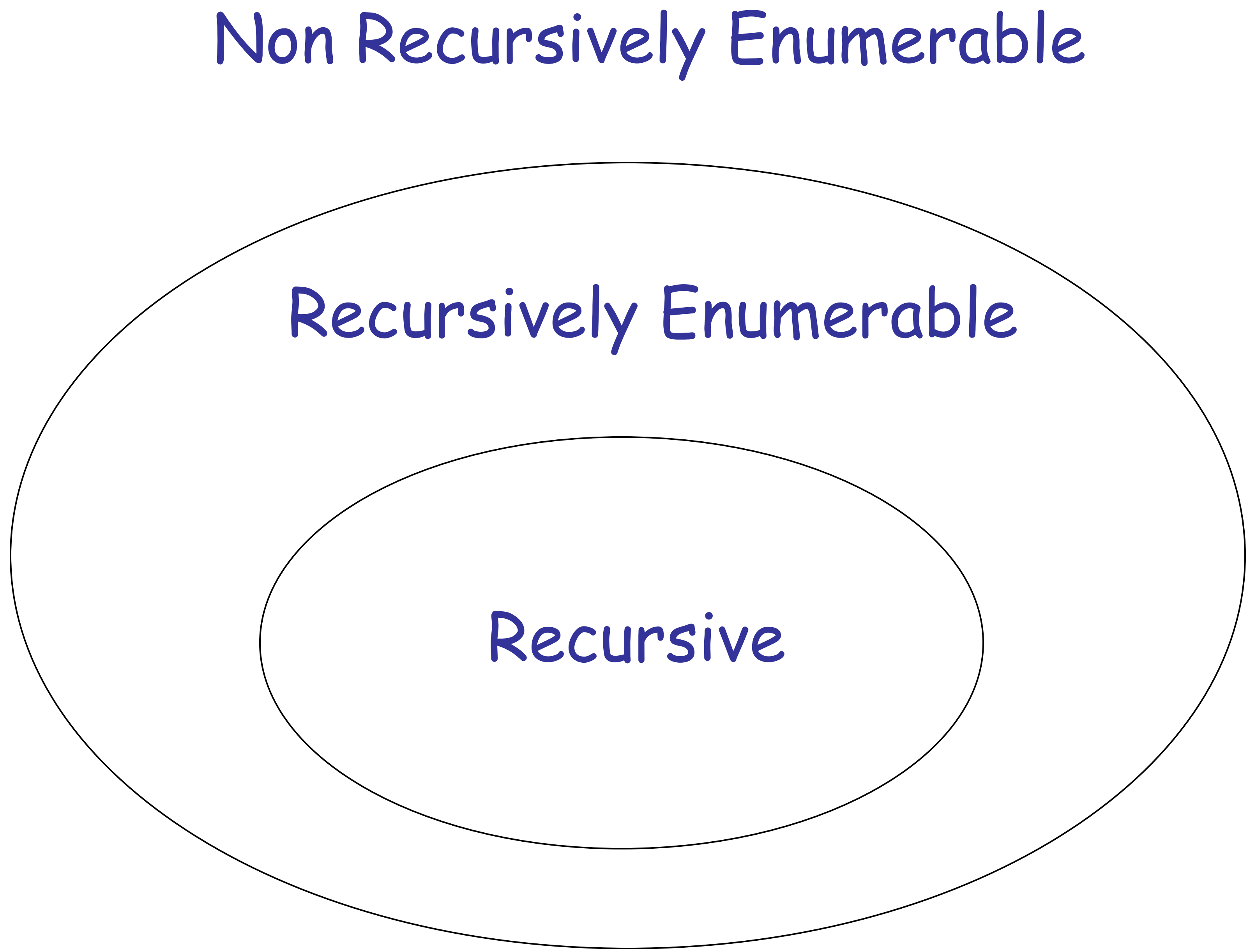
if $w \in L$ then M halts in a final state

if $w \notin L$ then M halts in a non-final state

We can prove:

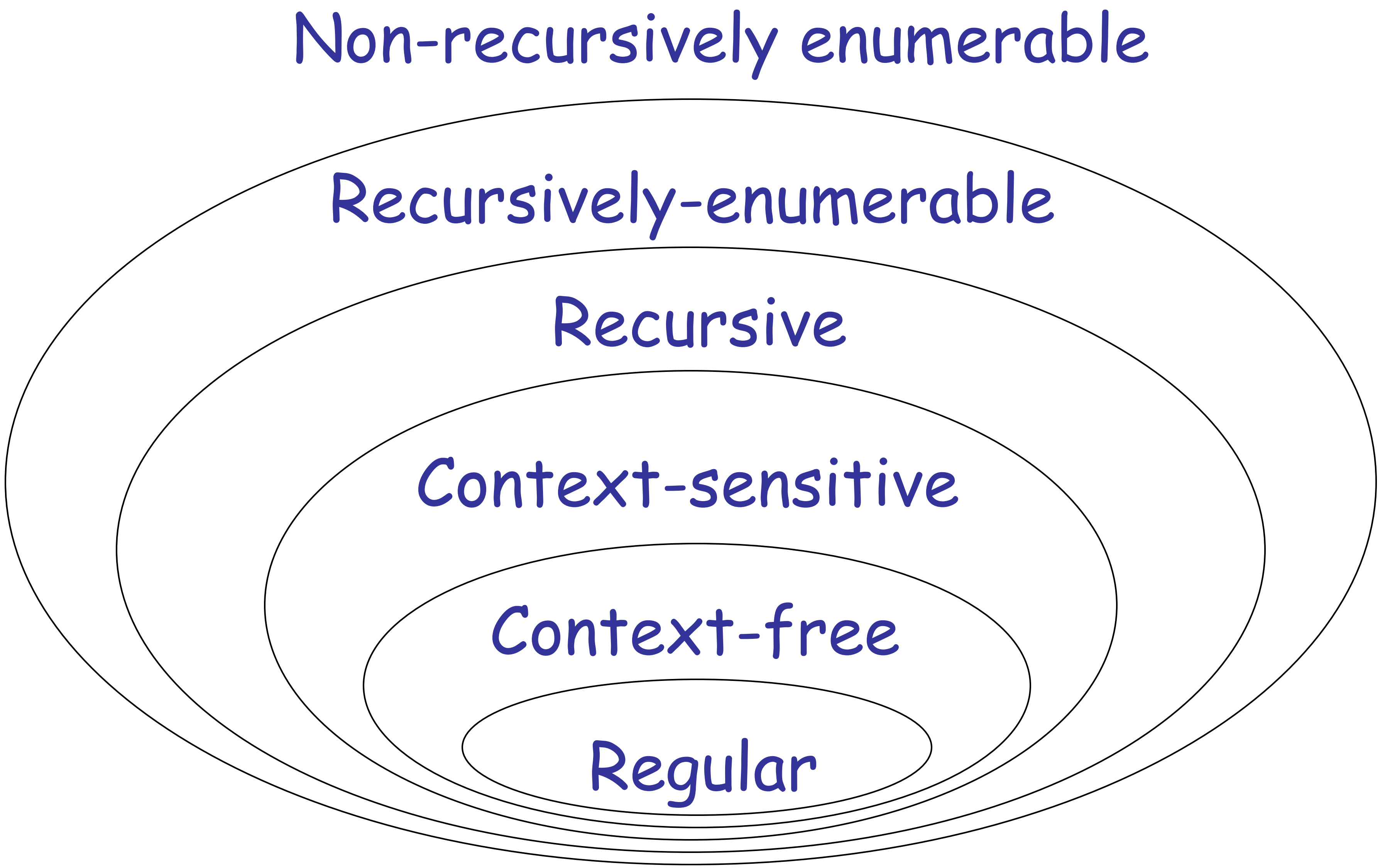
1. There is a specific language which is not recursively enumerable (not accepted by any Turing Machine)

2. There is a specific language which is recursively enumerable but not recursive


Non Recursively Enumerable
Recursively Enumerable
Recursive

The Chomsky Hierarchy

The Chomsky Hierarchy



The Church-Turing thesis

Any intuitive notion of algorithms is
equivalent
to TM algorithms