CSC 339 – Theory of Computation
Fall 2023

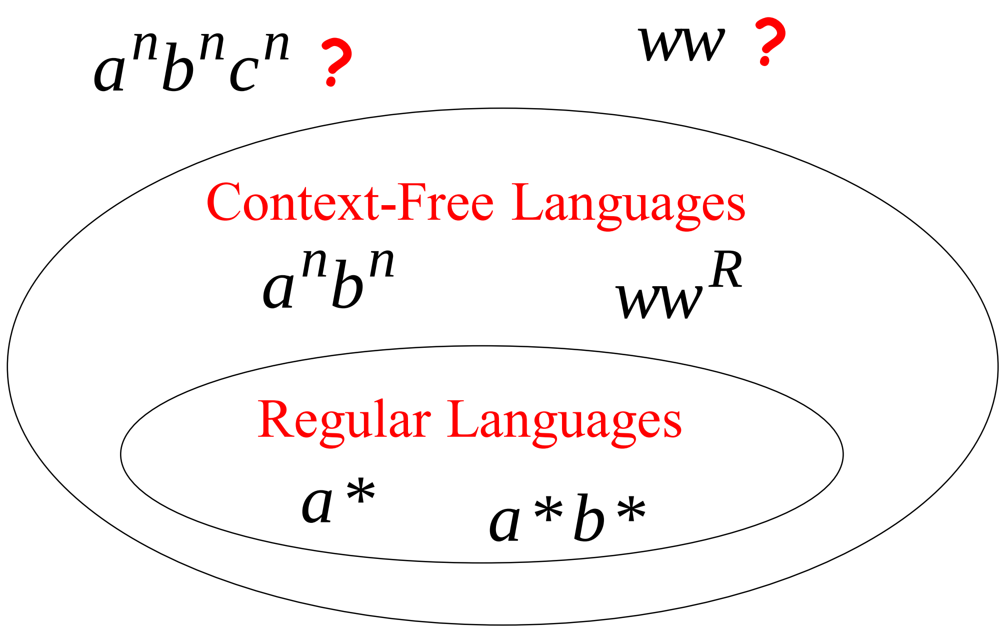
10. Turing Machines

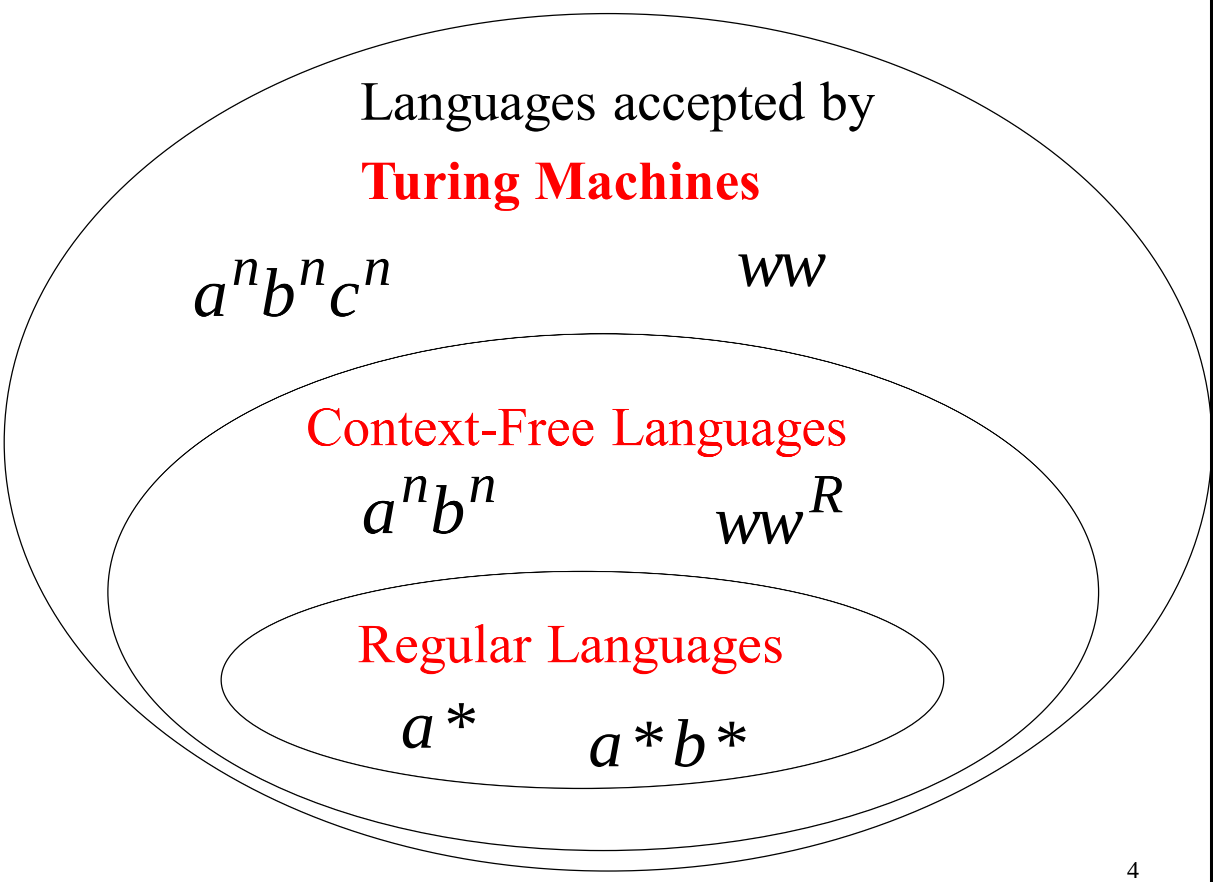
Outline

- Turing machines
- Formal definitions for Turing machines
- Computing functions with Turing machines
- Combining Turing machines

2

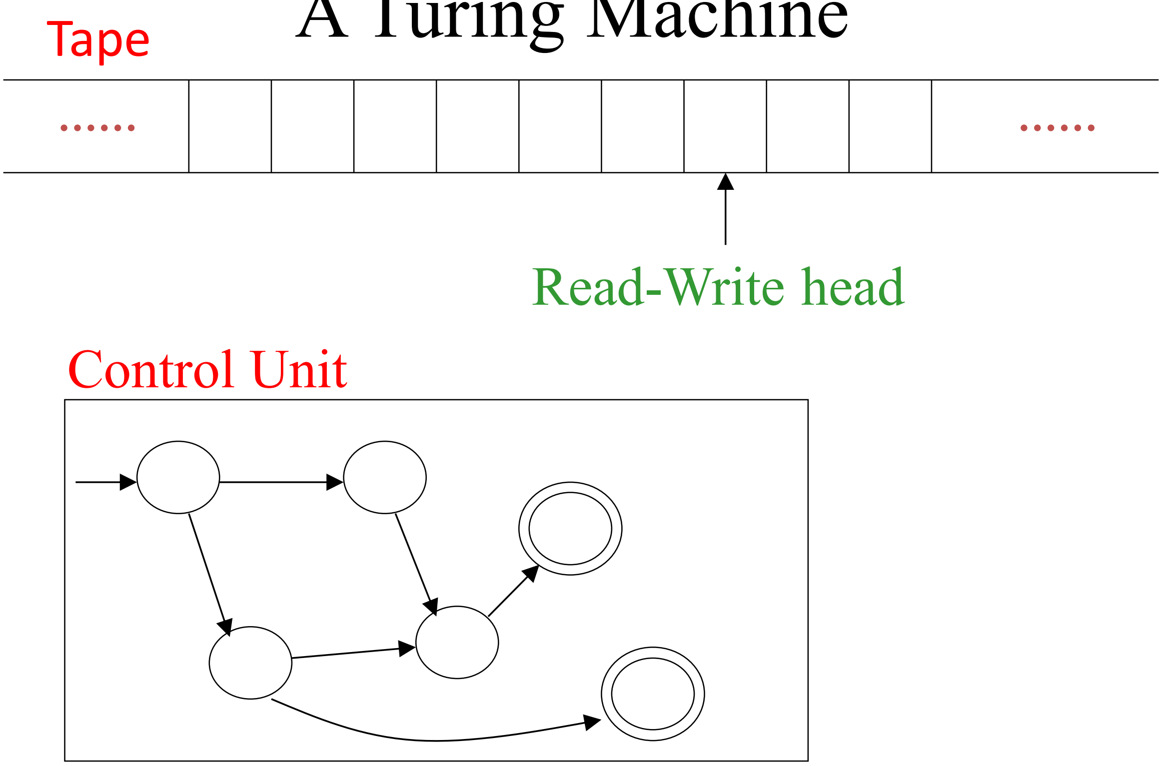
The Language Hierarchy




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

4

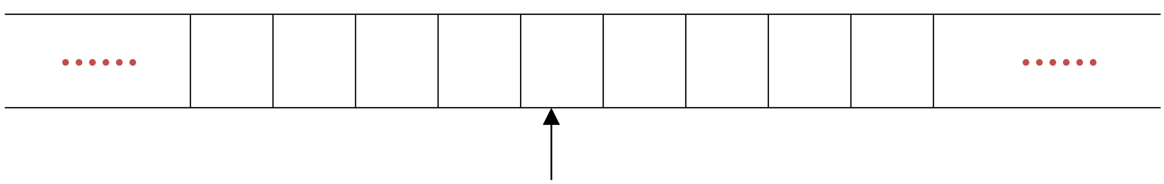
A Turing Machine



5

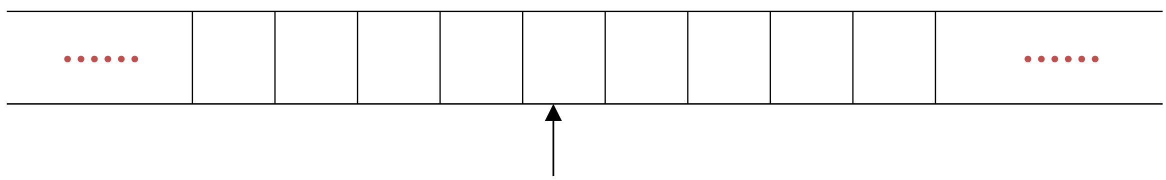
The Tape

No boundaries: infinite length



Read-Write head

The head moves Left or Right



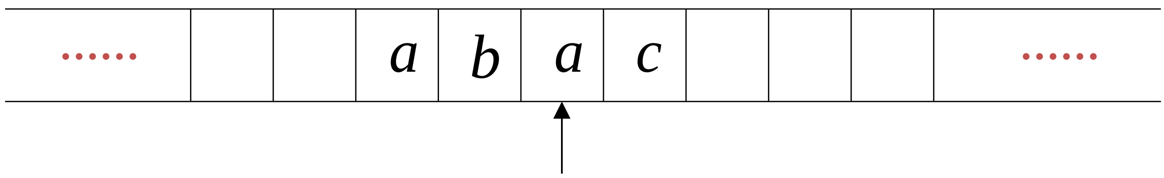
Read-Write head

The head at each transition (time step):

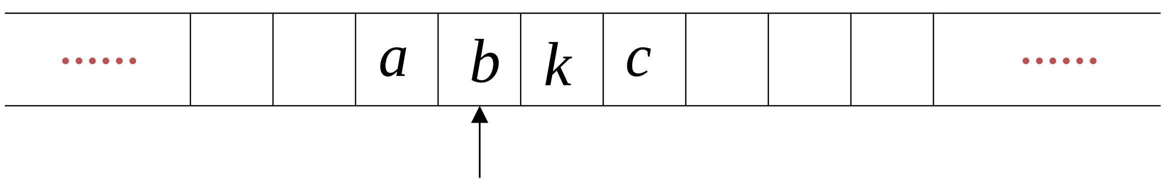
1. Reads a symbol
2. Writes a symbol
3. Moves Left or Right

Example:

Time 0

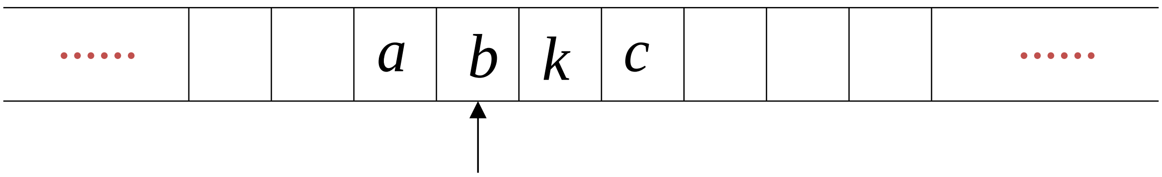


Time 1

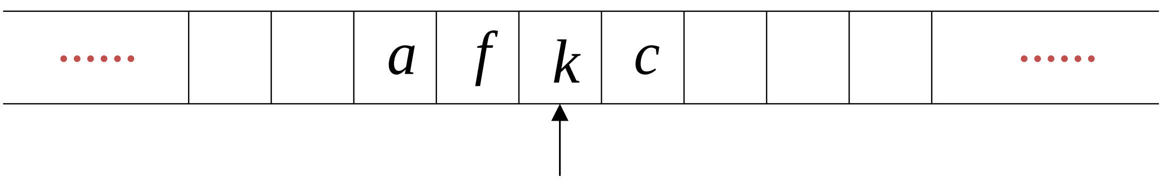


1. Reads a
2. Writes k
3. Moves Left

Time 1

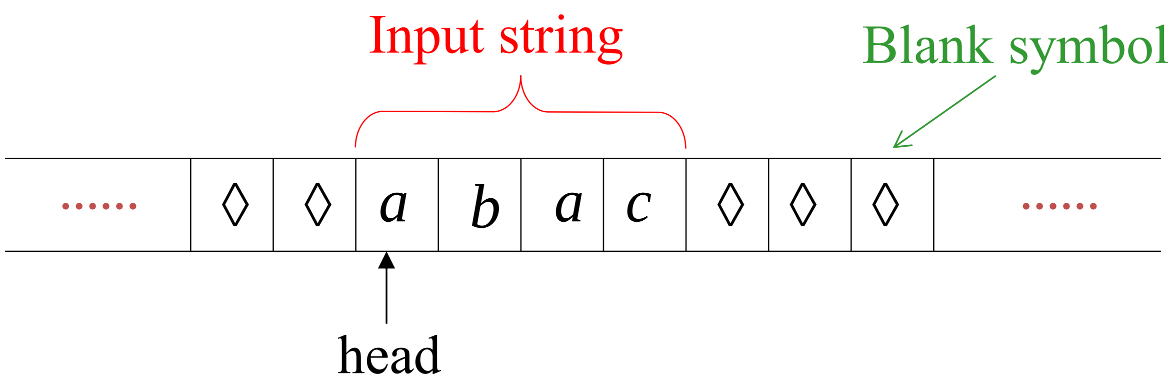


Time 2



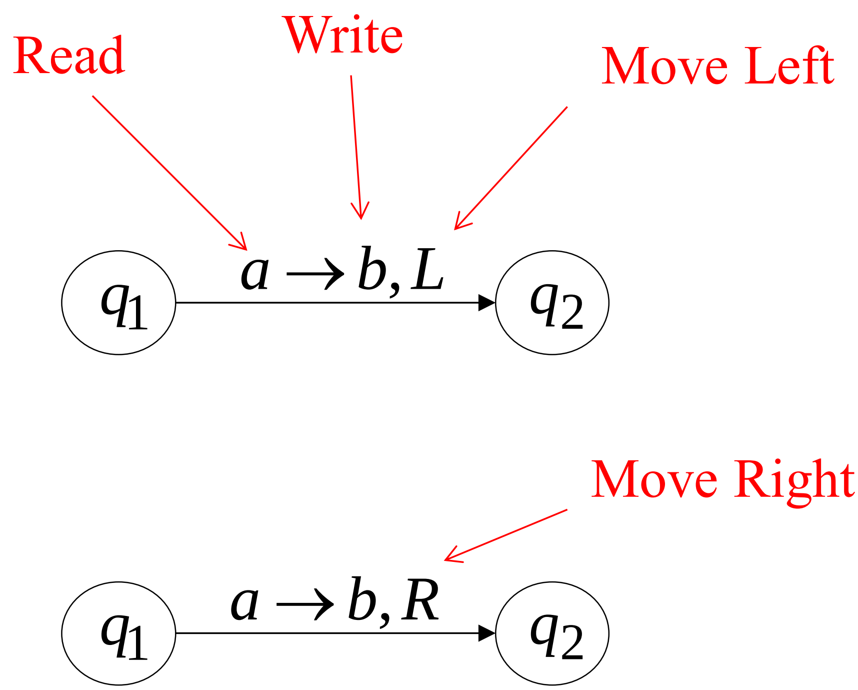
1. Reads b
2. Writes f
3. Moves Right

The Input String



Head starts at the leftmost position of the input string

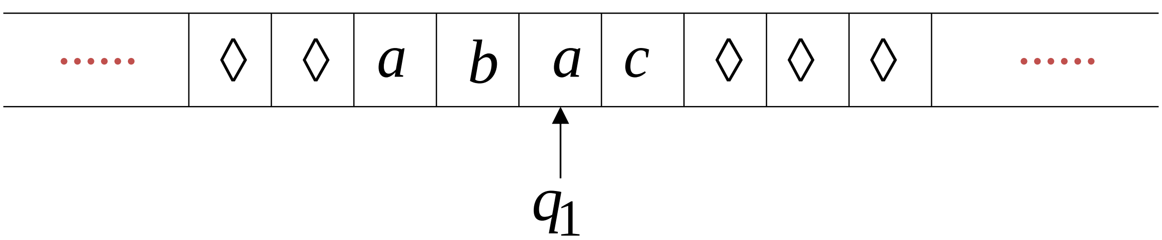
States & Transitions



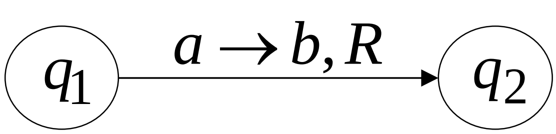
11

Example:

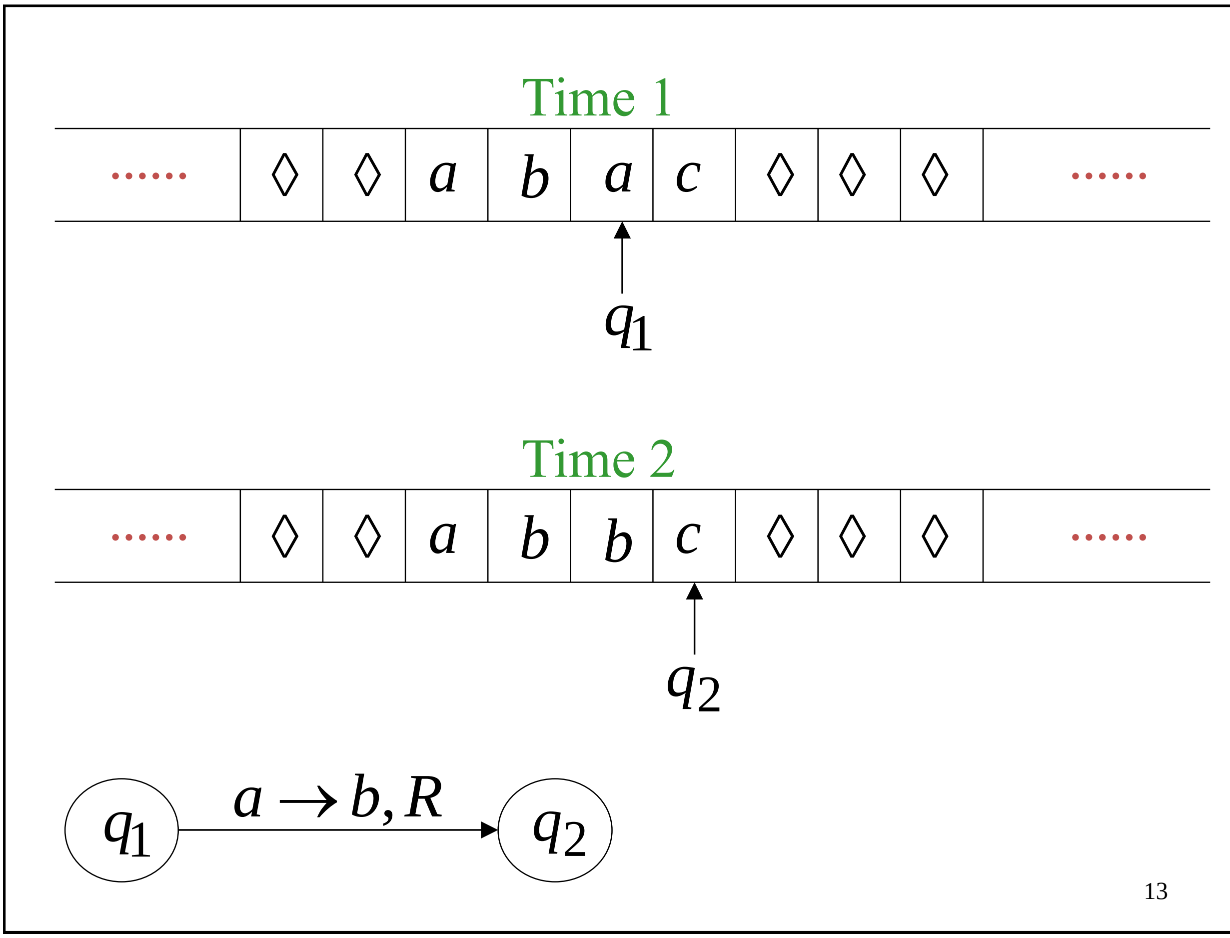
Time 1



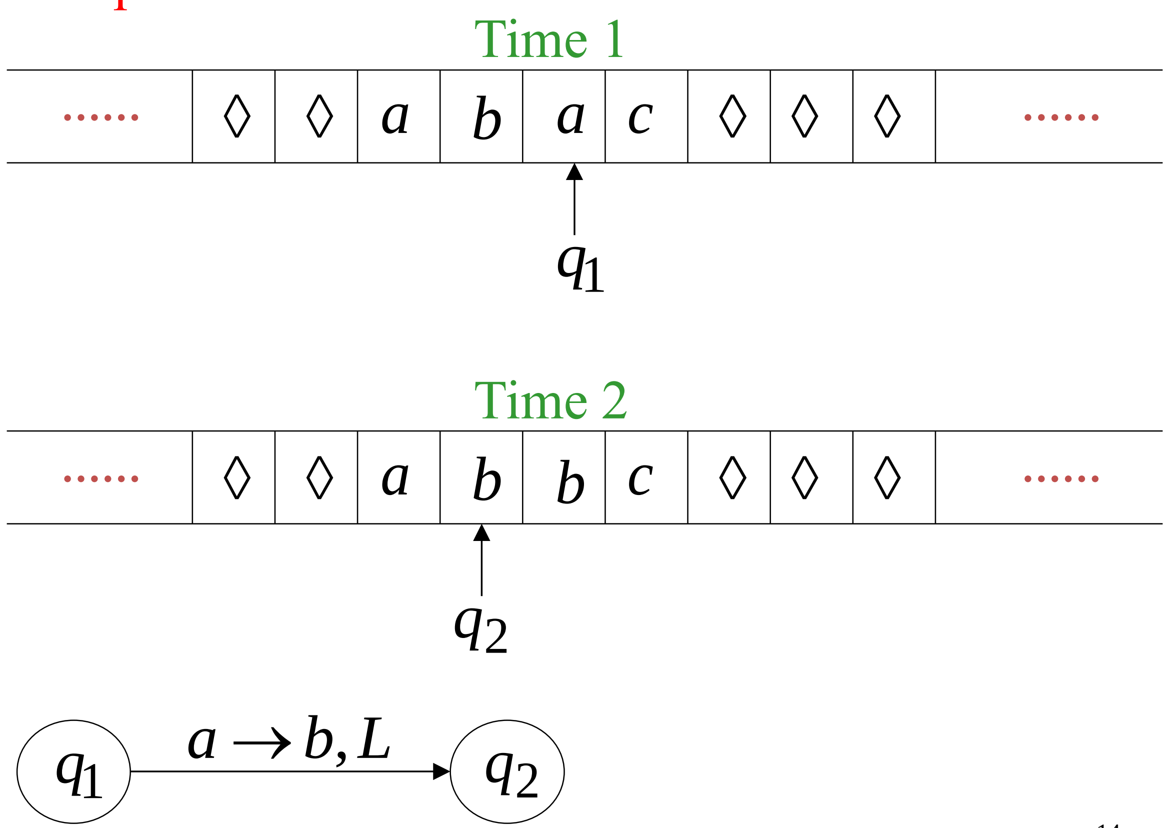
current state



12

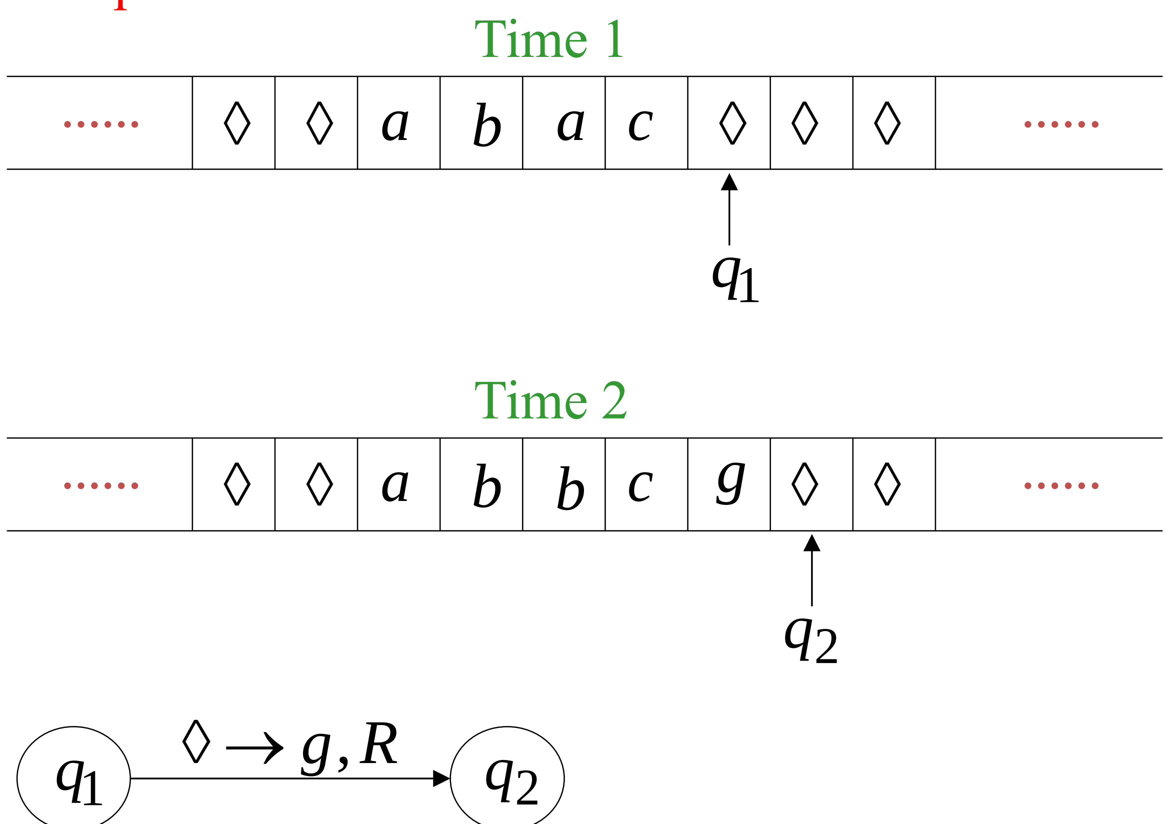

Time 1
◊ ◊ a b a c ◊ ◊ ◊
q1
Time 2
◊ ◊ a b b c ◊ ◊ ◊
q2
q1
a → b, R
q2
13

Example:



14

Example:

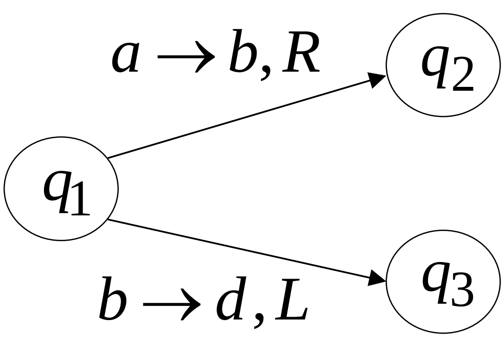


15

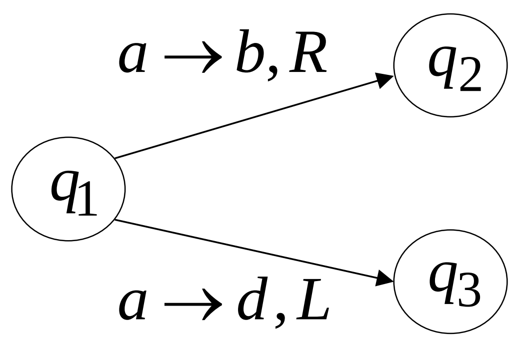
Determinism

Turing Machines are deterministic

Allowed



Not Allowed

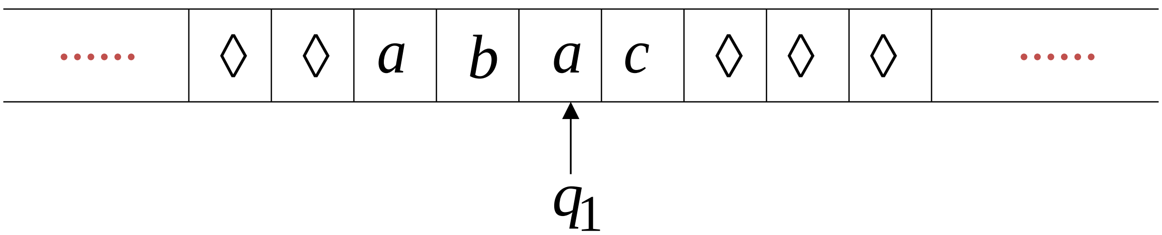


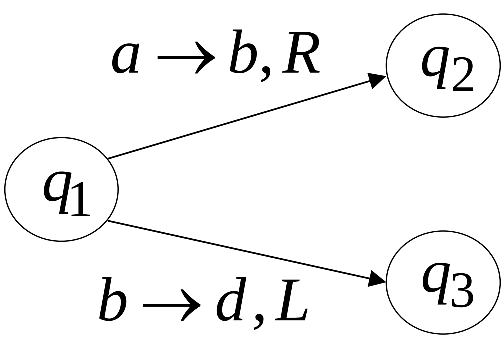
No epsilon transitions allowed

16

Partial Transition Function

Example:





Allowed:

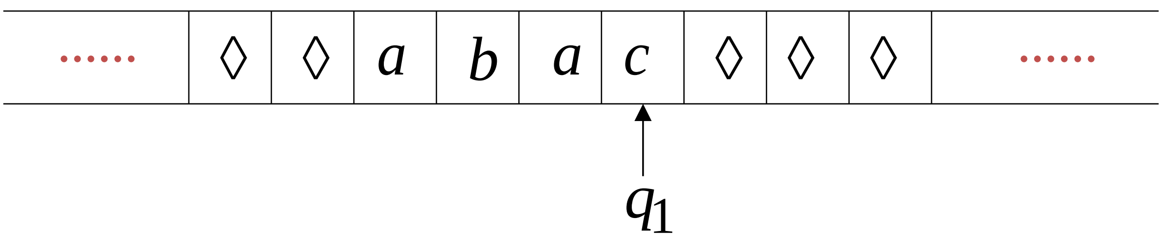
No transition

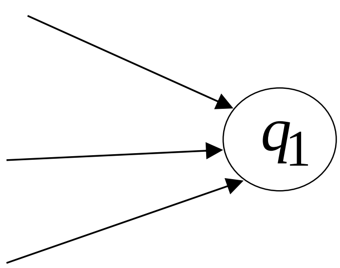
for input symbol c

Halting

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

Halting Example 1:

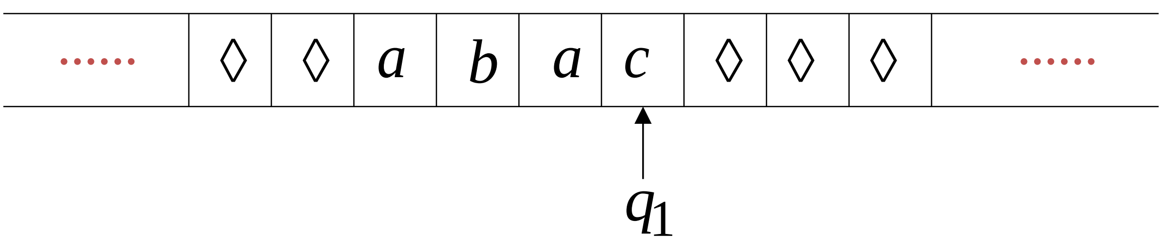


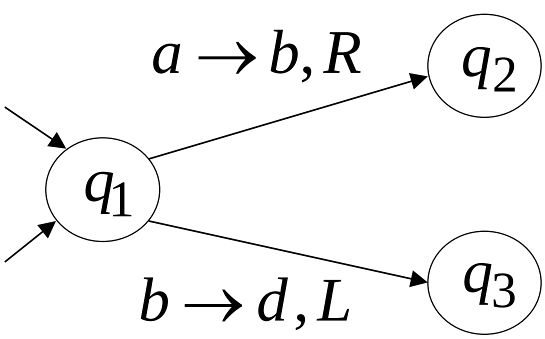


No transition from q_1

HALT!

Halting Example 2:



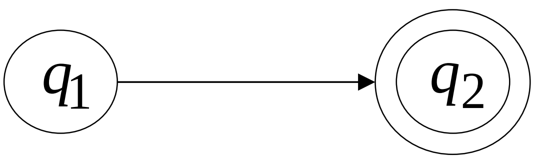


No possible transition from q_1 and symbol c

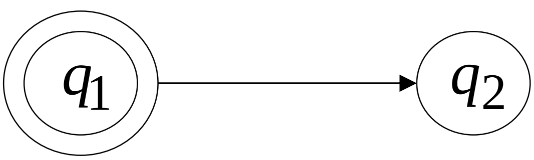
HALT!

20

Accepting States



Allowed



Not Allowed

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

Acceptance

Accept Input String → If machine halts in an accept state

Reject Input String → If machine halts in a non-accept state

or

If machine enters an *infinite loop*

22

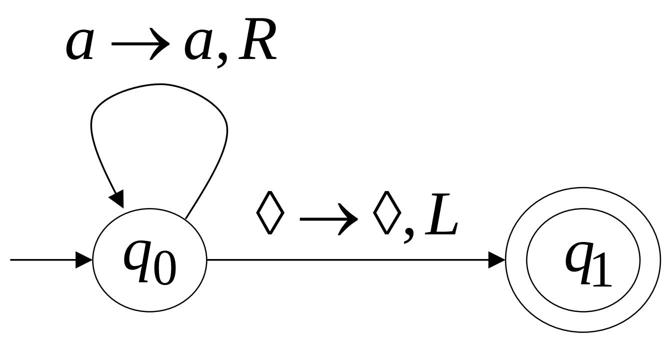
Observation:

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

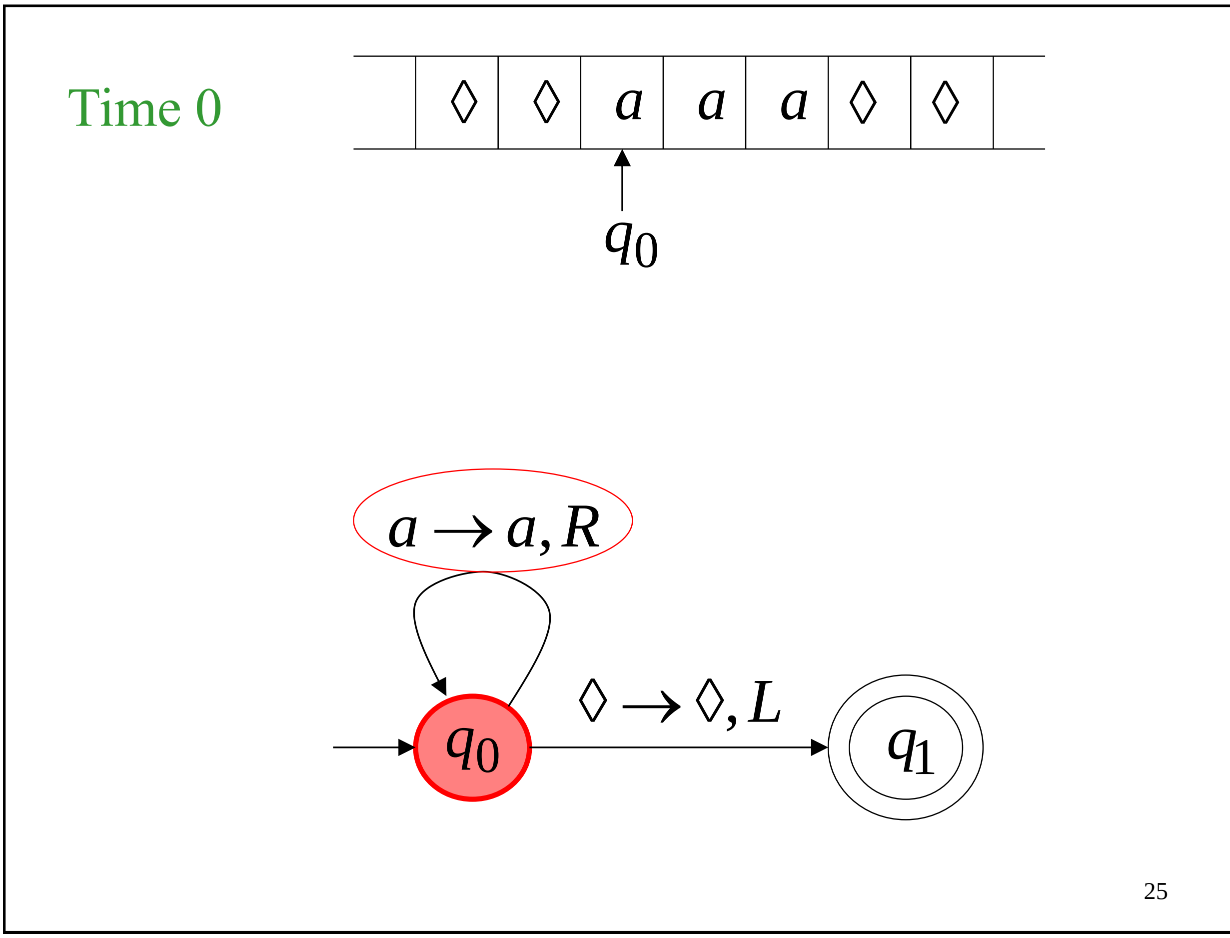
Turing Machine Example

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

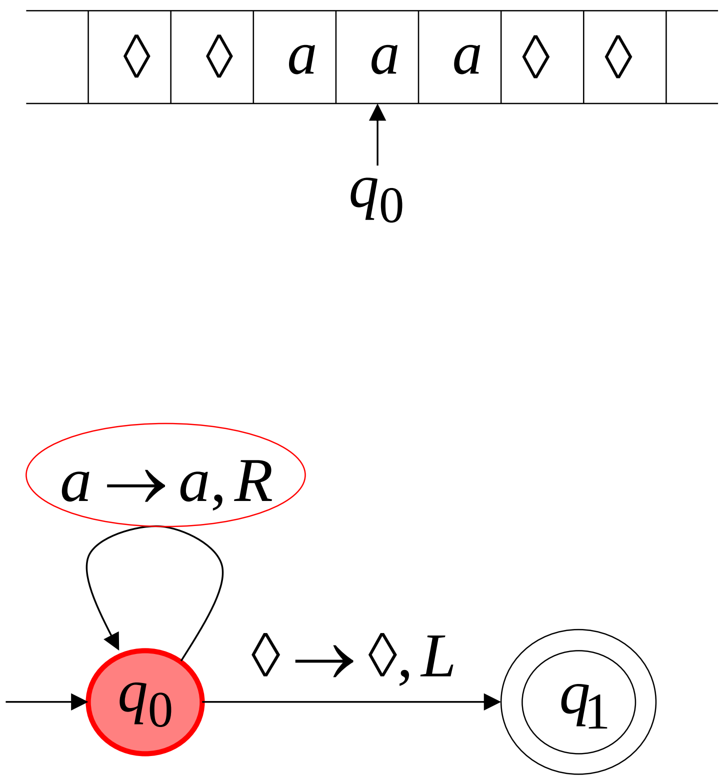
Accepts the language: a^*



24


Time 0
◊
◊
a
a
a
◊
◊
q_0
$a \rightarrow a, R$
$◊ \rightarrow ◊, L$
q_0
q_1
25

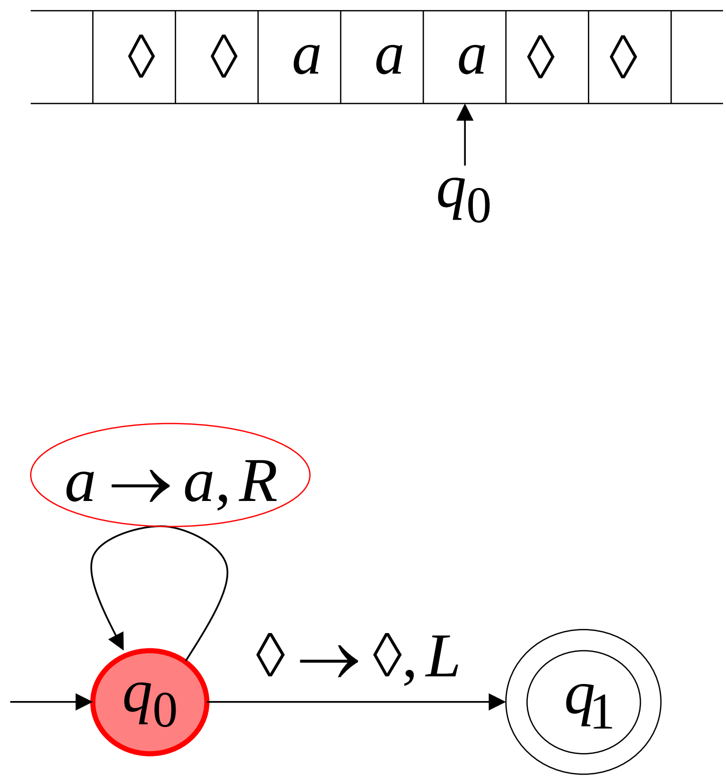
Time 1



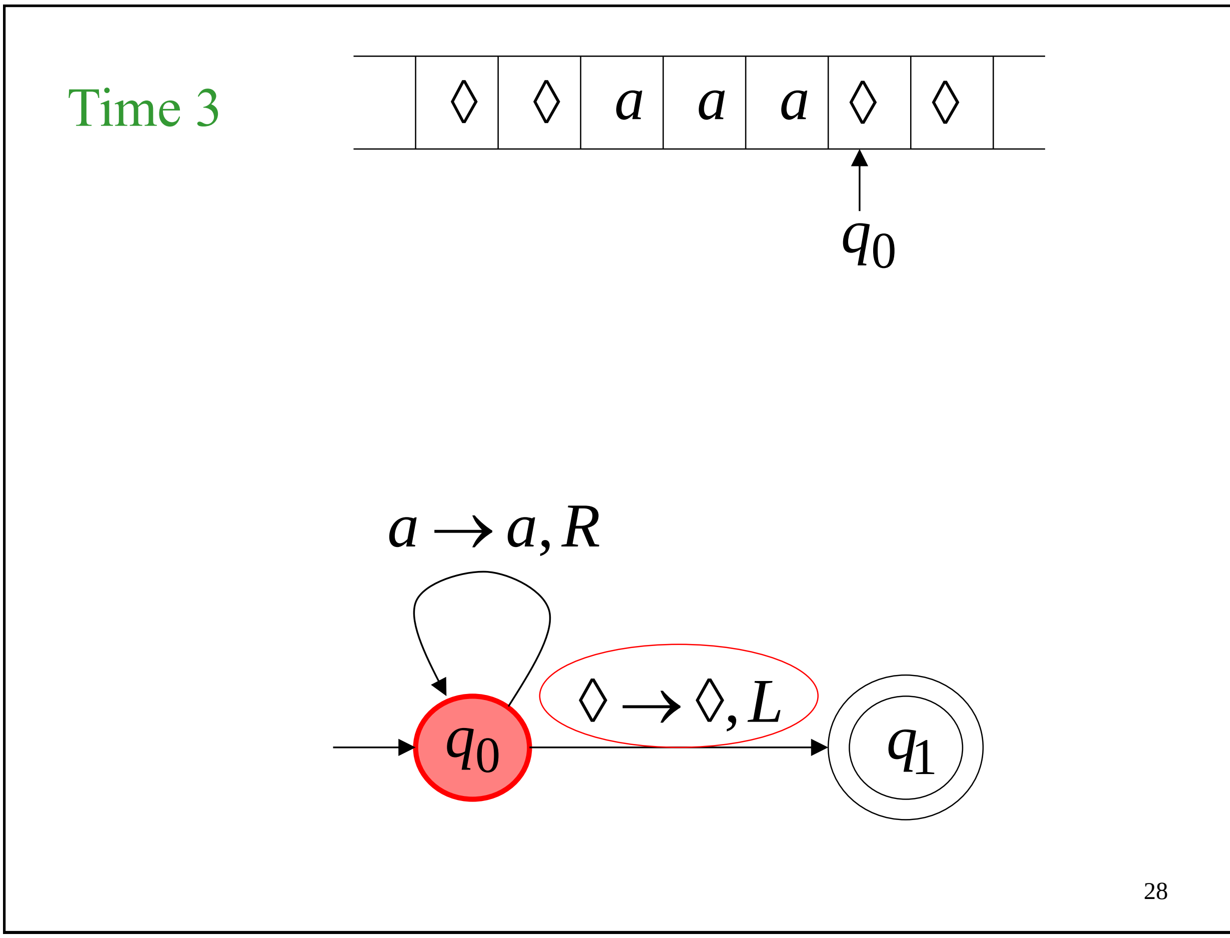
$\Diamond$ $\Diamond$ a a a $\Diamond$ $\Diamond$
q_0
$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

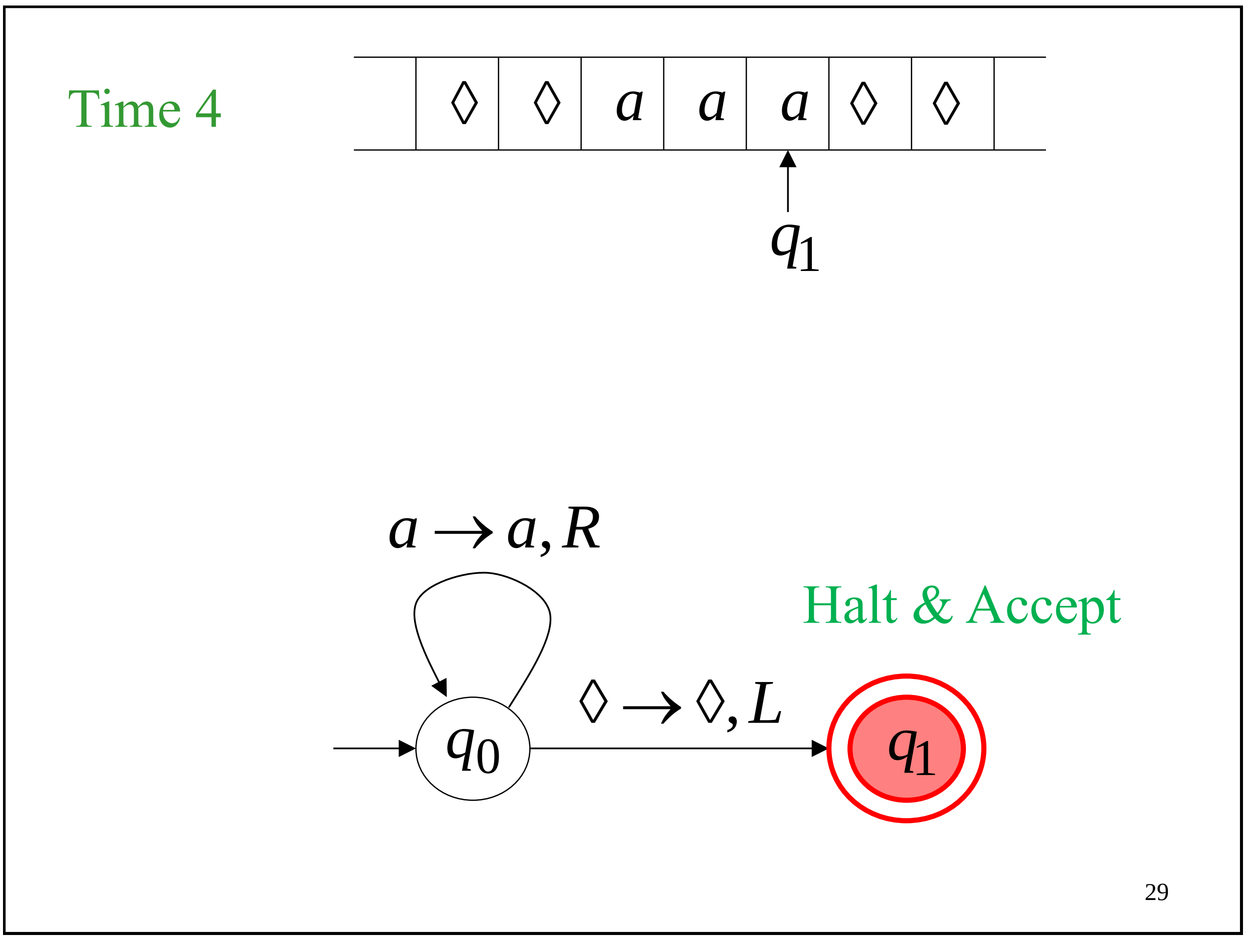
26

Time 2



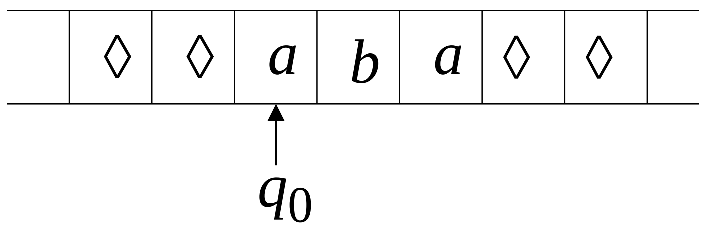
27

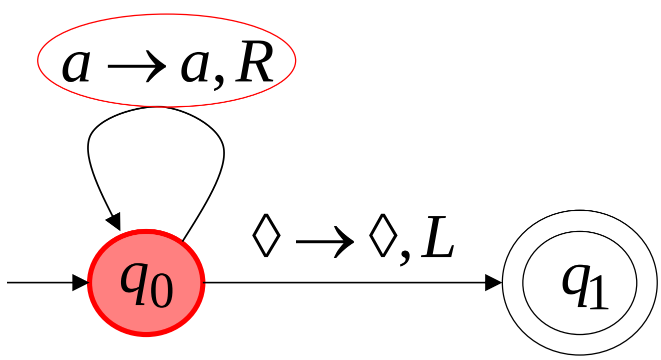

Time 3
◊
◊
a
a
a
◊
◊
q_0
$a \rightarrow a, R$
$\lozenge \rightarrow \lozenge, L$
q_0
q_1
28


Time 4
◊ ◊ a a a ◊ ◊
q1
a → a, R
Halt & Accept
◊ → ◊, L
q0
q1
29

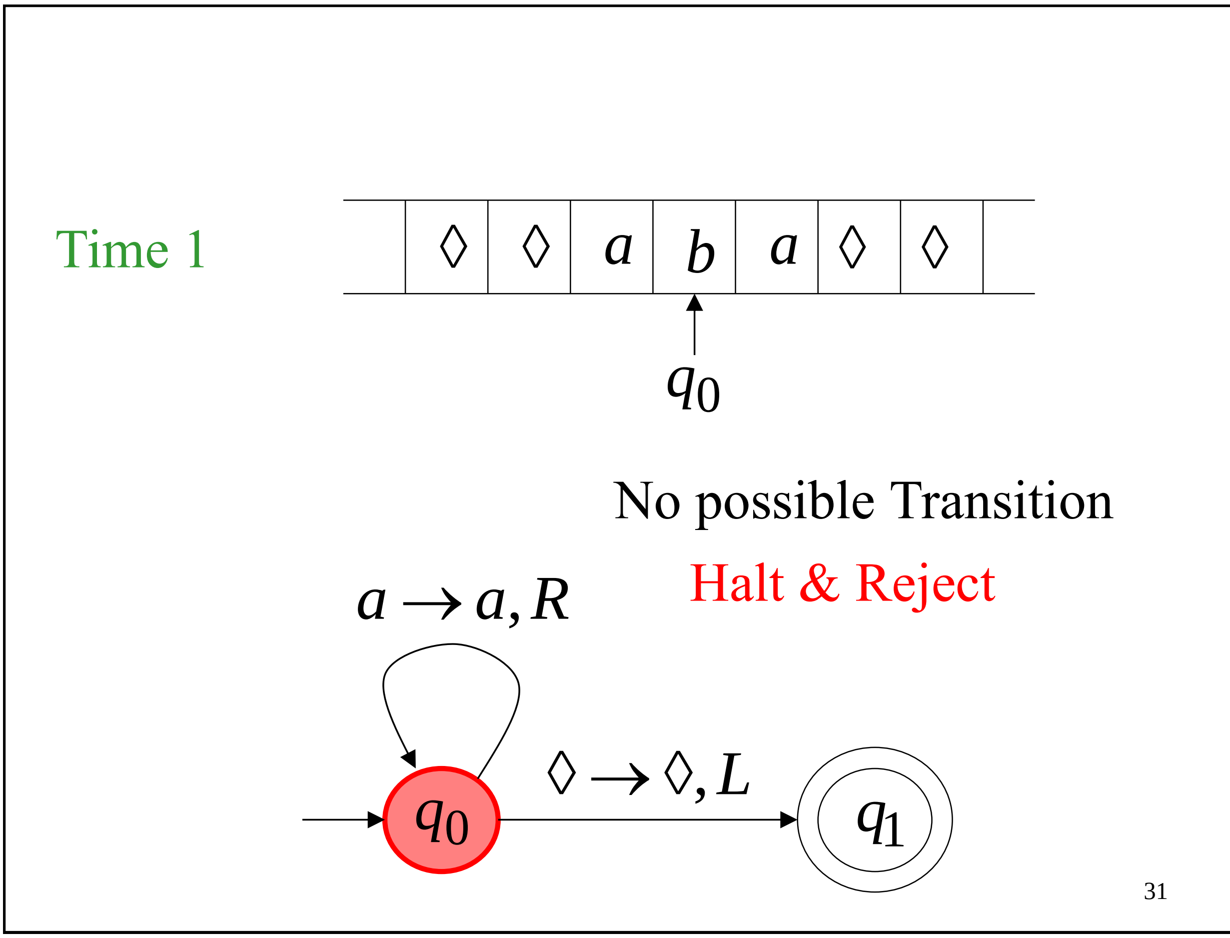
Rejection Example

Time 0





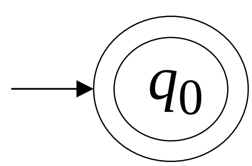
30


Time 1
◊
◊
a
b
a
◊
◊
q_0
No possible Transition
Halt & Reject
$a \rightarrow a, R$
◊ $\rightarrow$ ◊$, L$
q_0
q_1
31

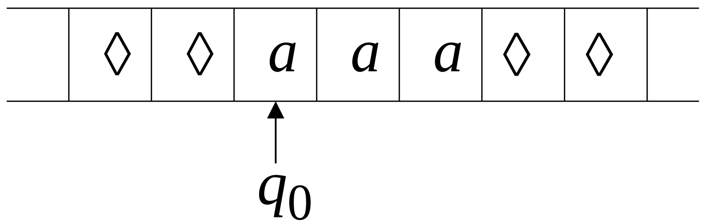
A simpler machine for the same language

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

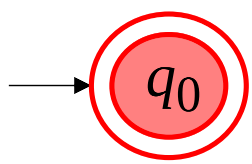
Accepts the language: a^*



Time 0



Halt & Accept

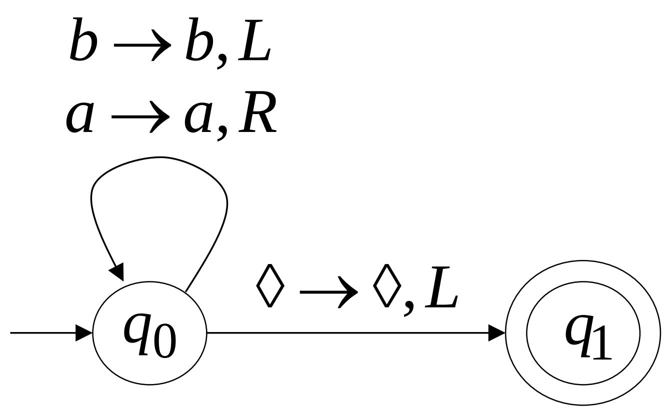


Not necessary to scan the input

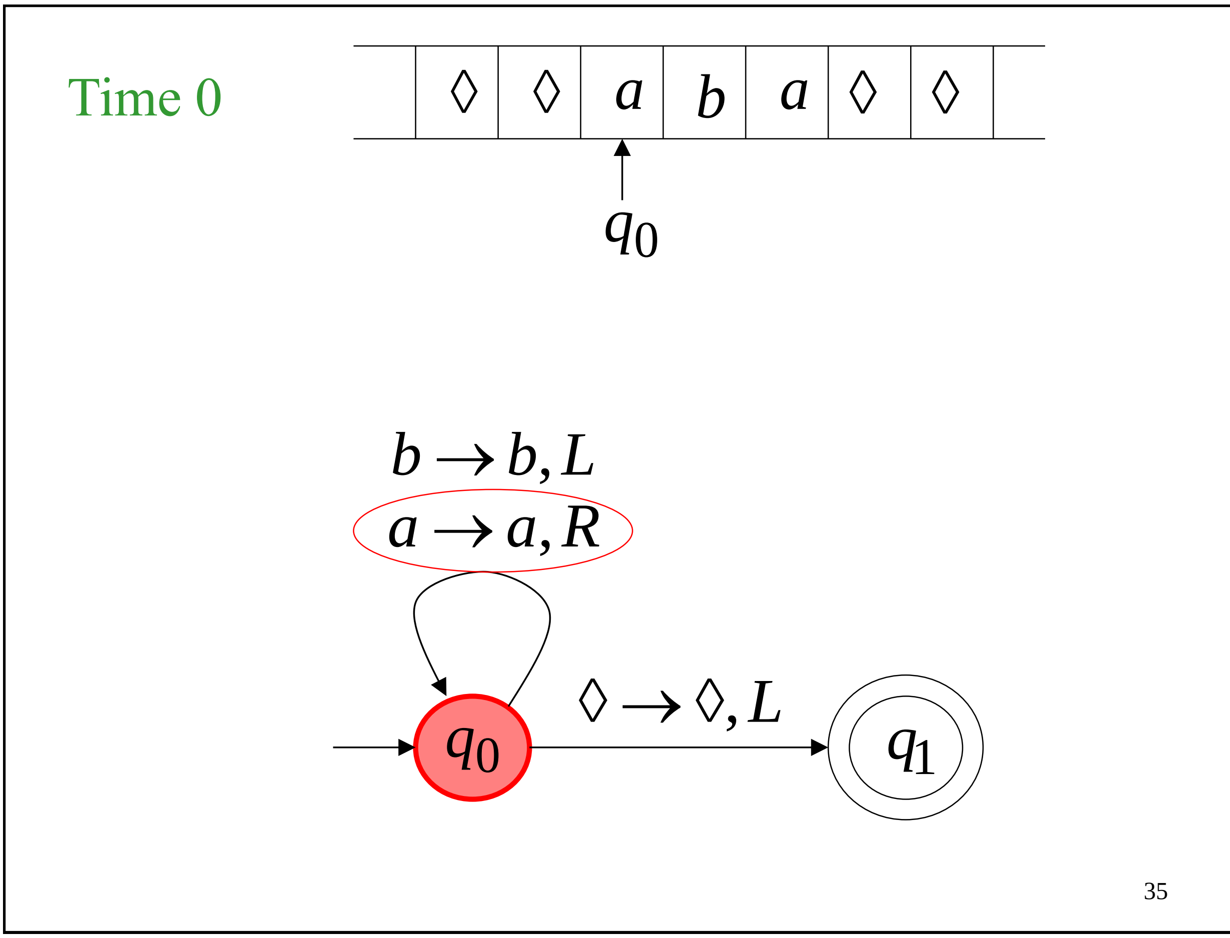
33

Infinite Loop Example

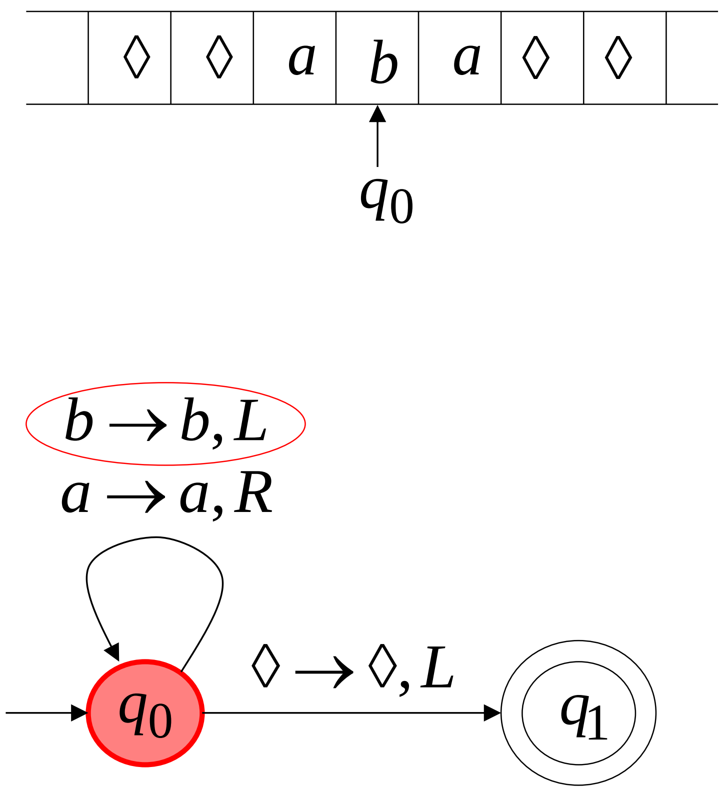
Turing machine:



34

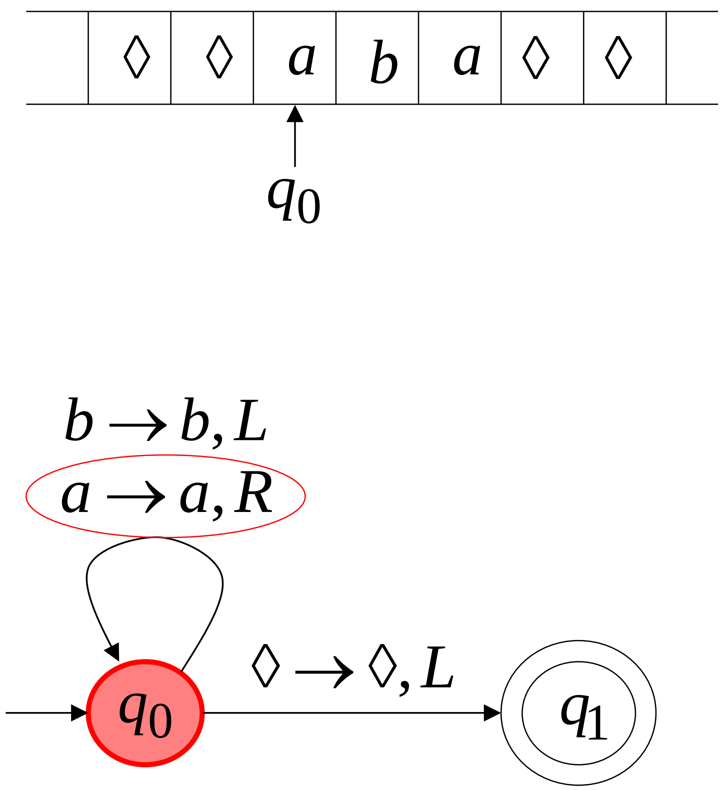

Time 0
◊
◊
a
b
a
◊
◊
q0
b → b, L
a → a, R
◊ → ◊, L
q0
q1
35

Time 1

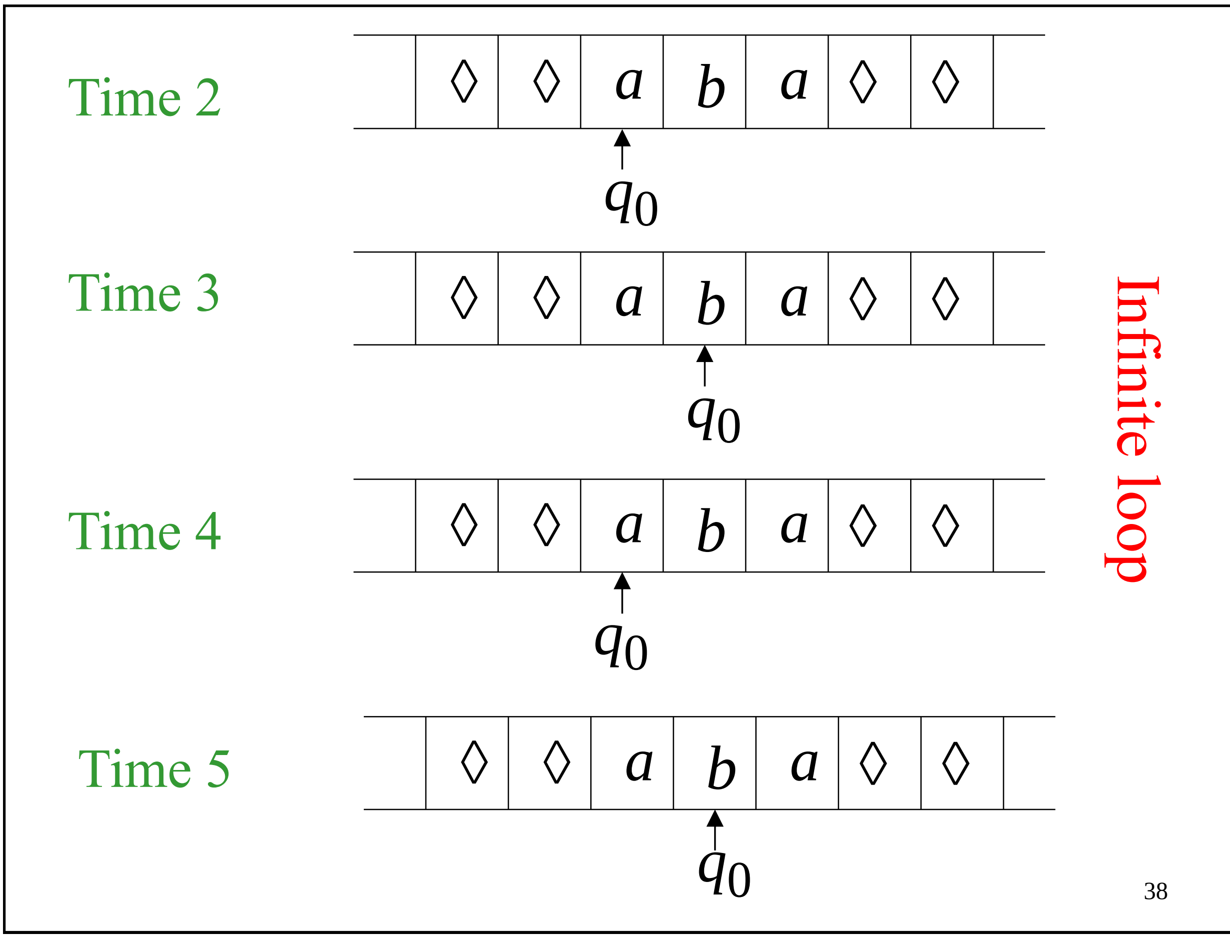

◊ ◊ a b a ◊ ◊
q_0
$b \rightarrow b, L$
$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

36

Time 2


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

37


Time 2
◊ ◊ a b a ◊ ◊
q_0
Time 3
◊ ◊ a b a ◊ ◊
q_0
Time 4
◊ ◊ a b a ◊ ◊
q_0
Time 5
◊ ◊ a b a ◊ ◊
q_0
Infinite loop
38

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 $\{a^n b^n \mid n \geq 1\}$

Basic Idea:

Match a's with b's:

Repeat:

replace leftmost a with x

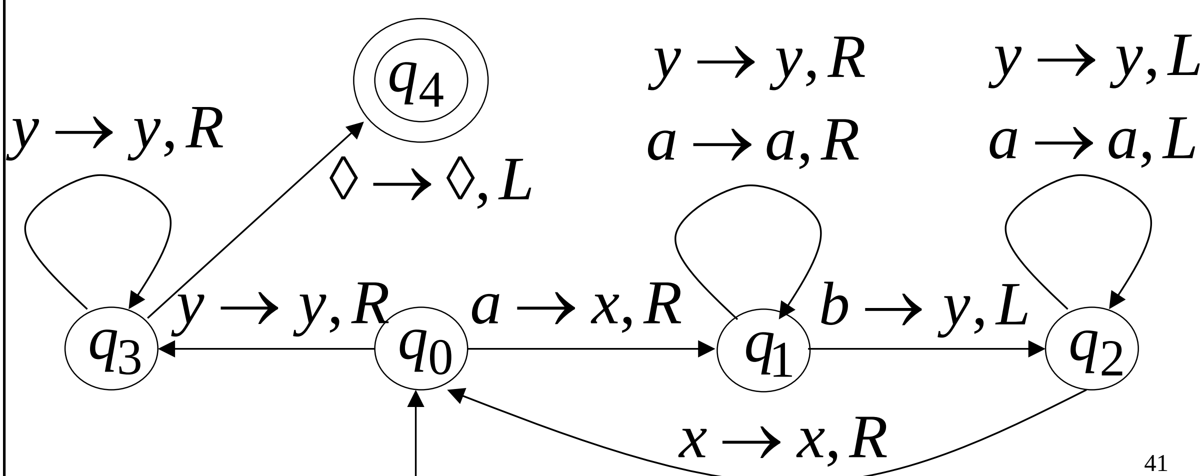
find leftmost b and replace it with y

Until there are no more a's or b's

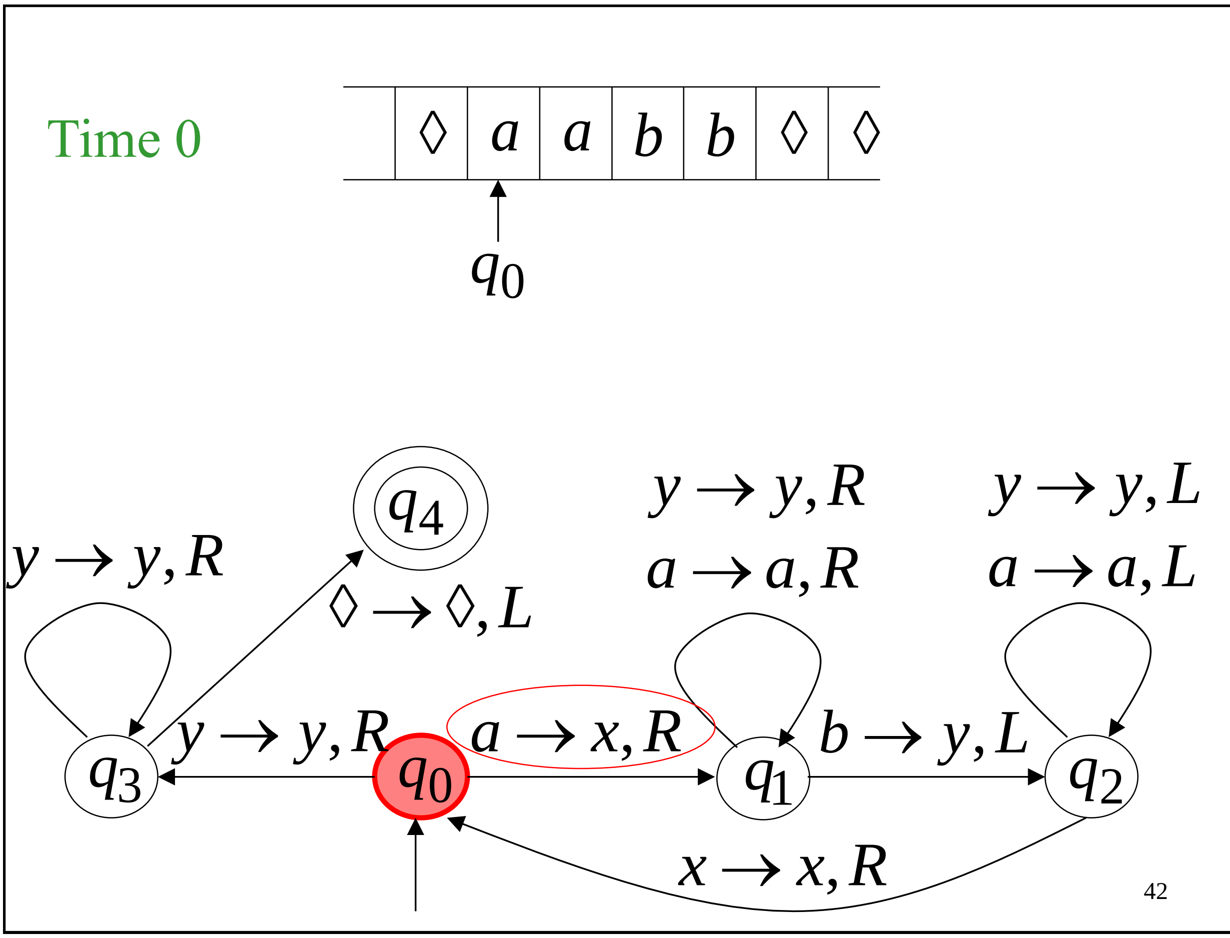
If there is a remaining a or b reject

Another Turing Machine Example

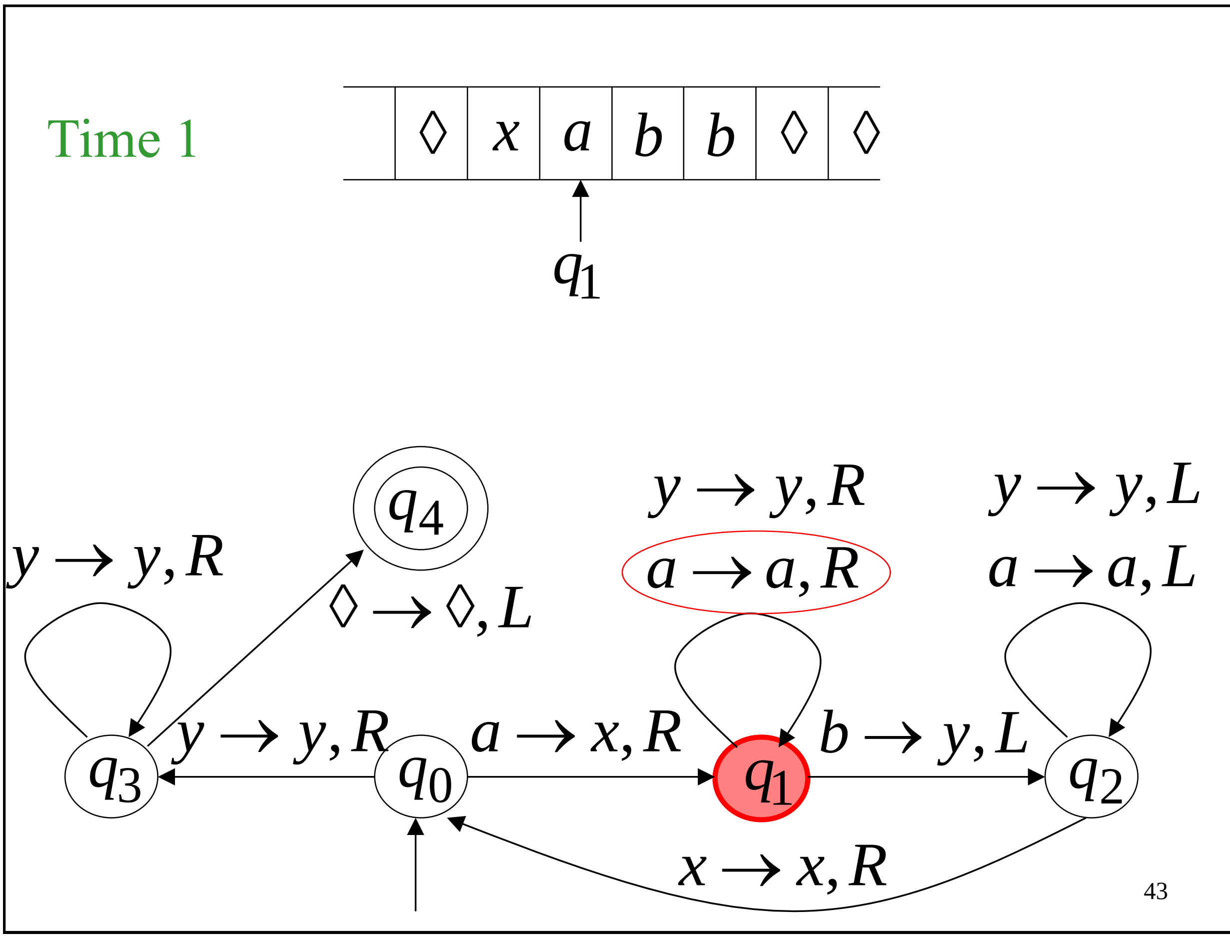
Turing machine for the language $\{a^n b^n \mid n \geq 1\}$



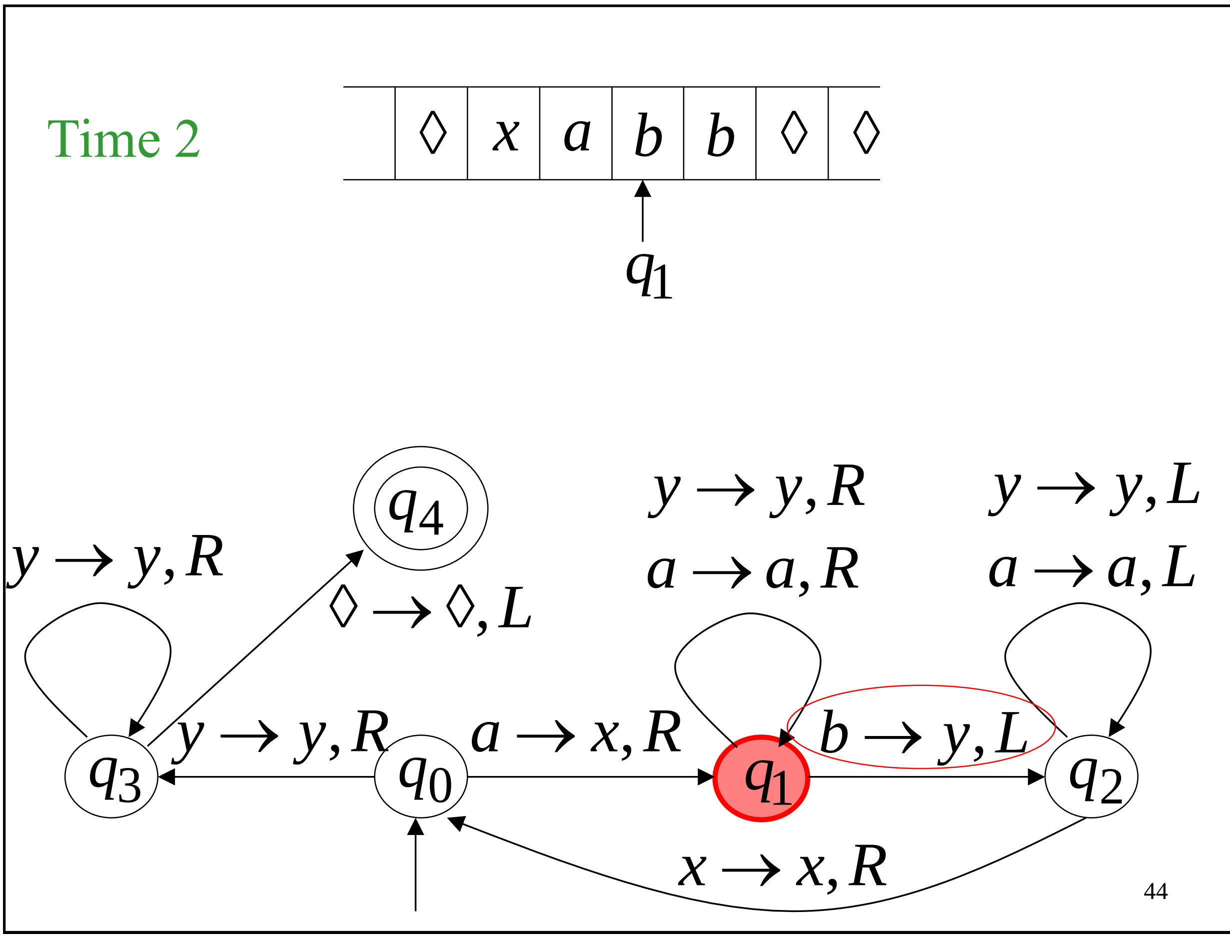
41



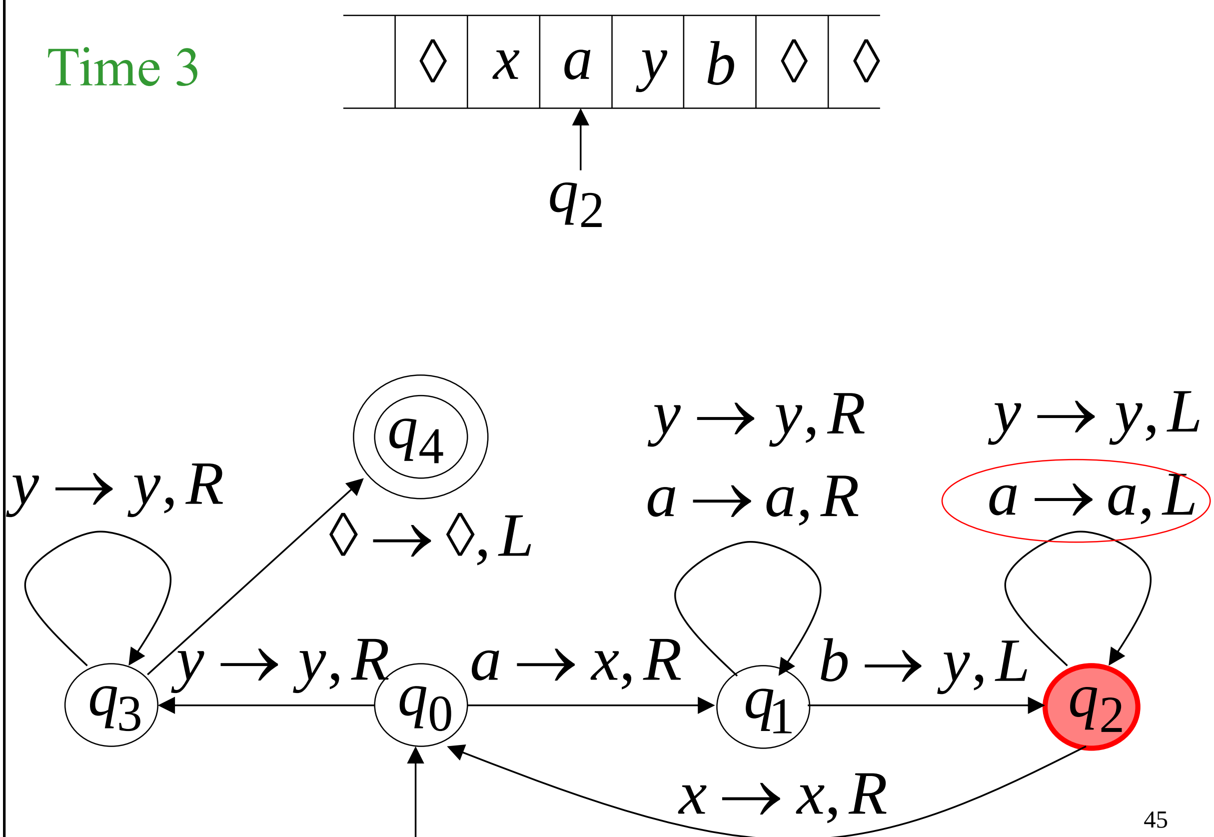
Time 0
◊
a
a
b
b
◊
◊
q_0
$y \to y, R$
q_4
$\lozenge \to \lozenge, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_0
q_1
q_2
$x \to x, R$
42



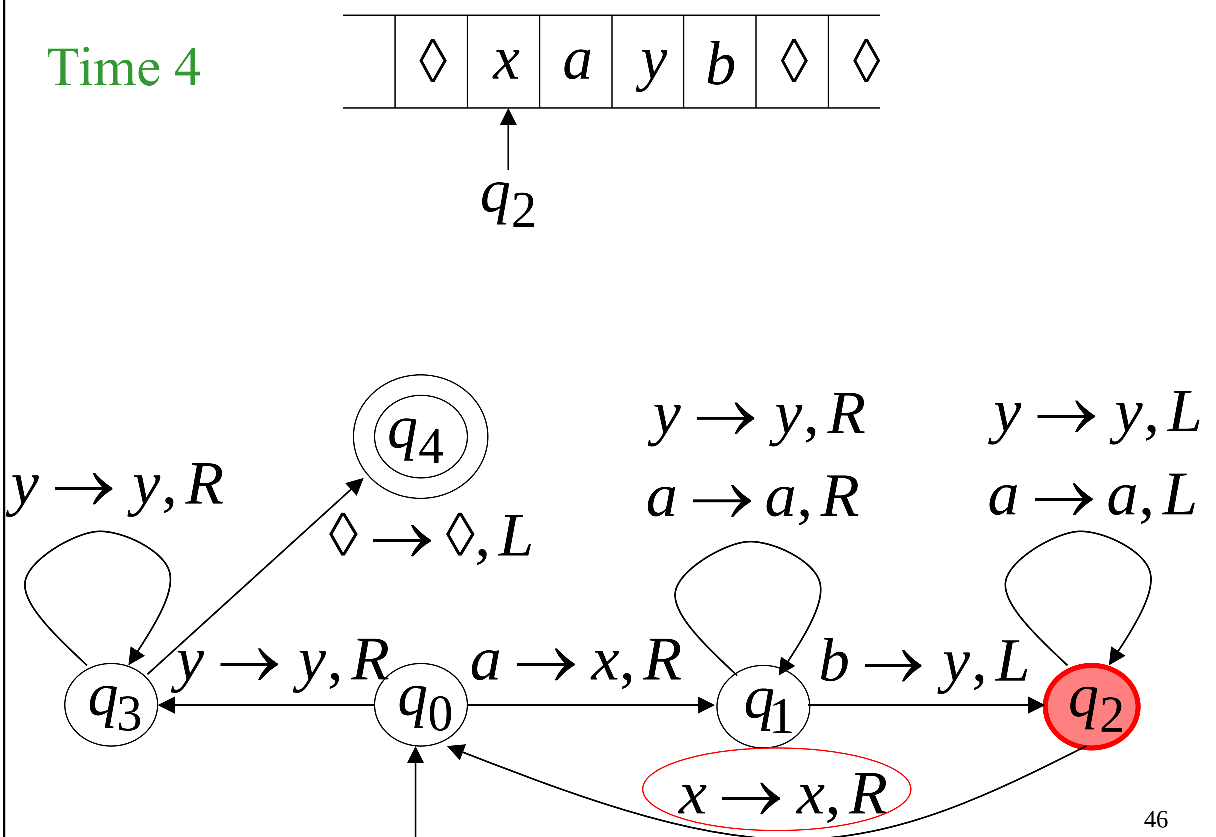
Time 1
◊
x
a
b
b
◊
◊
q_1
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$
43


Time 2
◊
x
a
b
b
◊
◊
q1
q4
y → y, R
◊ → ◊, L
y → y, R
a → x, R
y → y, R
a → a, R
y → y, L
a → a, L
b → y, L
x → x, R
q3
q0
q1
q2
44

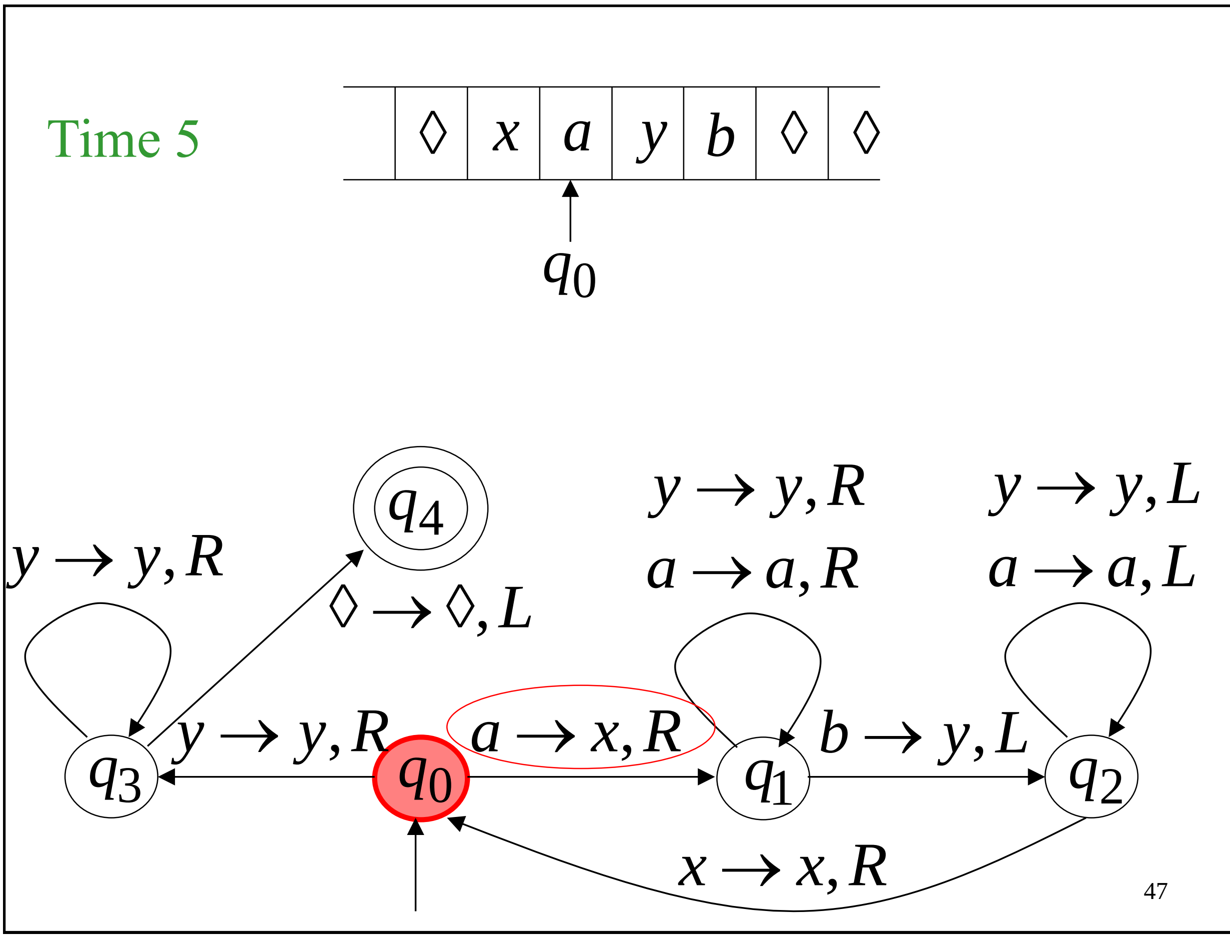
Time 3

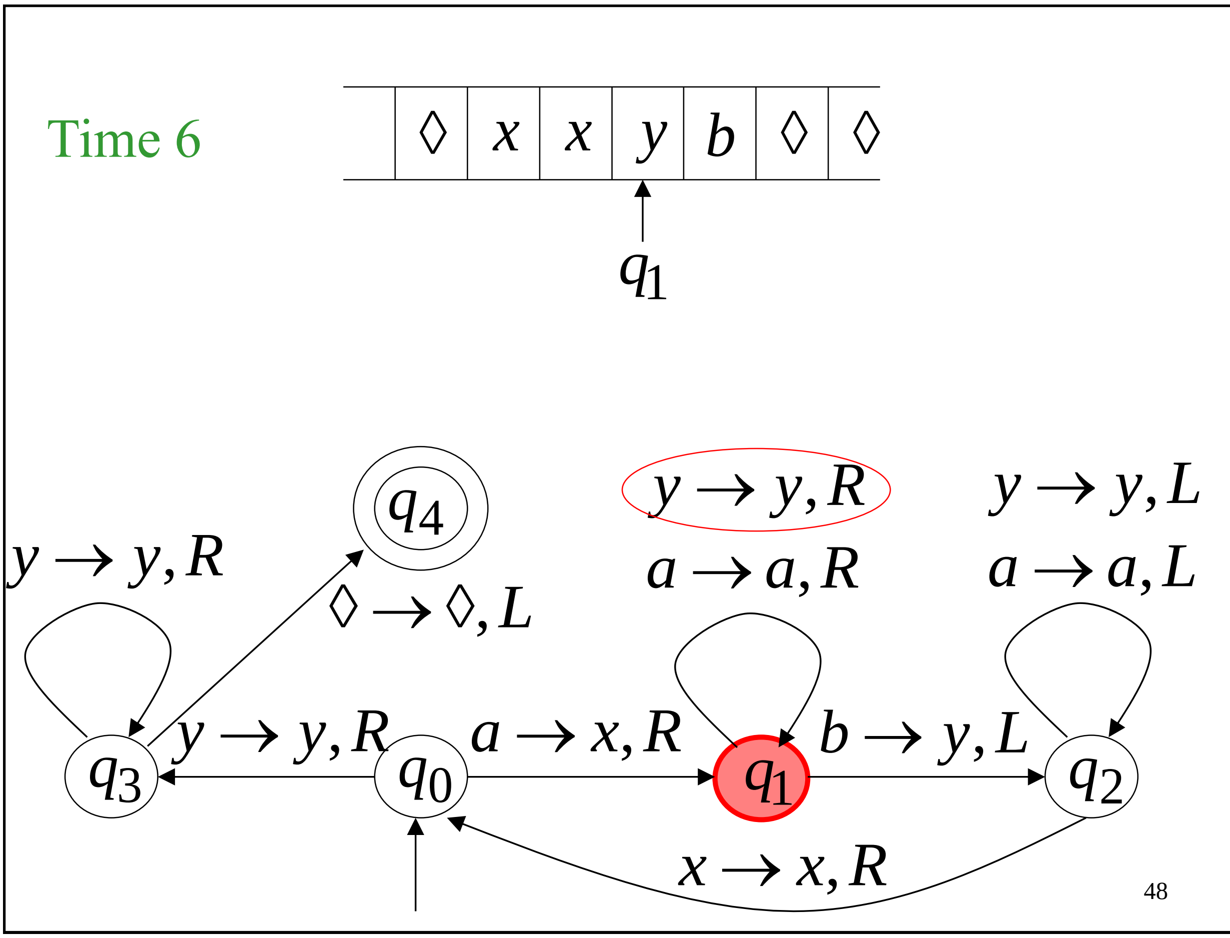

◊ x a y b ◊ ◊
q_2
q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
$y \to y, R$
$a \to x, R$
$b \to y, L$
q_3
q_0
q_1
q_2
$x \to x, R$

Time 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
$x \rightarrow x, R$

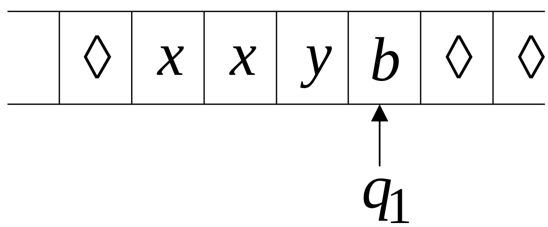
46

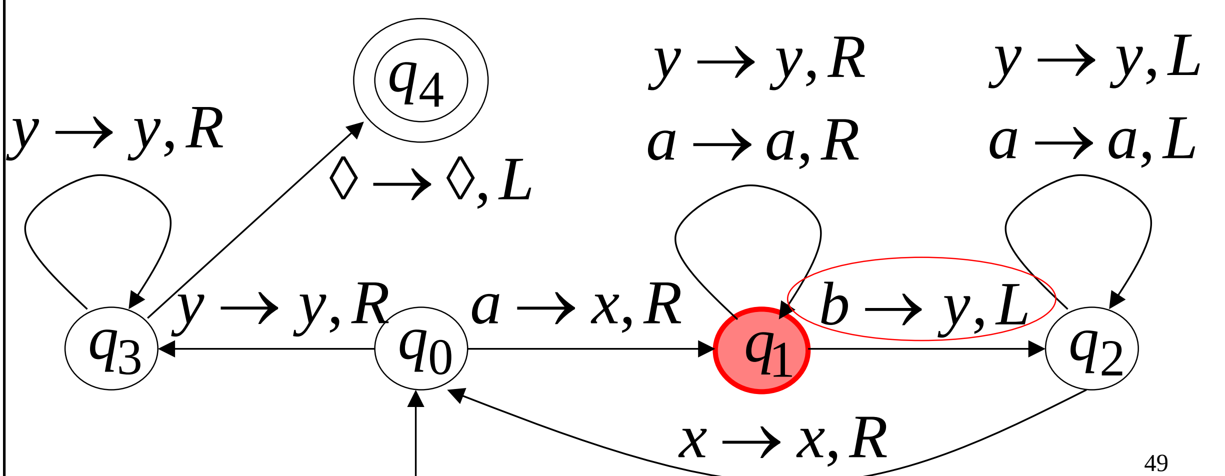

Time 5
◊ x a y b ◊ ◊
q_0
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$
47



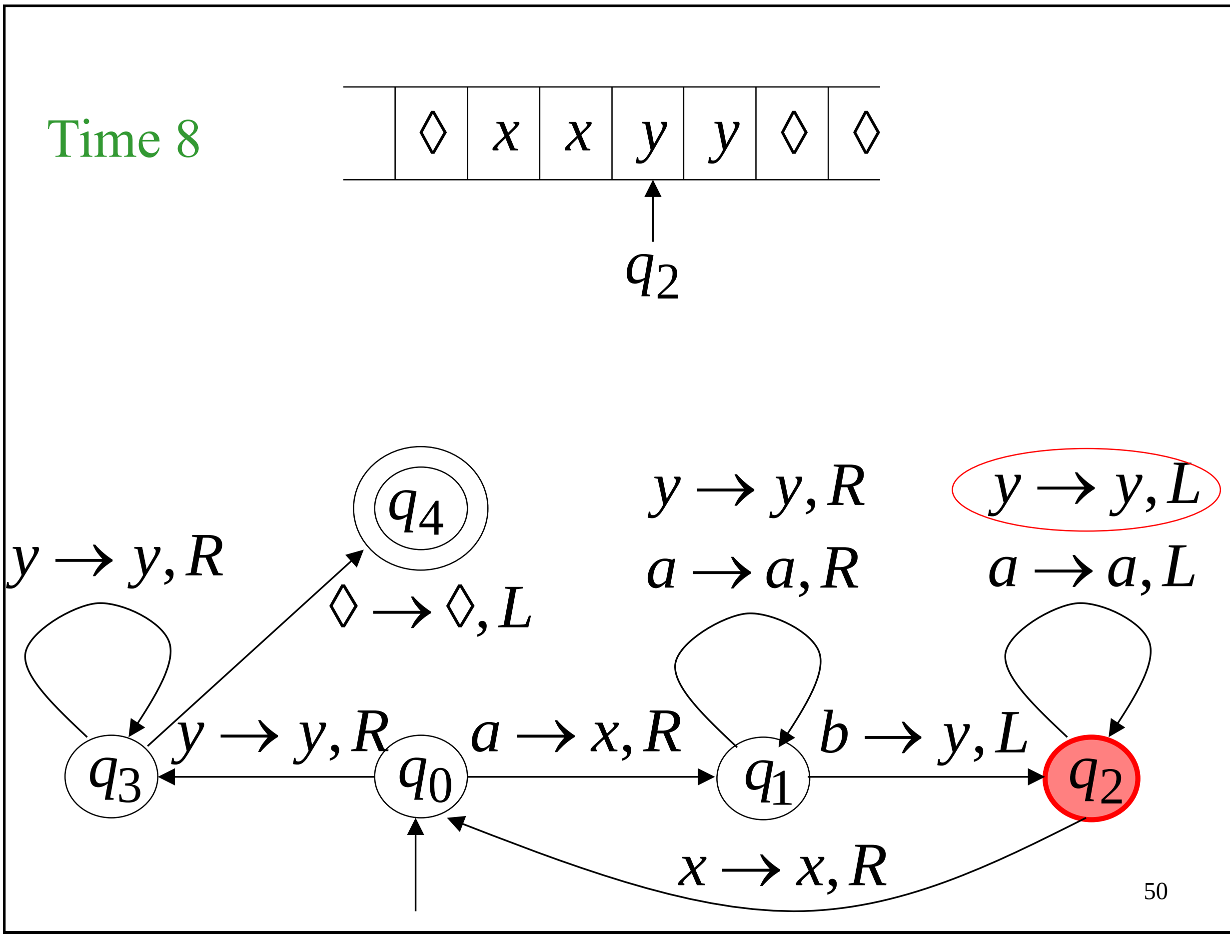
Time 6
◊
x
x
y
b
◊
◊
q_1
q_4
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$\lozenge \rightarrow \lozenge, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$
48

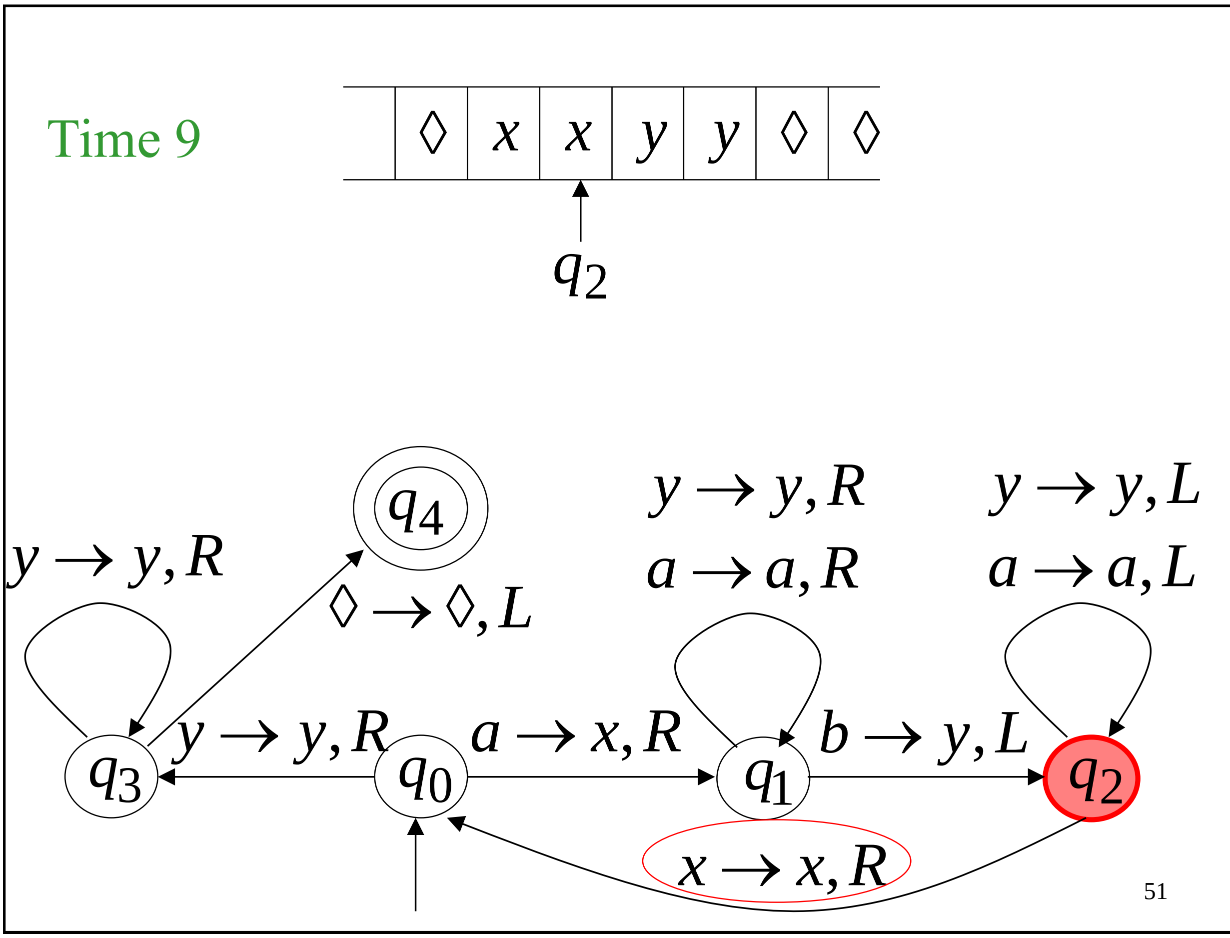
Time 7



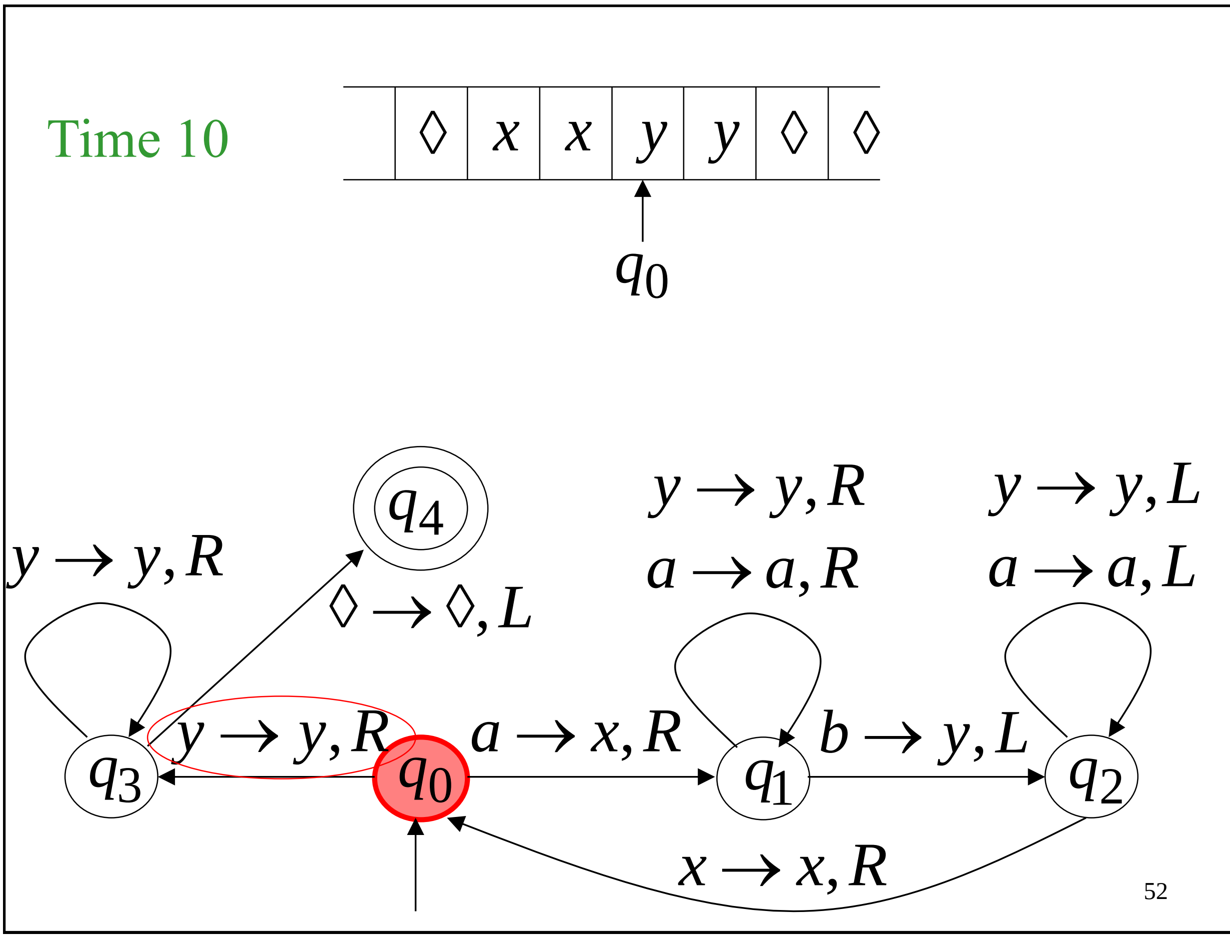


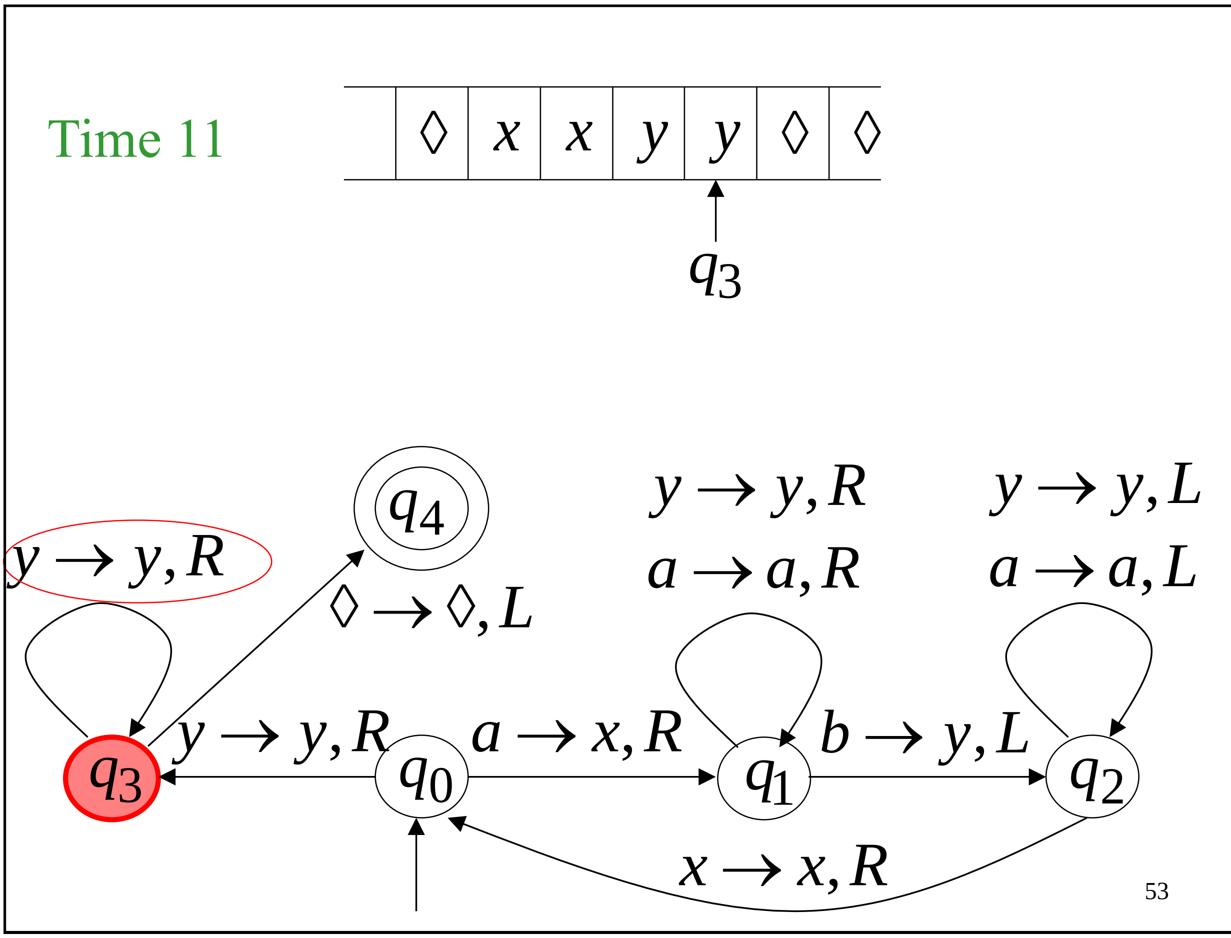
49

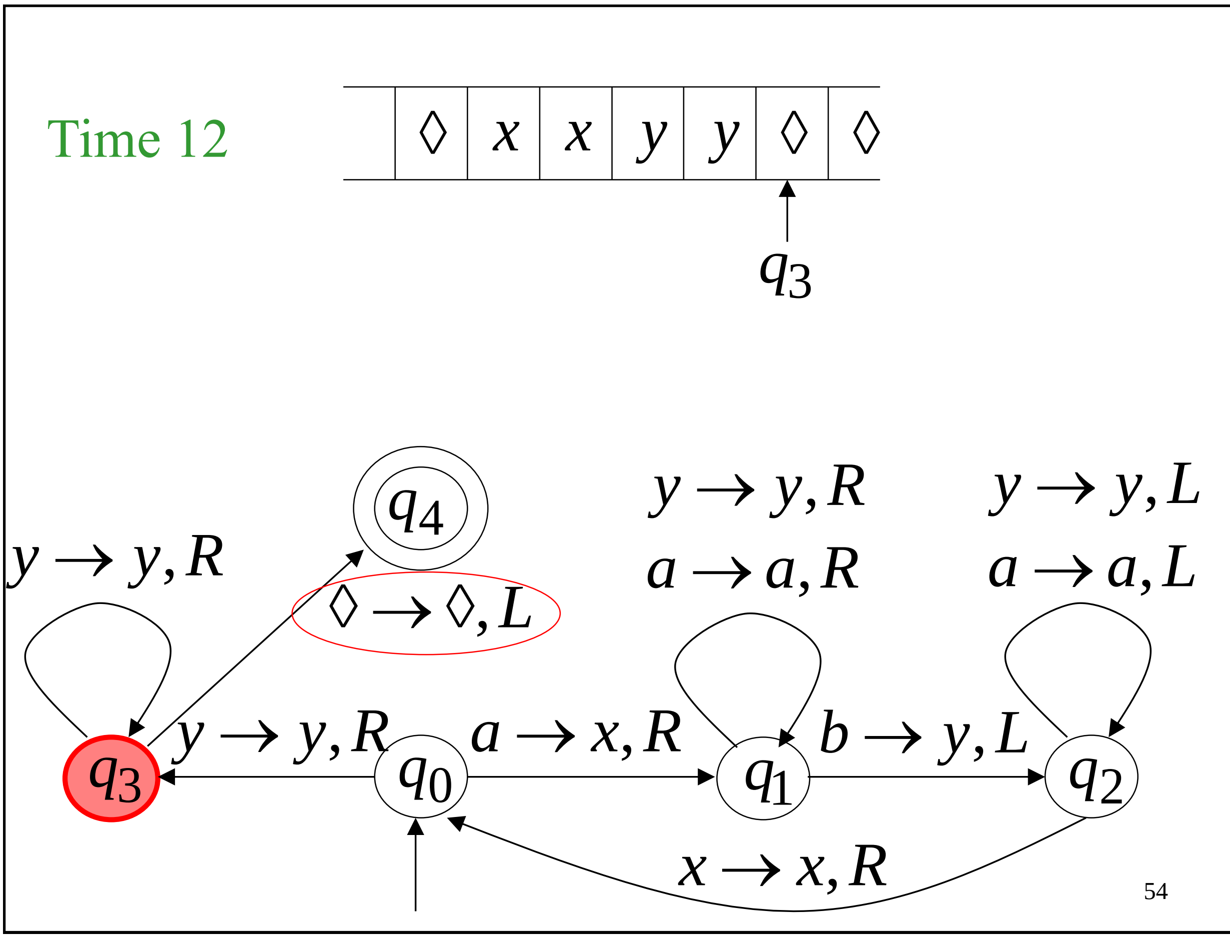

Time 8
◊
x
x
y
y
◊
◊
q_2
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$
50

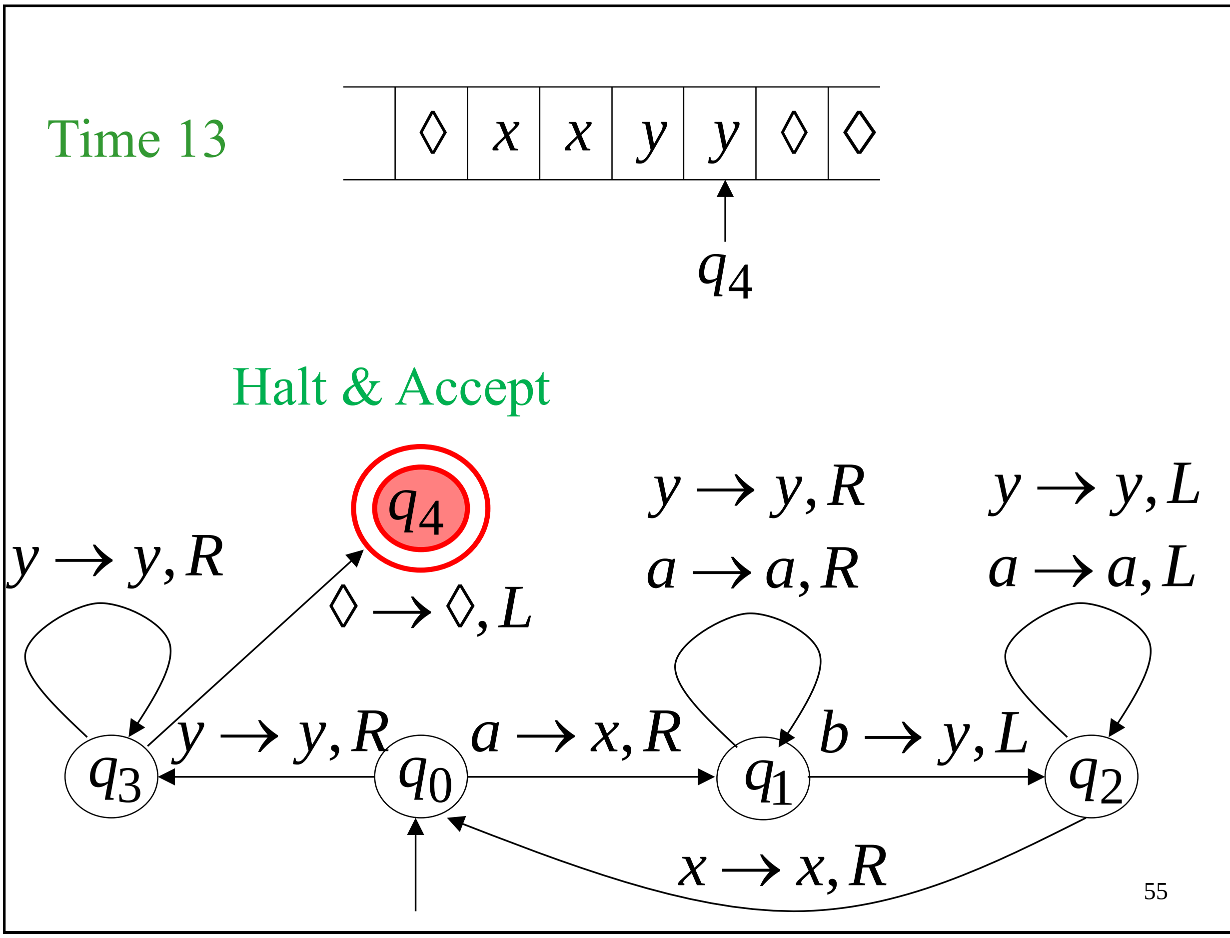


Time 9
◊ x x y y ◊ ◊
q_2
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$
51


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
$x \rightarrow x, R$
52


Time 11
◊
x
x
y
y
◊
◊
q_3
q_4
$y \rightarrow y, R$
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$
53


Time 12
◊ x x y y ◊ ◊
q3
q4
y → y, R
◊ → ◊, L
y → y, R
a → a, R
y → y, L
a → a, L
y → y, R
a → x, R
b → y, L
q3
q0
q1
q2
x → x, R
54


Time 13
◊ x x y y ◊ ◊
q_4
Halt & Accept
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
$x \rightarrow x, R$
55

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

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

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

Transition Function

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

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

Turing Machine:

States

Input alphabet

Tape alphabet

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

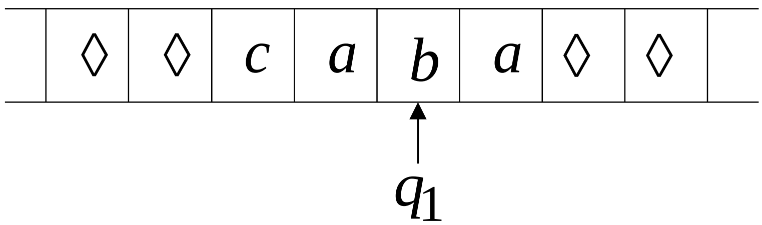
Transition function

Initial state

blank

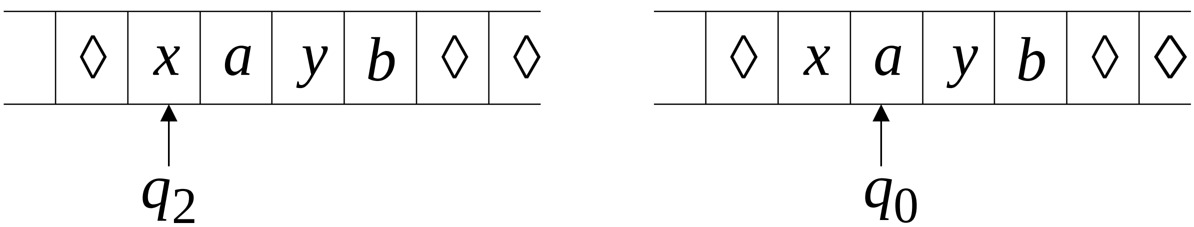
Accept states

Configuration



Instantaneous description: $ca\ q_1\ ba$

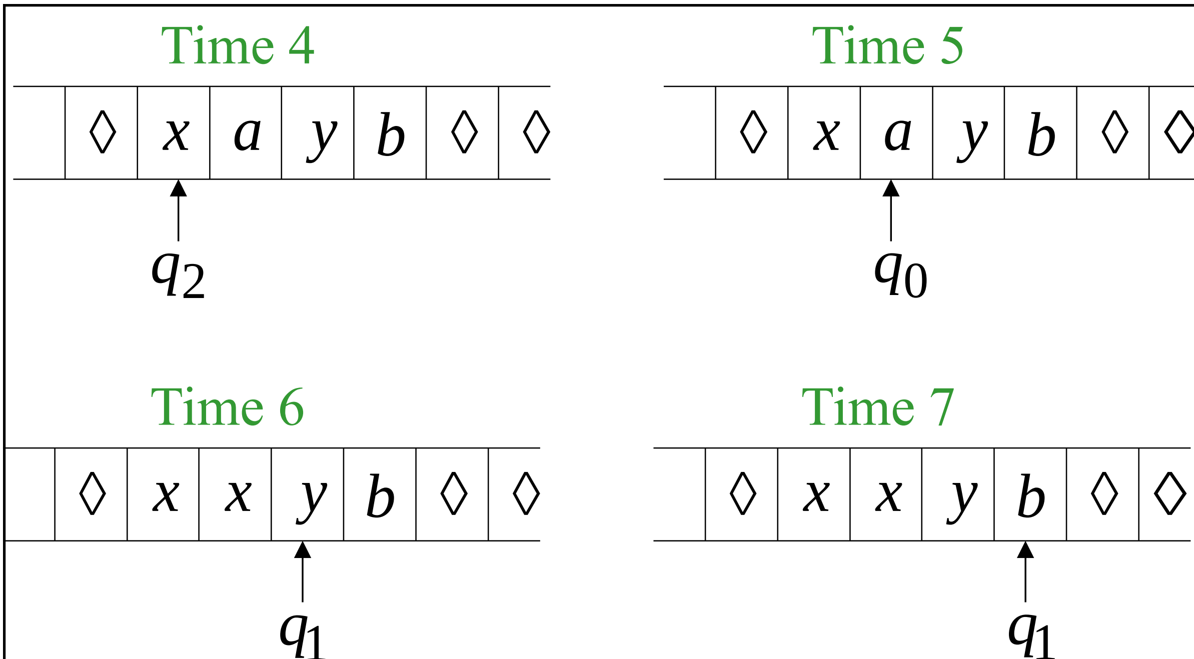
Time 4 Time 5



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

(yields in one mode)

62



A computation

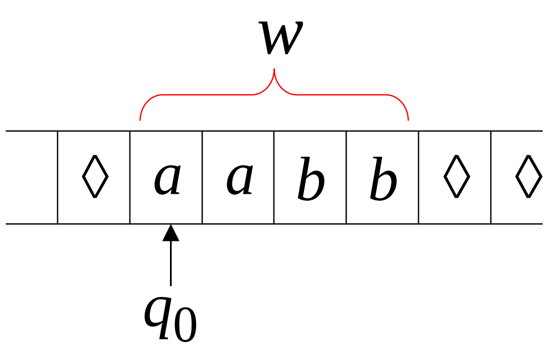
$$q_2\ xayb \succ x\ q_0\ ayb \succ xx\ q_1\ yb \succ xxy\ q_1\ b$$

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

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

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

Initial state

Accept state

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

- Turing Recognizable

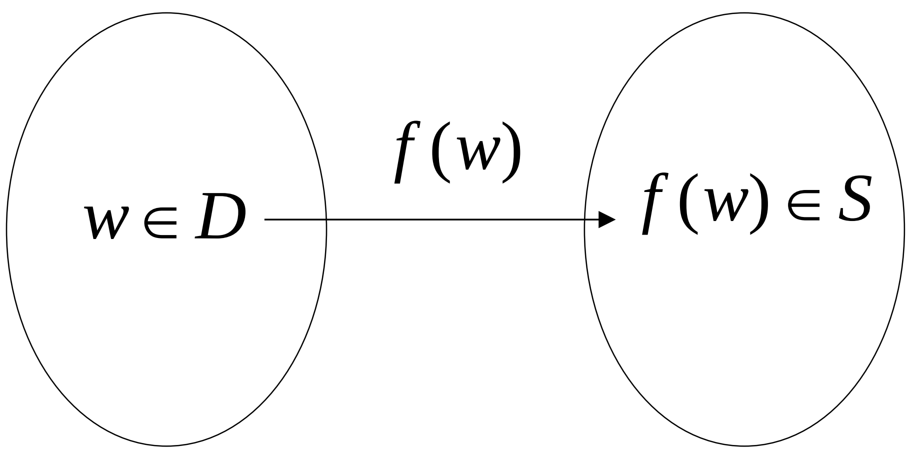
Other names used:

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

Integer Domain

Decimal: 5

Binary: 101

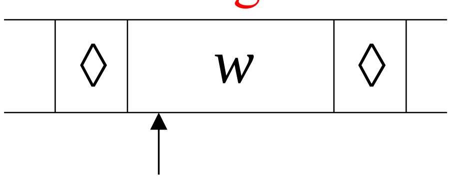
Unary: 11111

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:

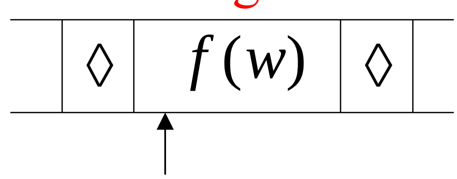
Initial configuration



q_0

initial state

Final configuration



q_f

accept state

For all $w \in D$ Domain

71

In other words:

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

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

Initial Configuration

Final Configuration

For all $w \in D$ Domain

Example

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

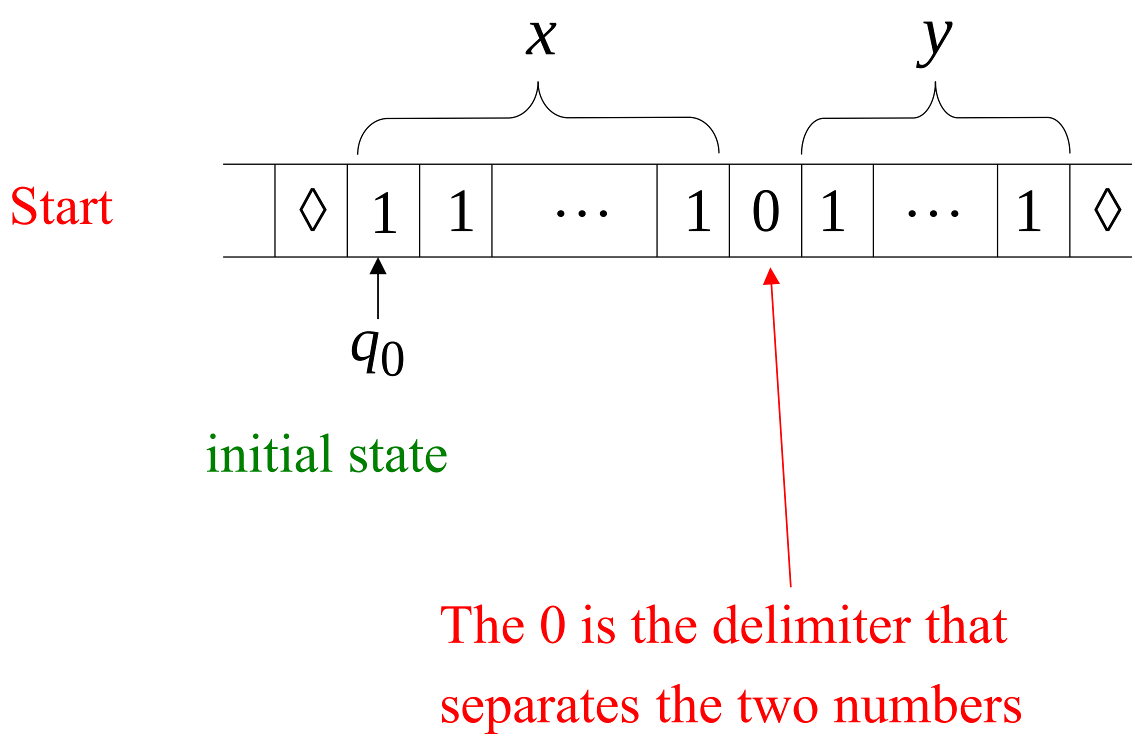
x, y are integers

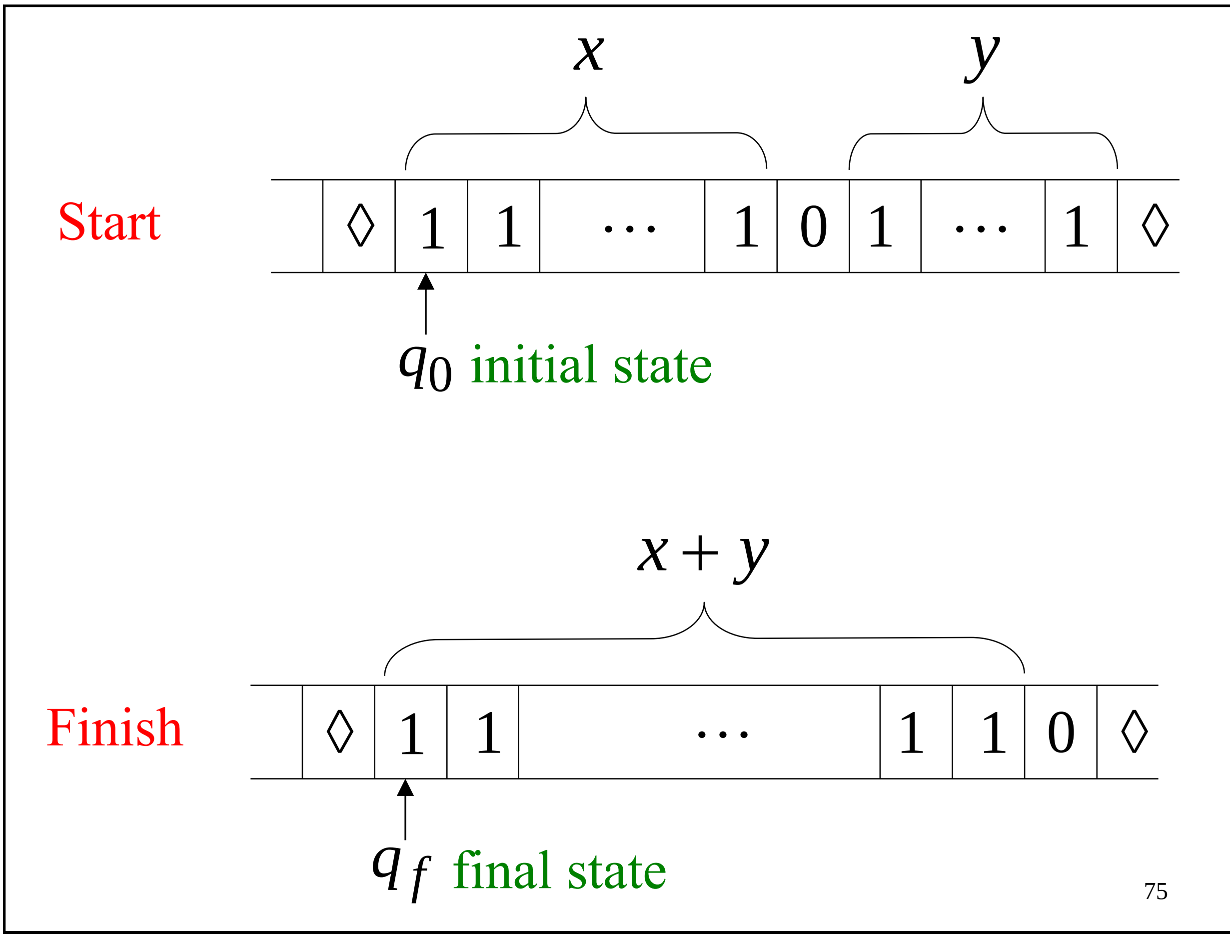
Turing Machine:

Input string: $x0y$ unary

Output string: $xy0$ unary

73

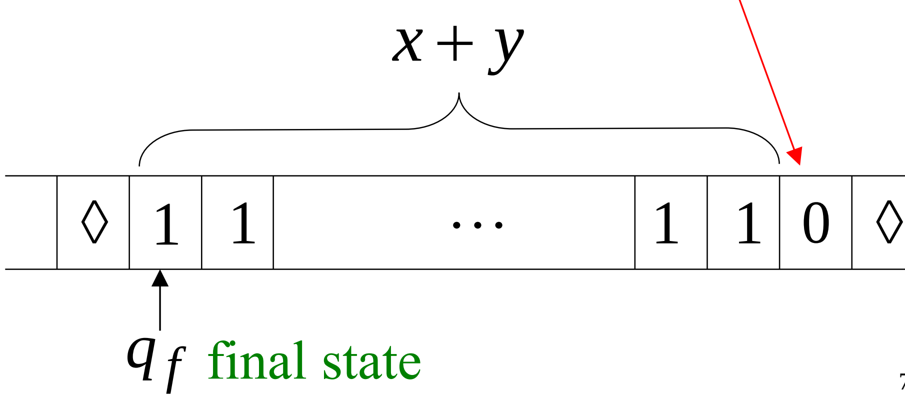

x
y
Start
◊ 1 1 ⋯ 1 0 1 ⋯ 1 ◊
q_0
initial state
The 0 is the delimiter that
separates the two numbers



x
y
Start
◊ 1 1 ⋯ 1 0 1 ⋯ 1 ◊
q_0 initial state
$x + y$
Finish
◊ 1 1 ⋯ 1 1 0 ◊
q_f final state
75

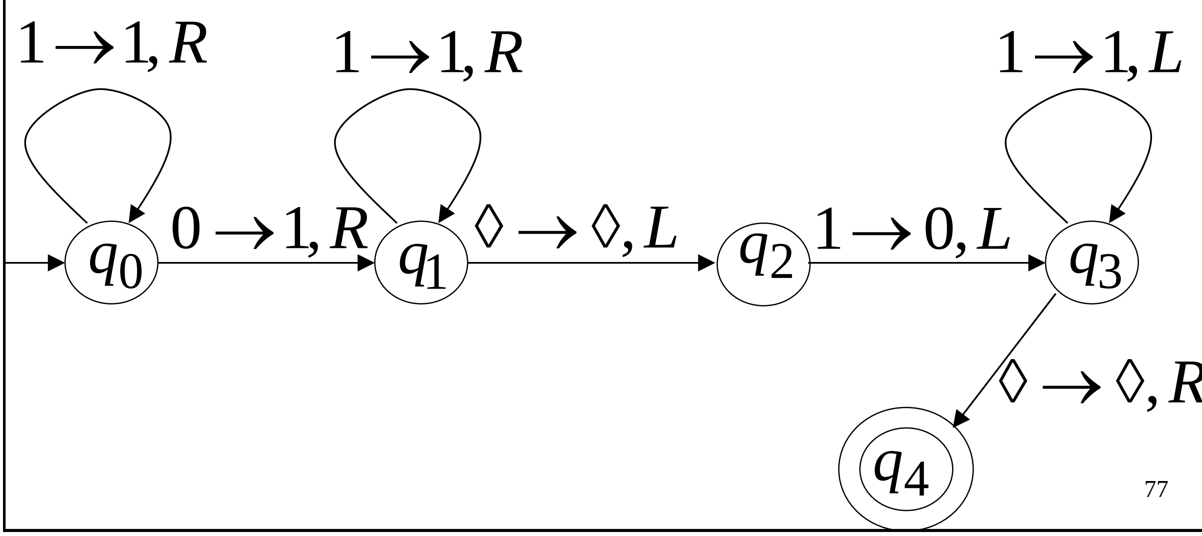
The 0 here helps when we use the result for other operations

Finish



76

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



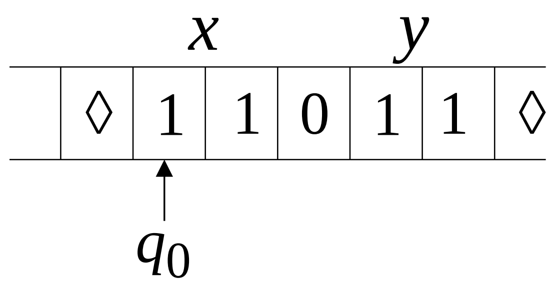
77

Execution Example:

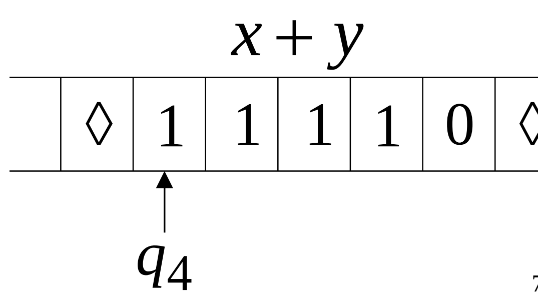
Time 0

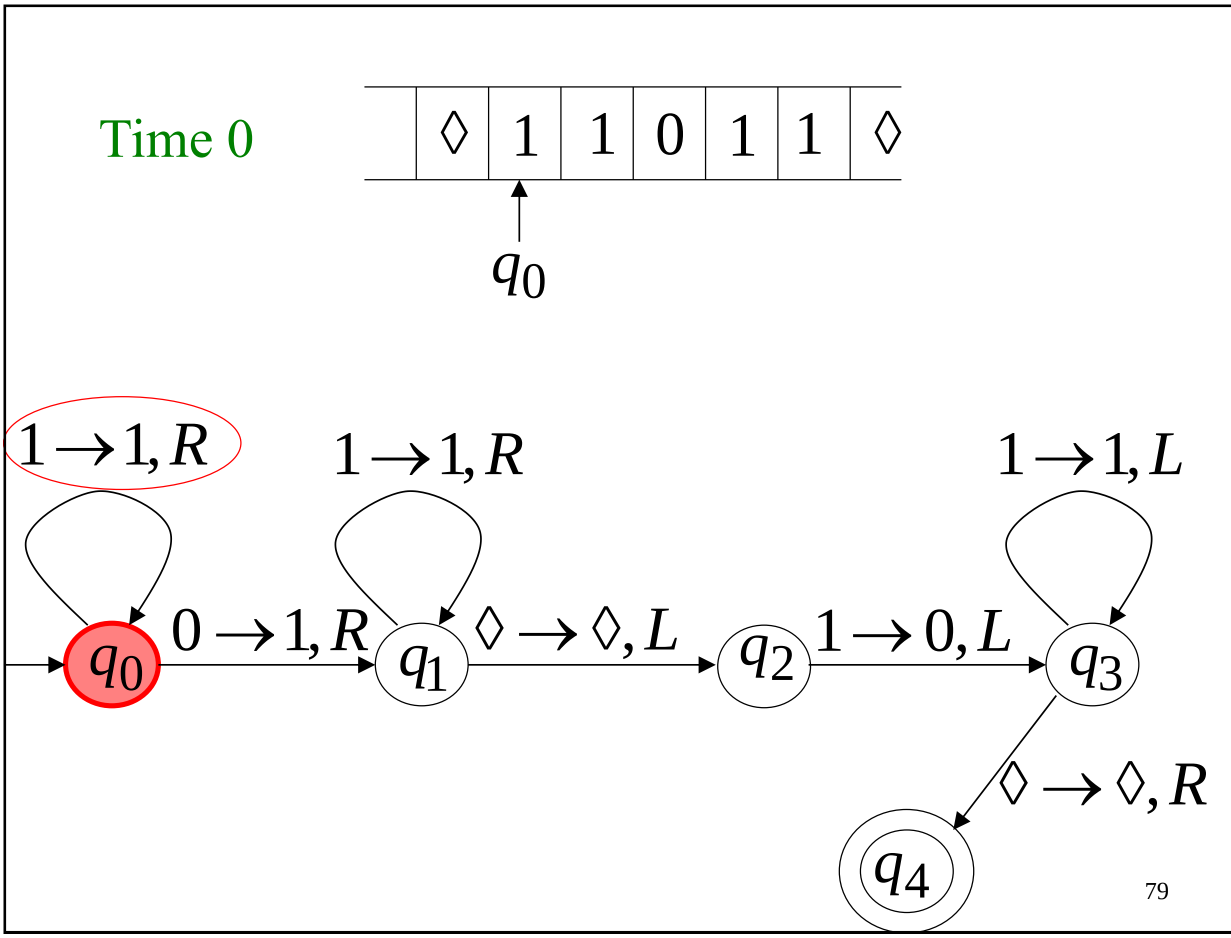
$x = 11$

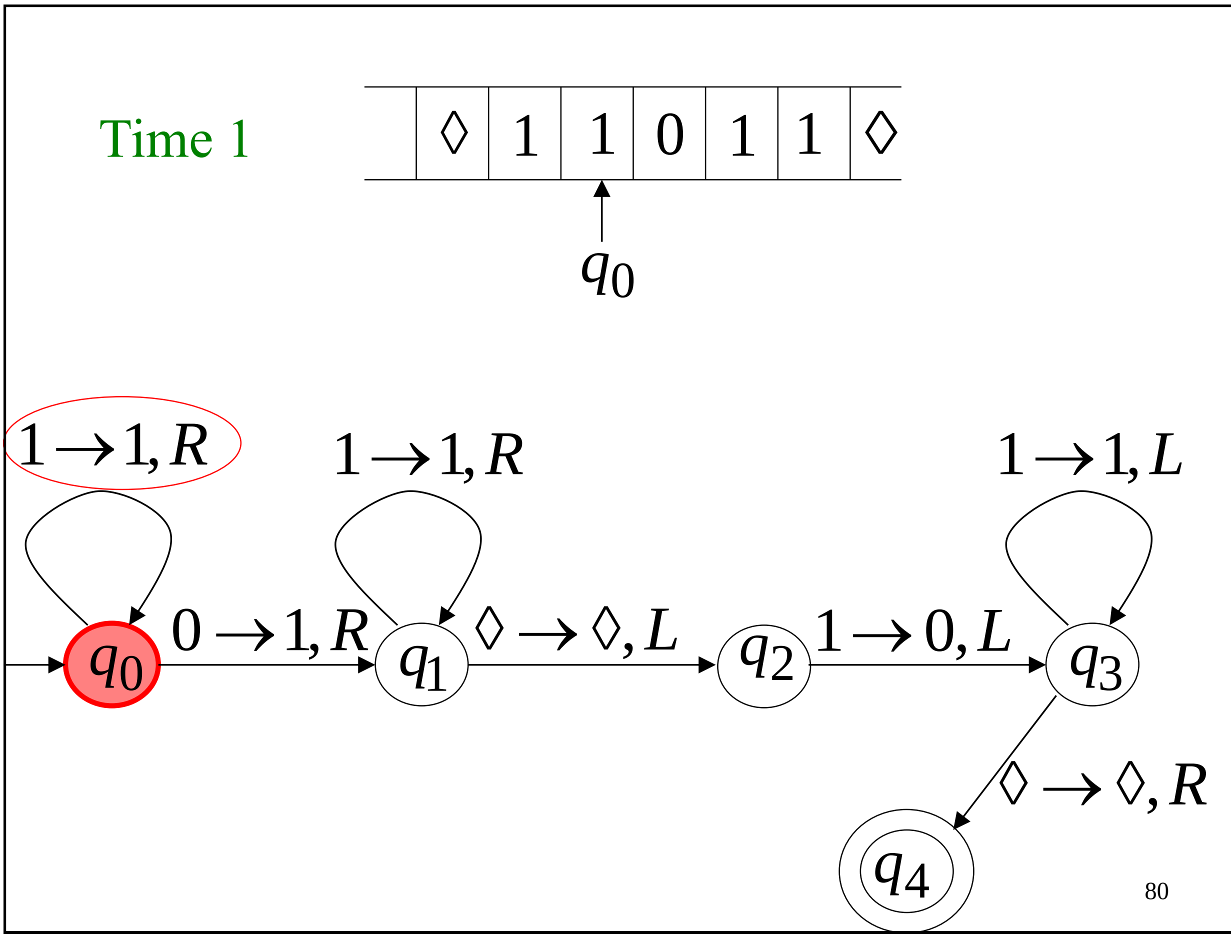
$y = 11$

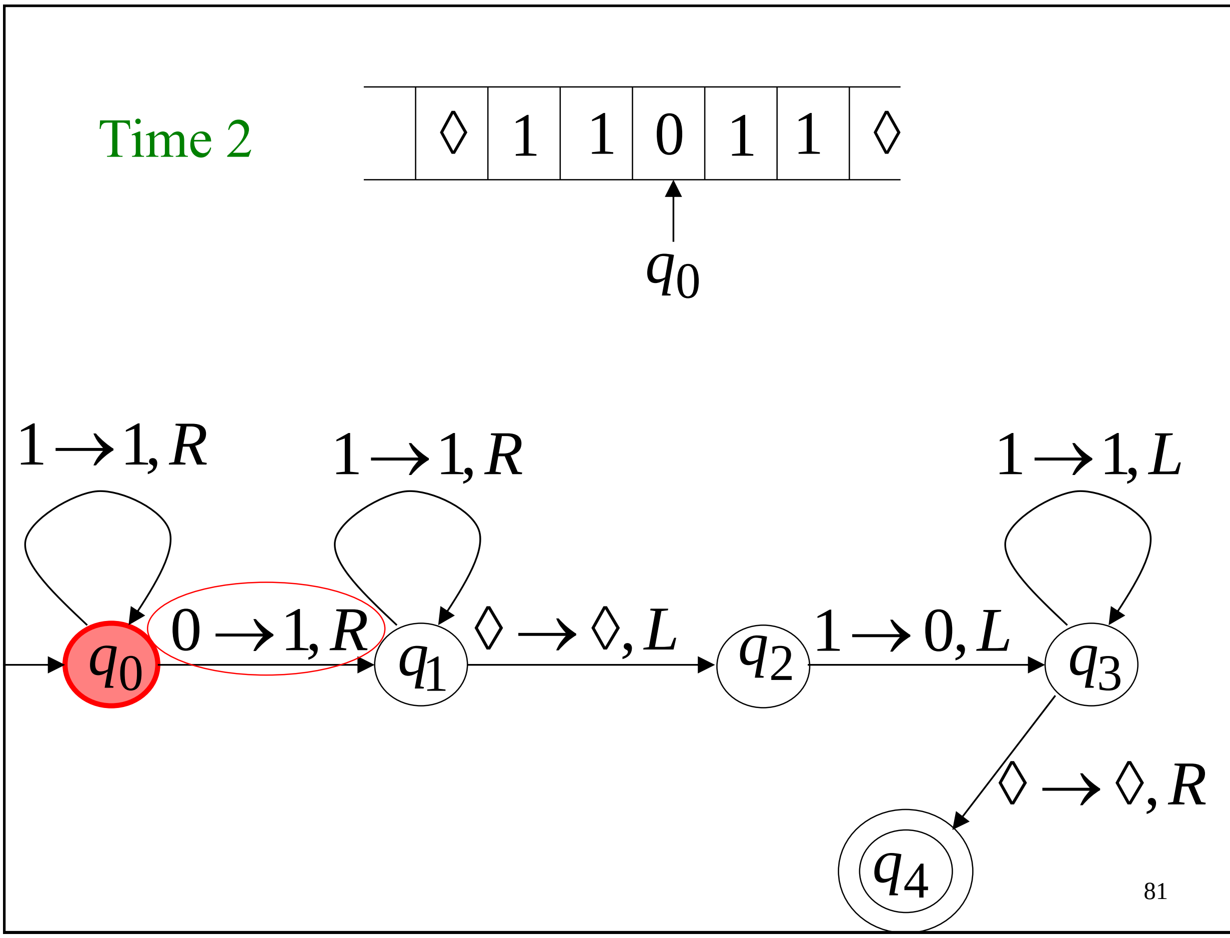


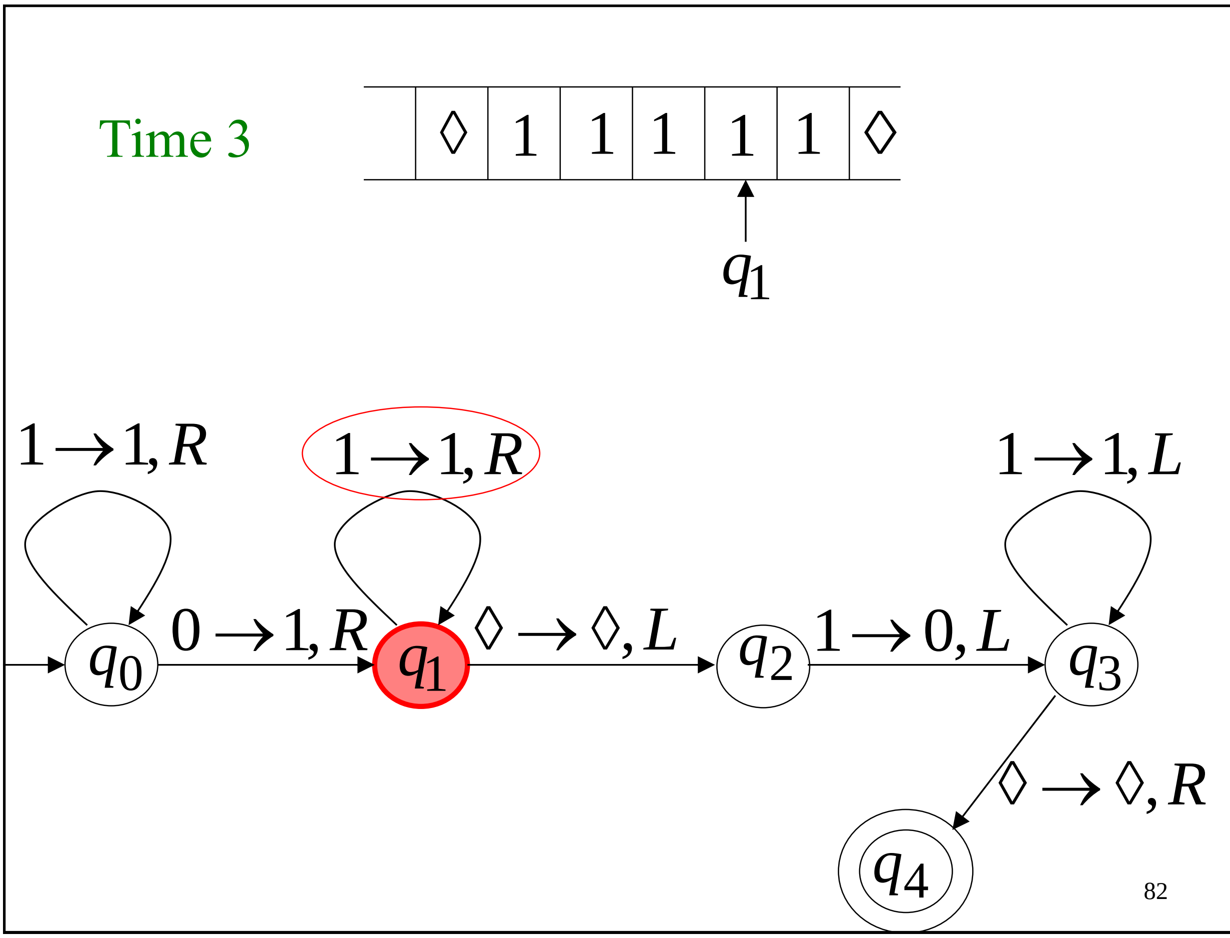
Final Result



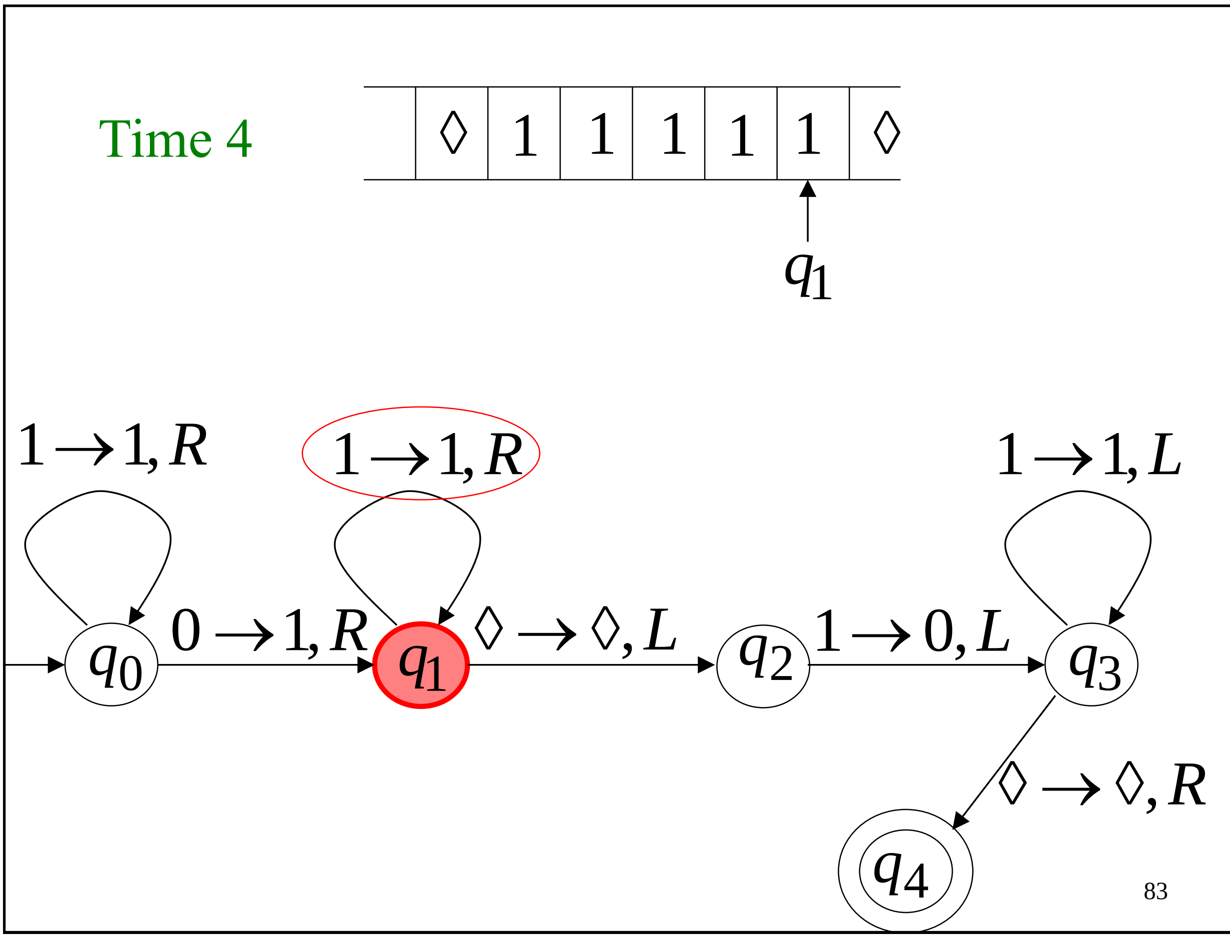

Time 0
◊
1
1
0
1
1
◊
q_0
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4
79


Time 1
◊ 1 1 0 1 1 ◊
q_0
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
$\lozenge \rightarrow \lozenge, R$
q_0
q_1
q_2
q_3
q_4
80

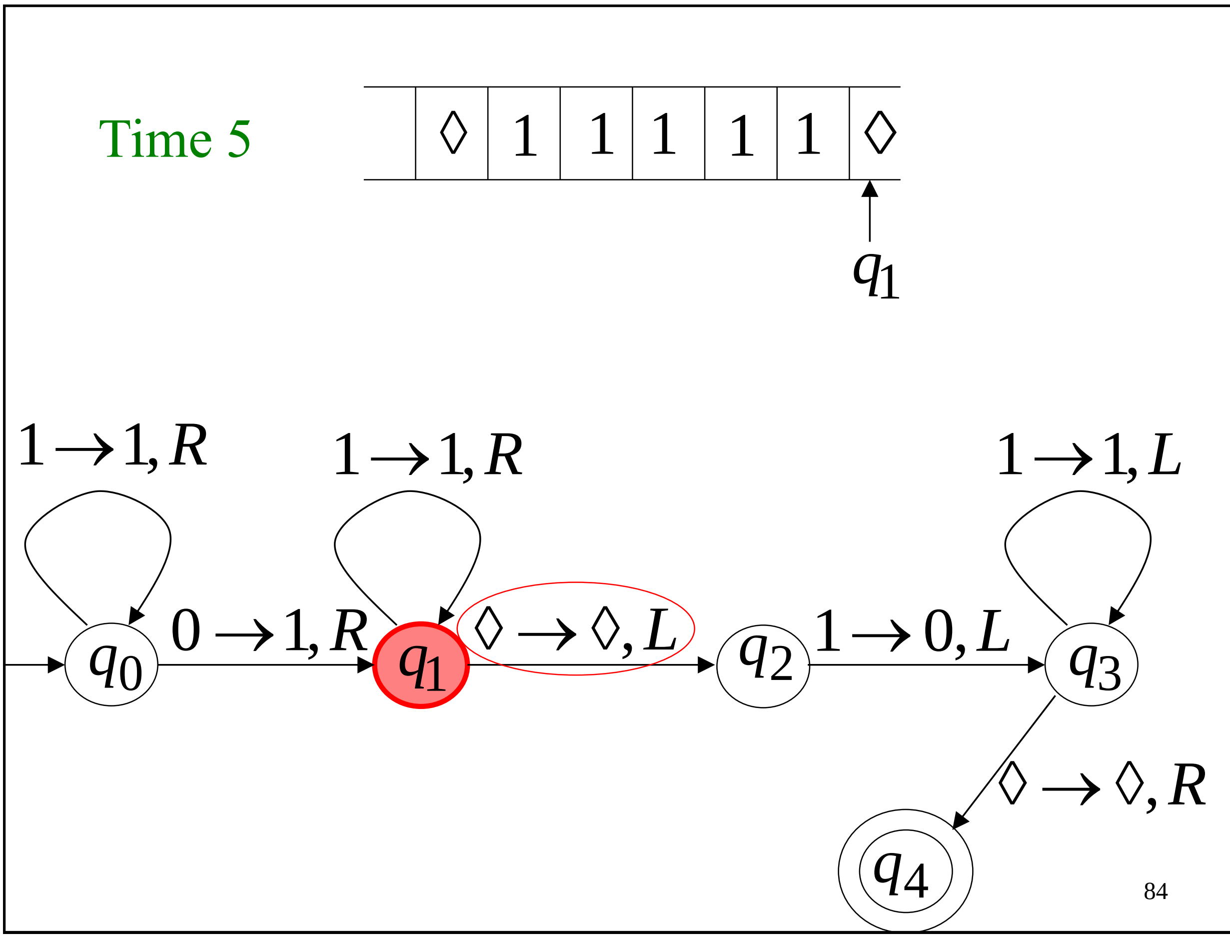

Time 2
◊ 1 1 0 1 1 ◊
q_0
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
$\lozenge \rightarrow \lozenge, R$
q_0
q_1
q_2
q_3
q_4
81

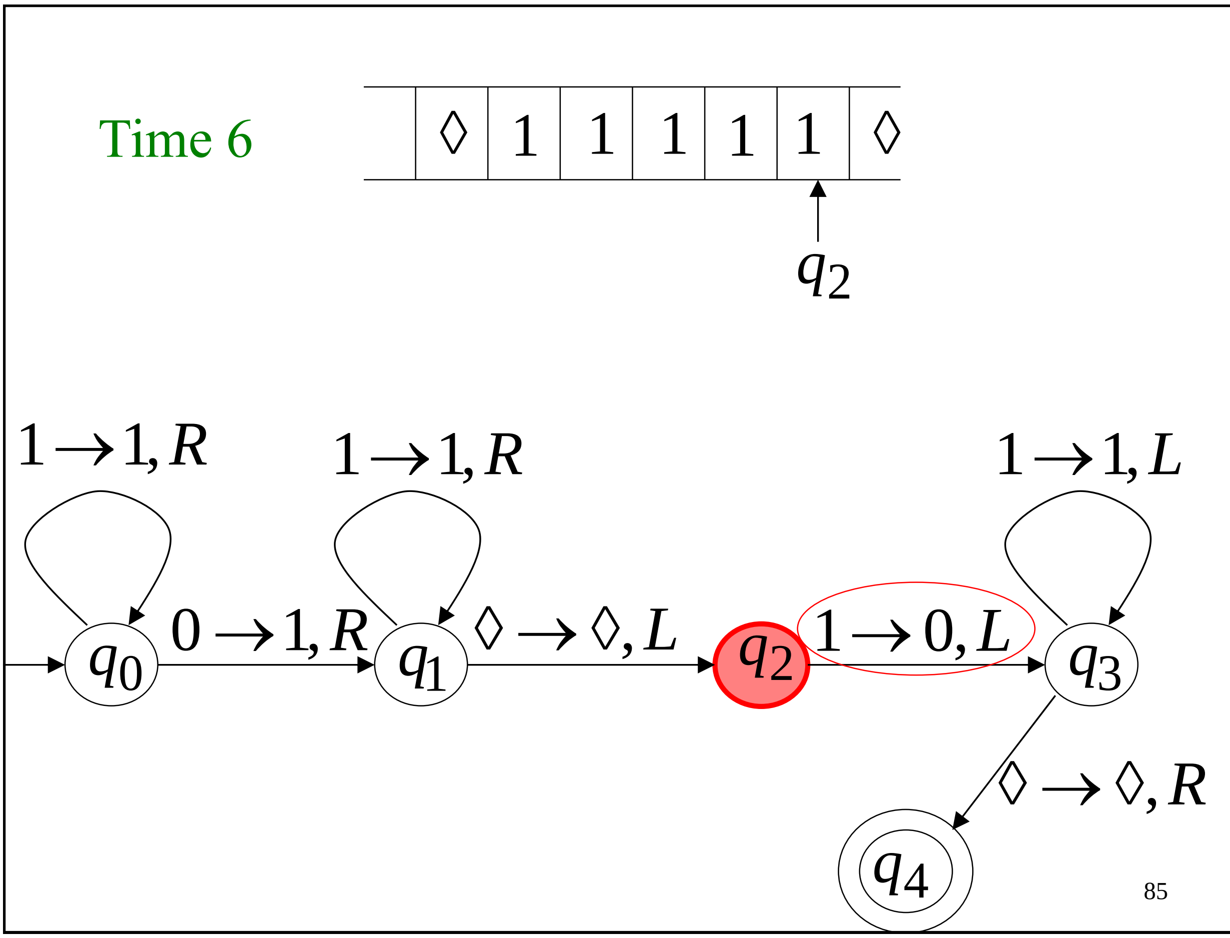


Time 3
◊ 1 1 1 1 1 ◊
q_1
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\Diamond \rightarrow \Diamond, L$
q_2
$1 \rightarrow 0, L$
q_3
$\Diamond \rightarrow \Diamond, R$
q_4
82

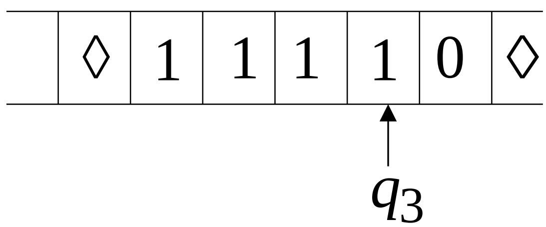


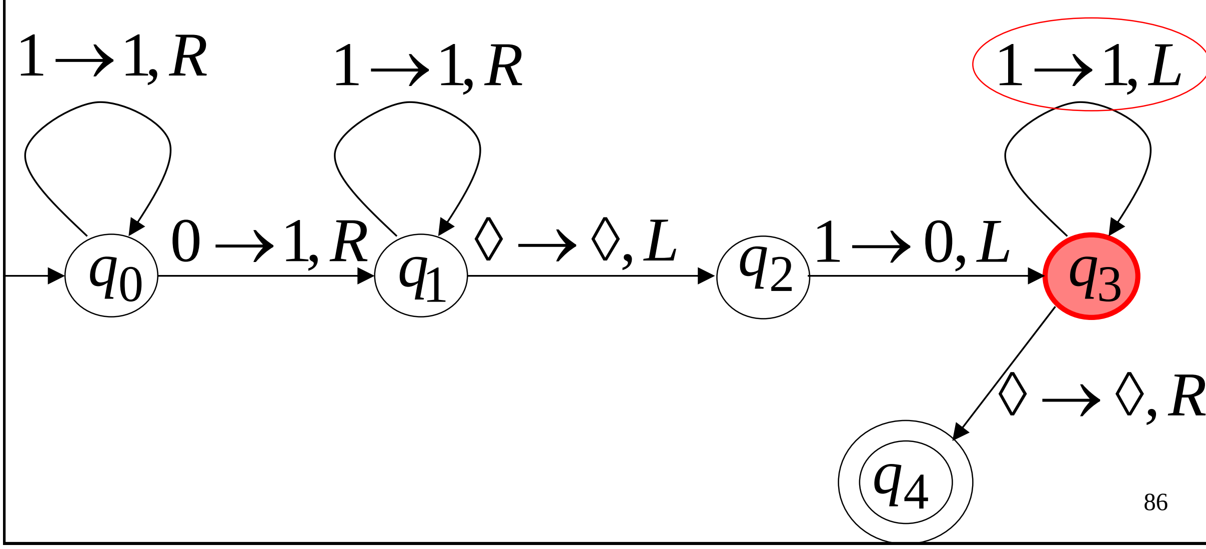
Time 4
◊ 1 1 1 1 1 ◊
q_1
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4
83


Time 5
◊ 1 1 1 1 1 ◊
q_1
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4
84

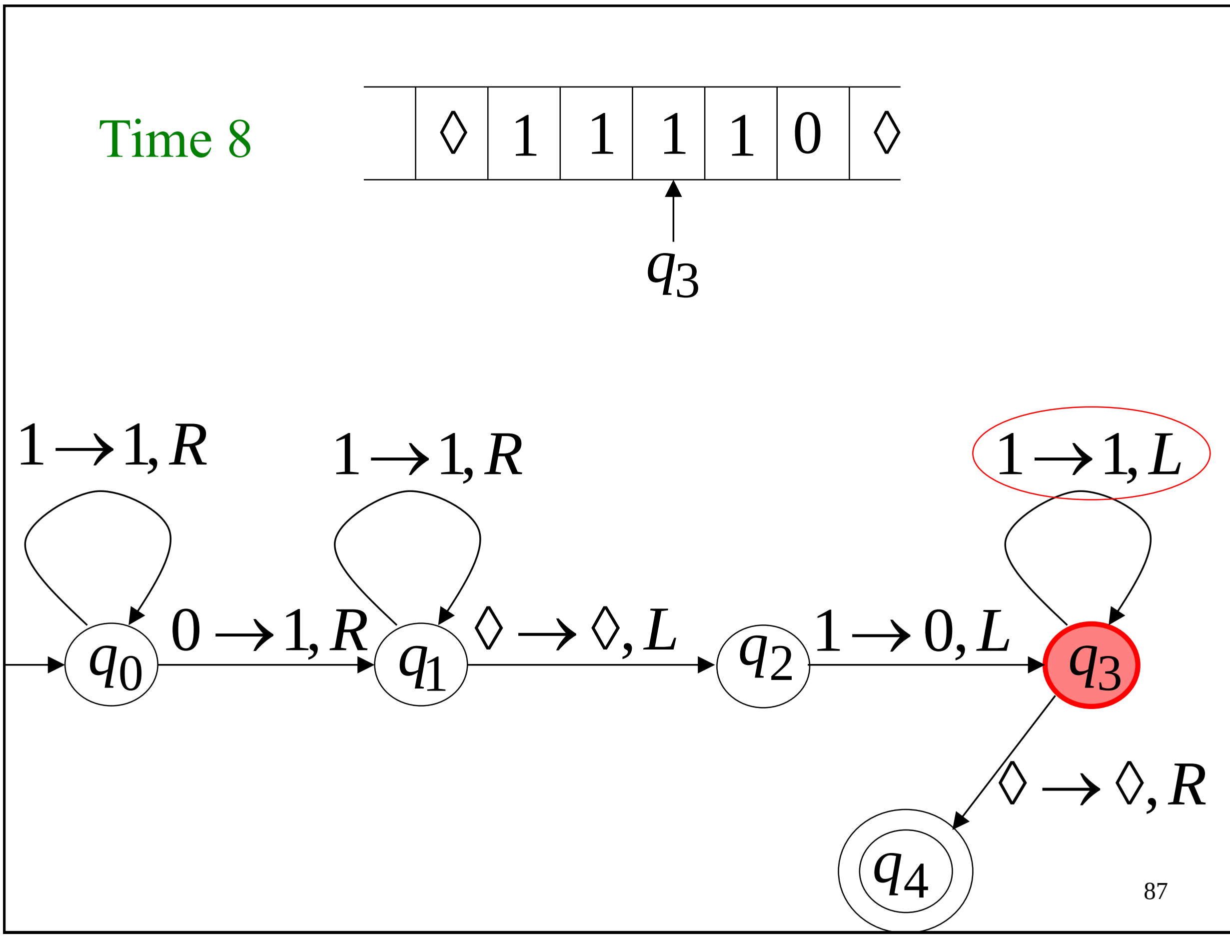

Time 6
◊
1
1
1
1
1
◊
q_2
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
q_0
q_1
q_2
q_3
$\lozenge \rightarrow \lozenge, R$
q_4
85

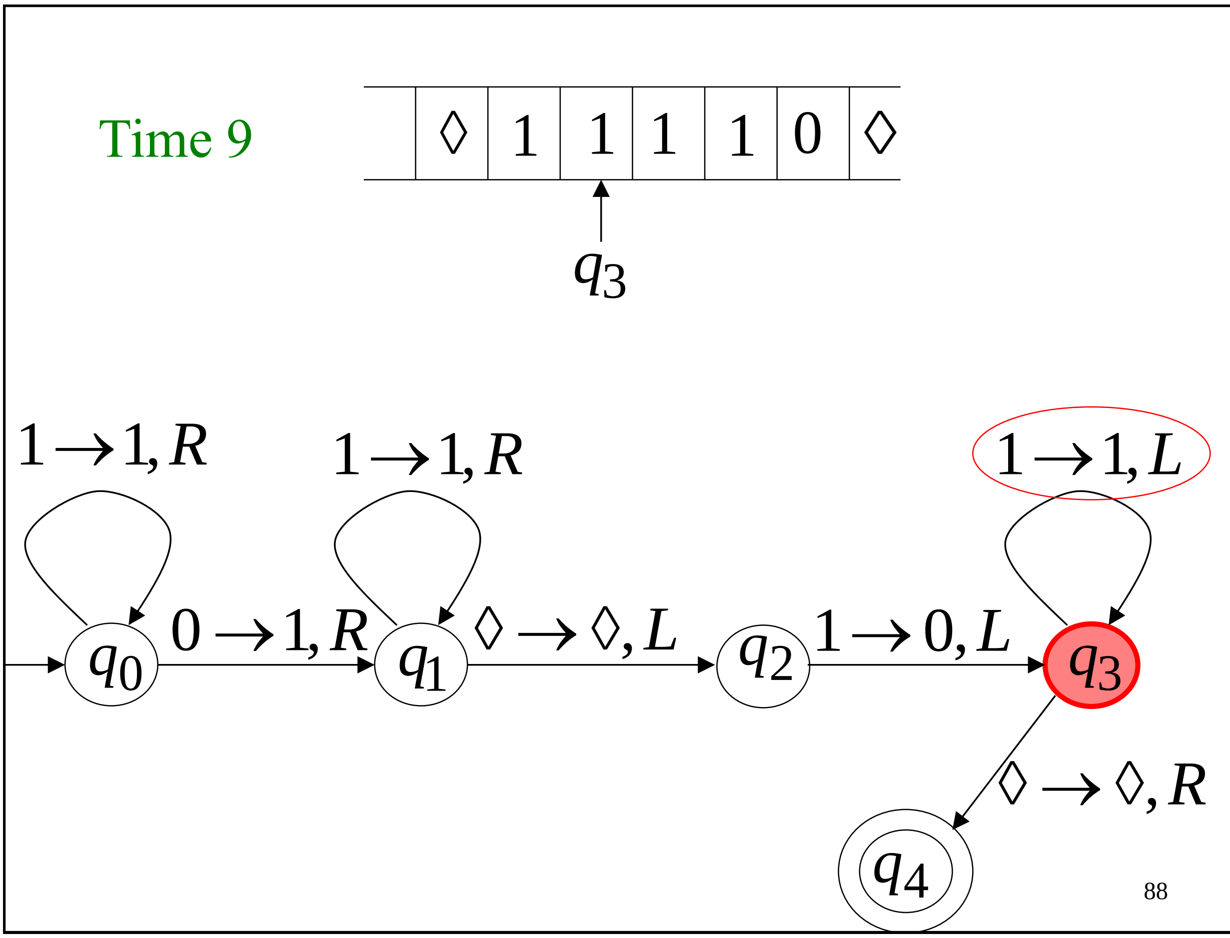
Time 7


◊
1
1
1
1
0
◊
q_3

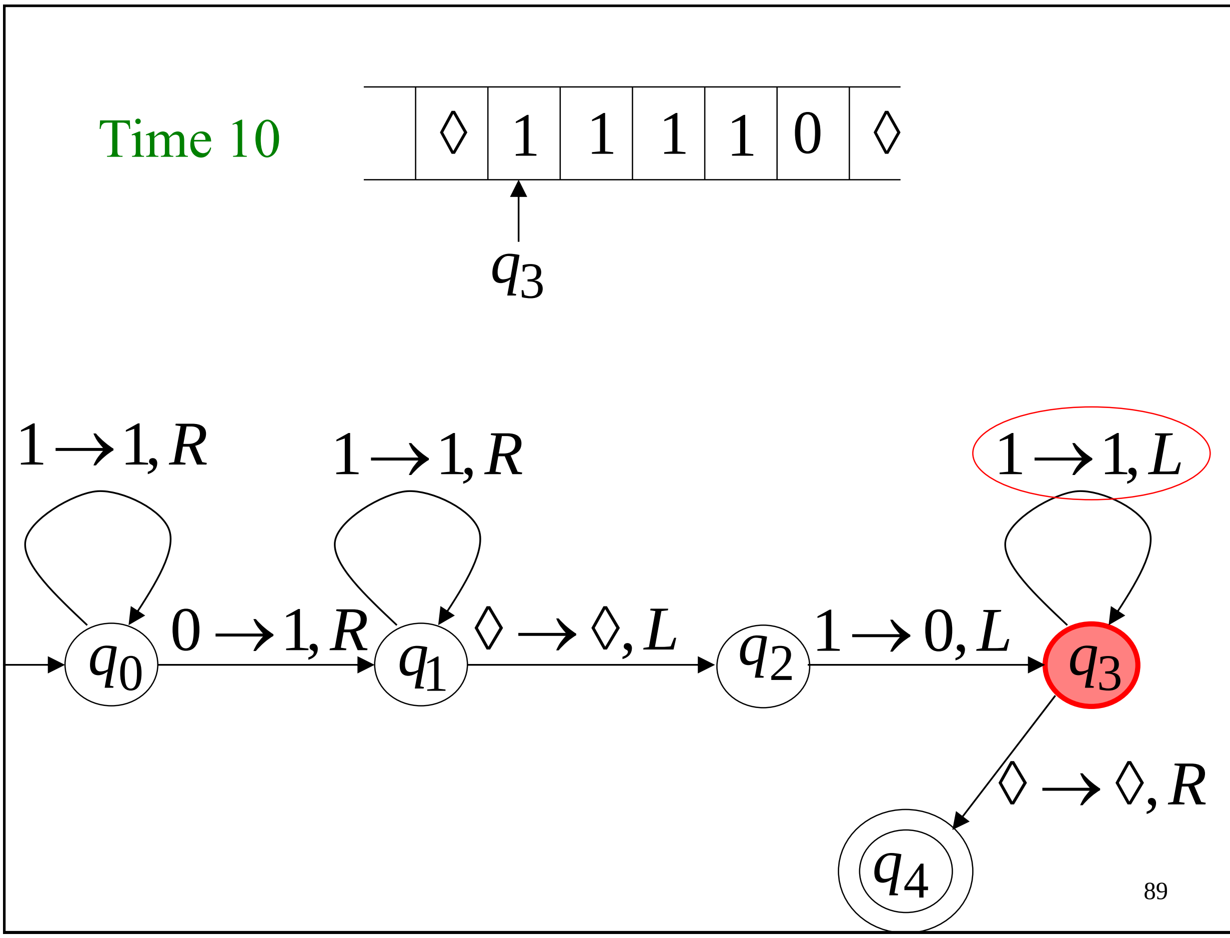

$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
q_0
$0 \to 1, R$
q_1
$\Diamond \to \Diamond, L$
q_2
$1 \to 0, L$
q_3
$\Diamond \to \Diamond, R$
q_4

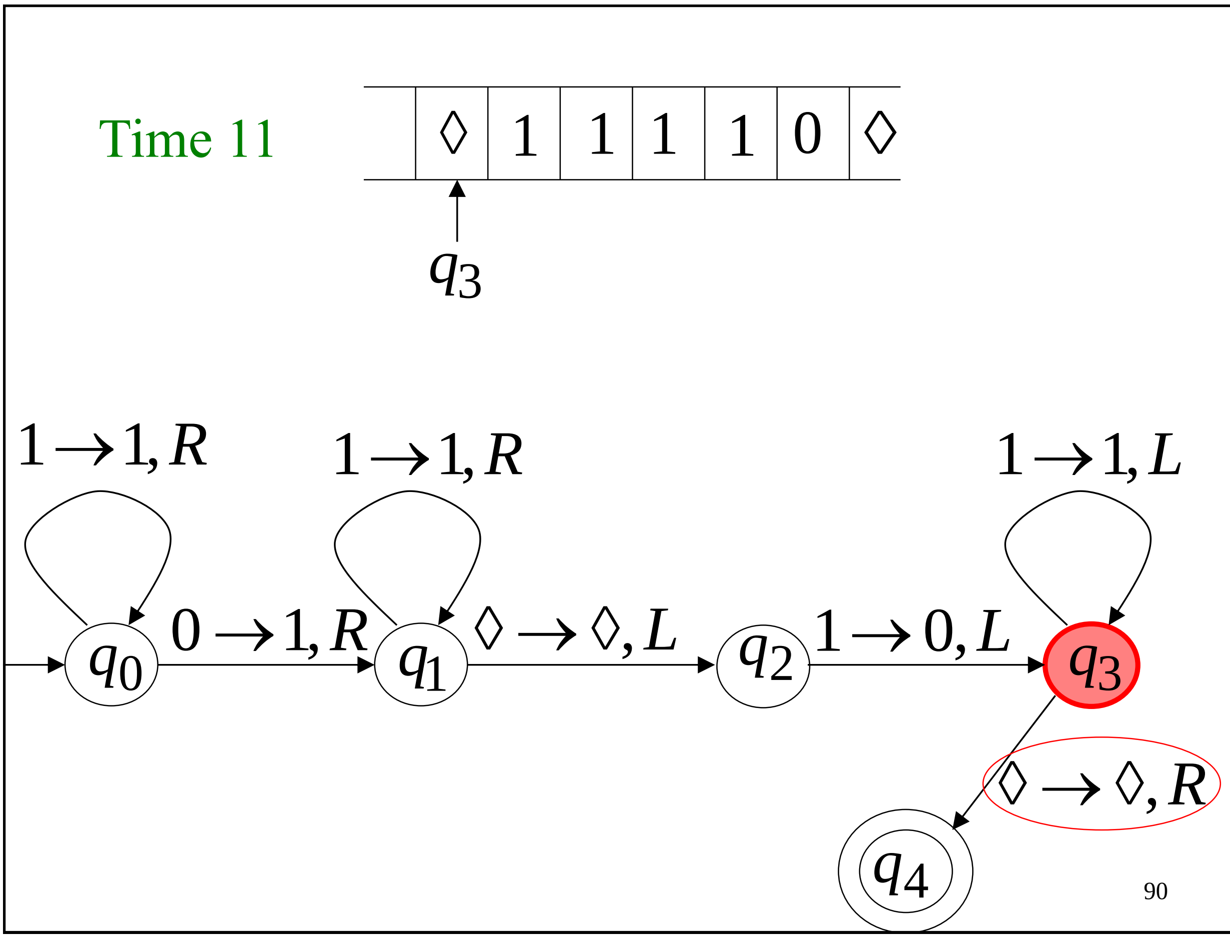
86

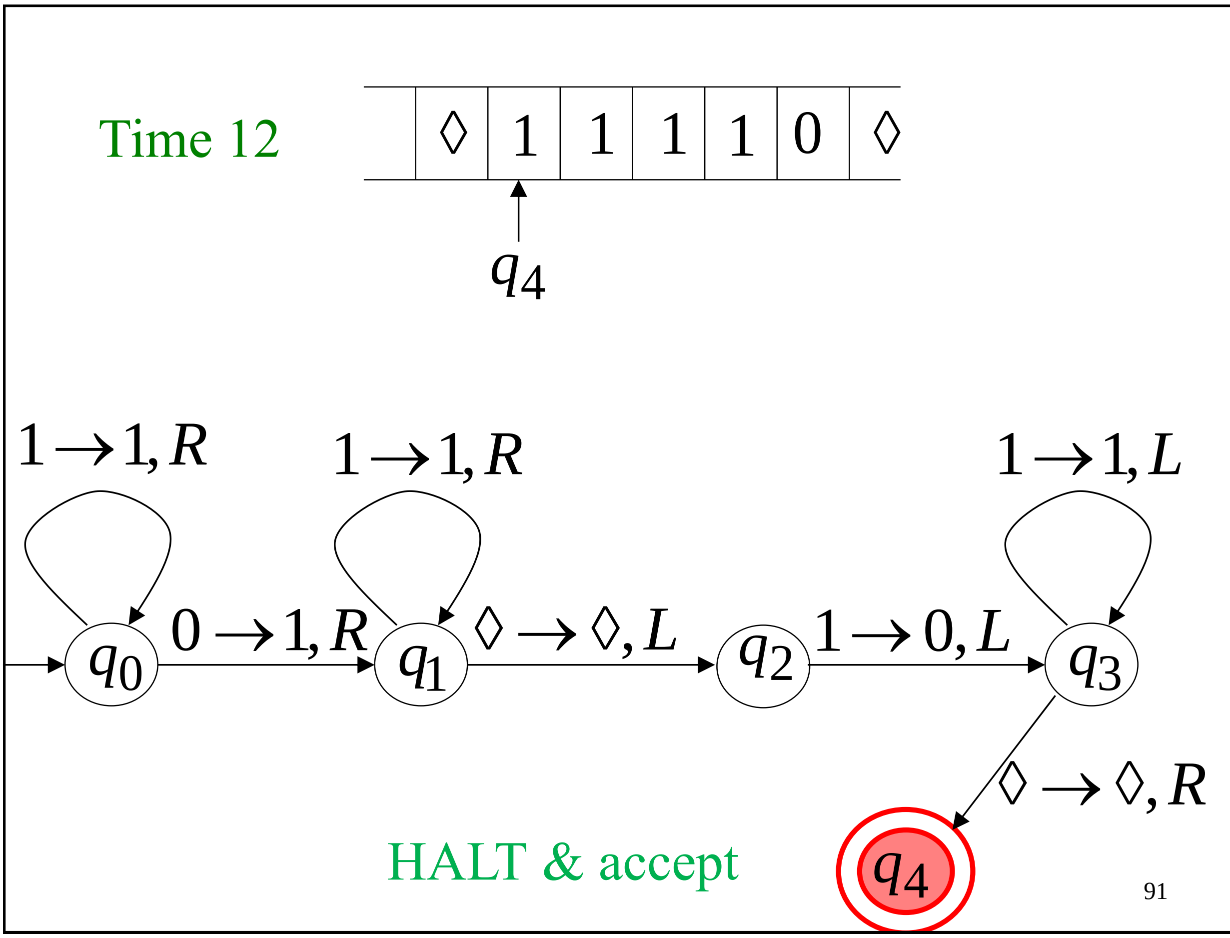

Time 8
◊
1
1
1
1
0
◊
q_3
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\lozenge \rightarrow \lozenge, L$
q_2
$1 \rightarrow 0, L$
q_3
$\lozenge \rightarrow \lozenge, R$
q_4
87



Time 9
◊ 1 1 1 1 0 ◊
q_3
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$1 \rightarrow 0, L$
$\Diamond \rightarrow \Diamond, R$
q_0
q_1
q_2
q_3
q_4
88


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
$◊ \rightarrow ◊, L$
q_2
$1 \rightarrow 0, L$
q_3
$◊ \rightarrow ◊, R$
q_4
89


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
90



Time 12
◊ 1 1 1 1 0 ◊
q_4
$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
$0 \to 1, R$
$\lozenge \to \lozenge, L$
$1 \to 0, L$
$\lozenge \to \lozenge, R$
q_0
q_1
q_2
q_3
q_4
HALT & accept
91

Another Example

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

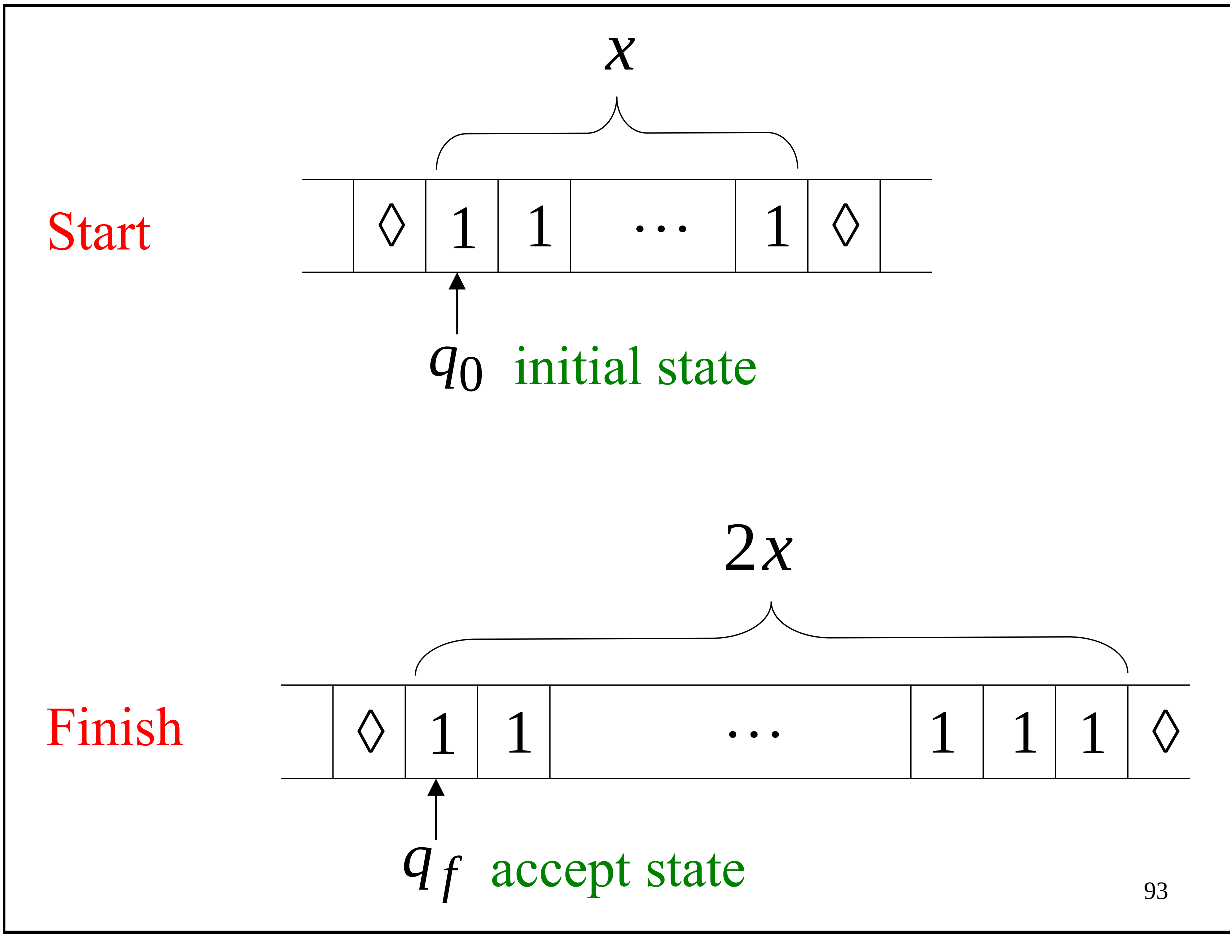
x is an integer

Turing Machine:

Input string: x unary

Output string: xx unary

92


x
Start
◊ 1 1 ⋯ 1 ◊
q_0 initial state
$2x$
Finish
◊ 1 1 ⋯ 1 1 1 ◊
q_f accept state
93

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

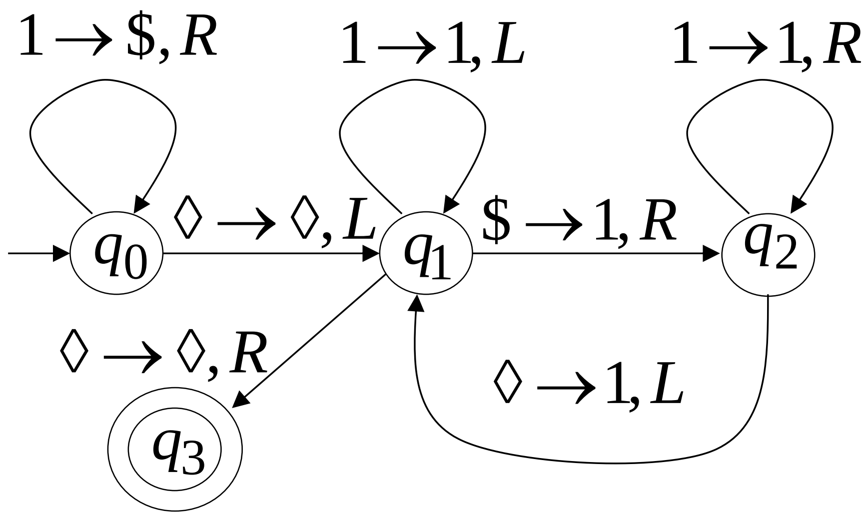
- Replace every 1 with $

- Repeat:

  - Find the rightmost $, replace it with 1
  - Go to the right end, insert 1

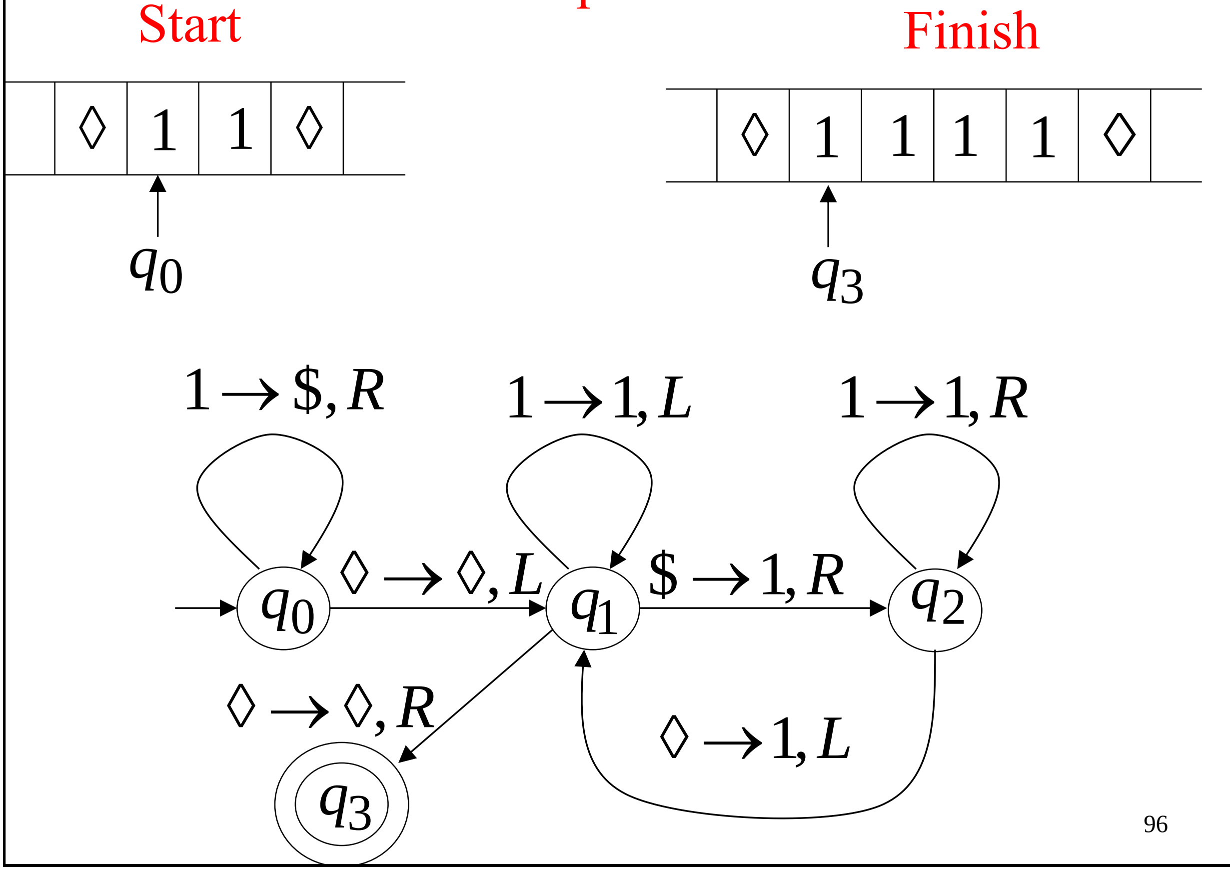
  Until no more $ remain

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



95

Example


Start
◊ 1 1 ◊
q_0
Finish
◊ 1 1 1 1 ◊
q_3
$1 \rightarrow \$, R$
$1 \rightarrow 1, L$
$1 \rightarrow 1, R$
$\Diamond \rightarrow \Diamond, L$
$\$ \rightarrow 1, R$
q_0
q_1
q_2
$\Diamond \rightarrow \Diamond, R$
$\Diamond \rightarrow 1, L$
q_3
96

Another Example

The function is computable

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

Input: $x0y$

Output: 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 $(x > y)$

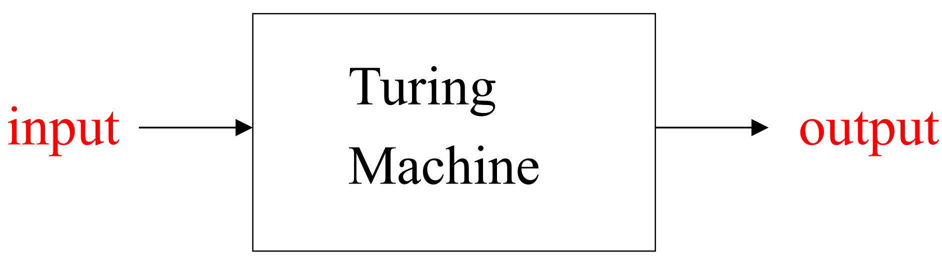
  else

  erase tape, write 0 $(x \leq y)$

98

Combining Turing Machines

Block Diagram



Example:

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