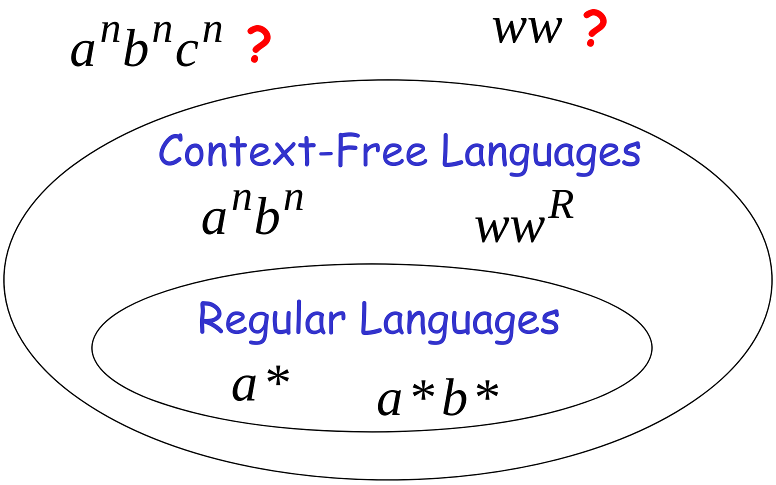
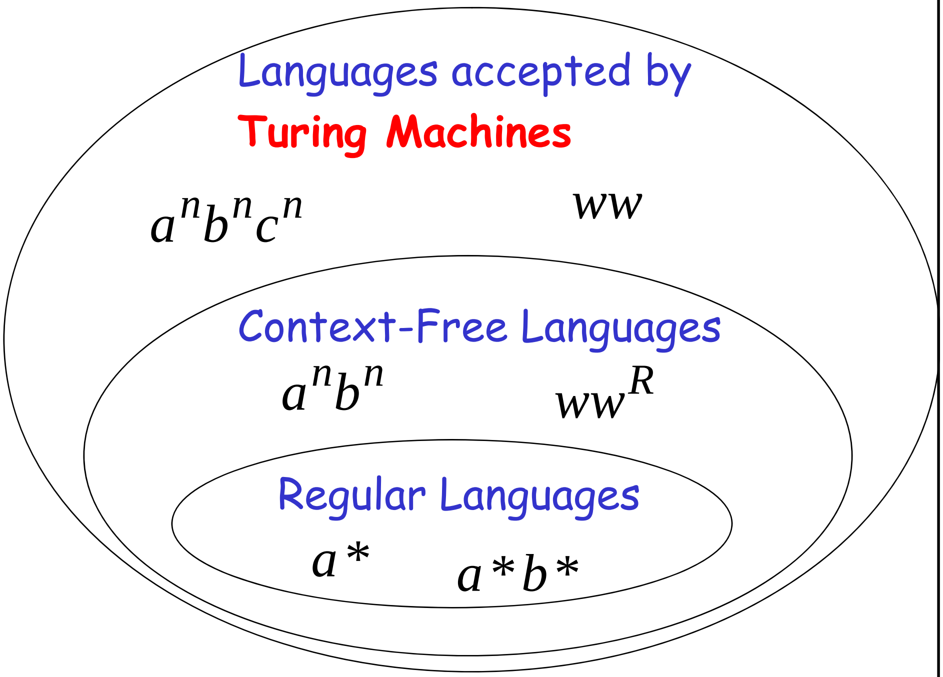
Turing Machines

1

The Language Hierarchy

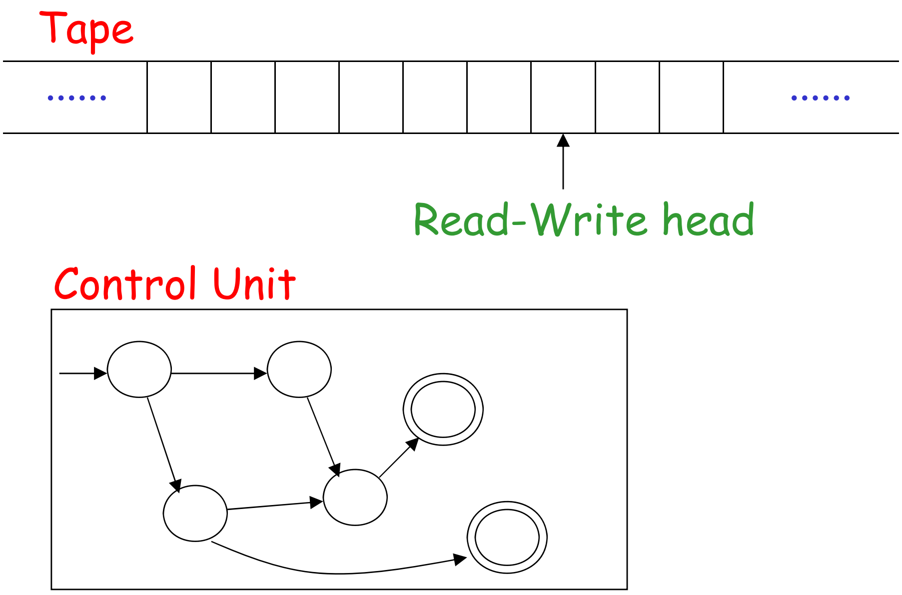


2



3

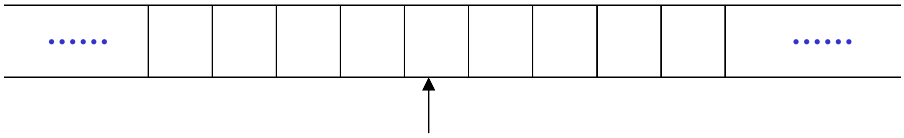
A Turing Machine



4

The Tape

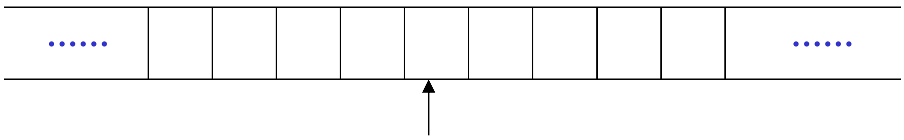
No boundaries -- infinite length



Read-Write head

The head moves Left or Right

5



Read-Write head

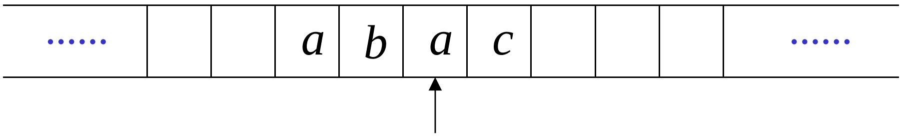
The head at each time step:

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

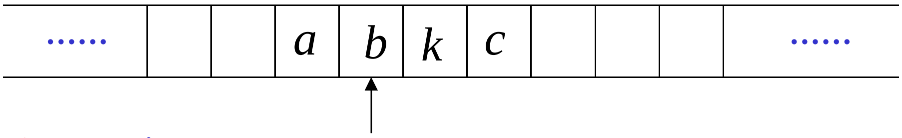
6

Example:

Time 0



Time 1

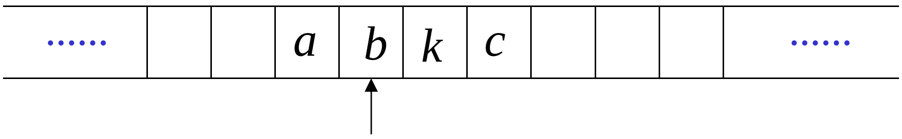


1. Reads a

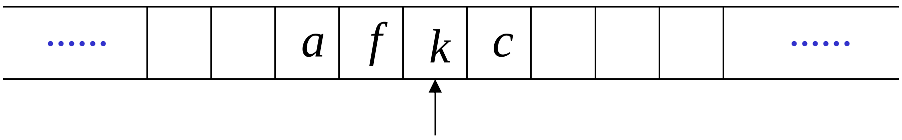
2. Writes k

3. Moves Left

Time 1



Time 2

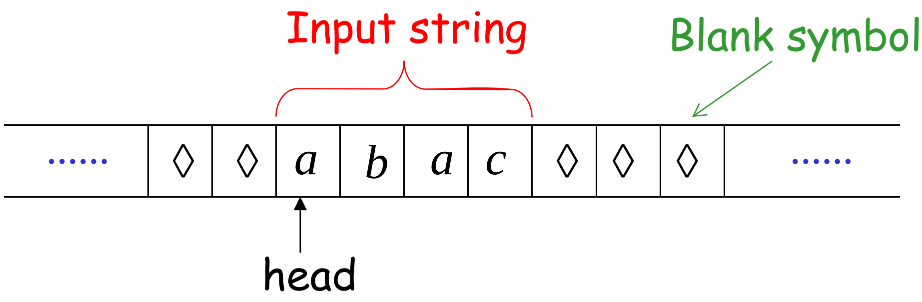


1. Reads b

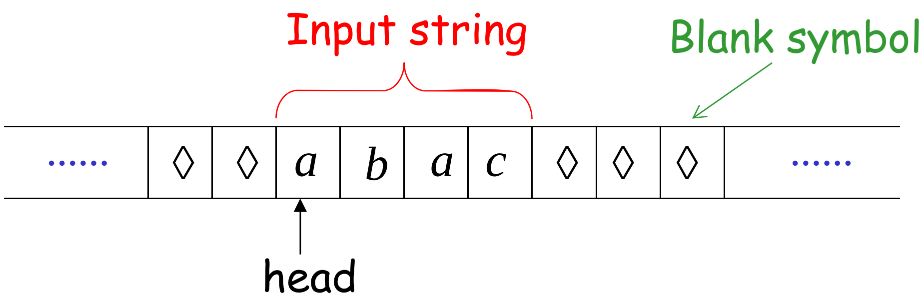
2. Writes f

3. Moves Right

The Input String

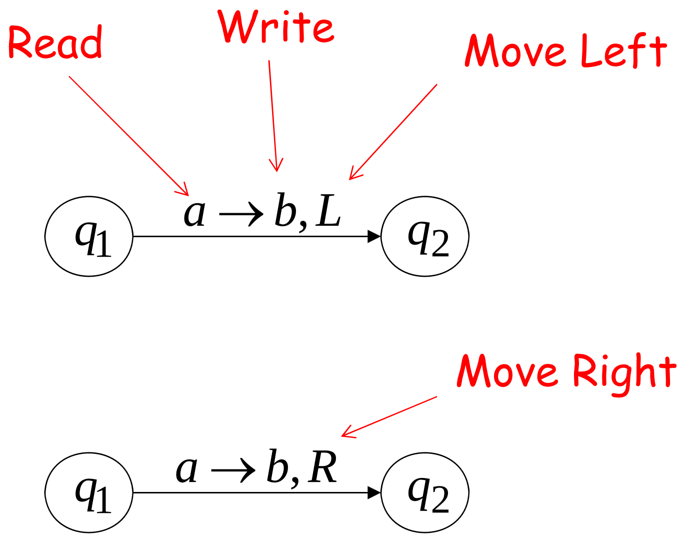


Head starts at the leftmost position of the input string



Remark: the input string is never empty

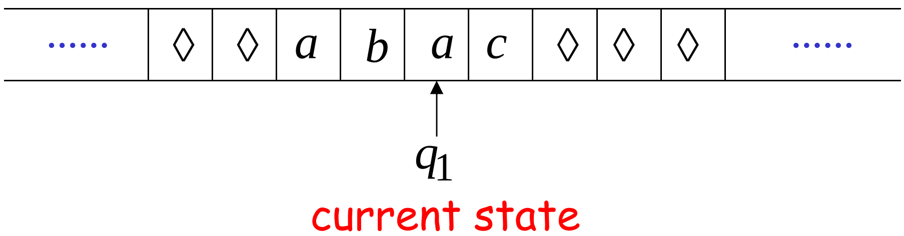
States & Transitions

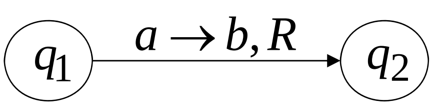


11

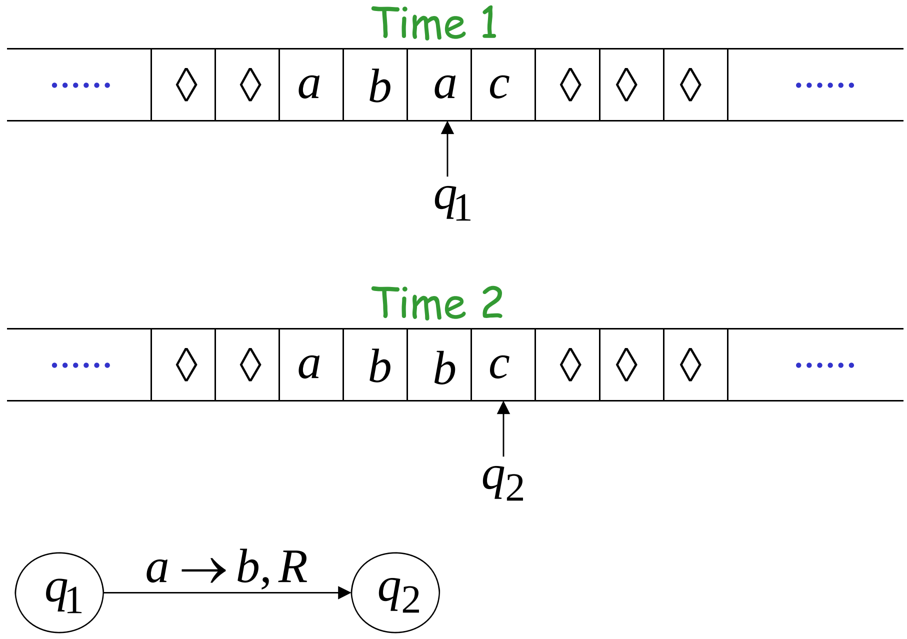
Example:

Time 1



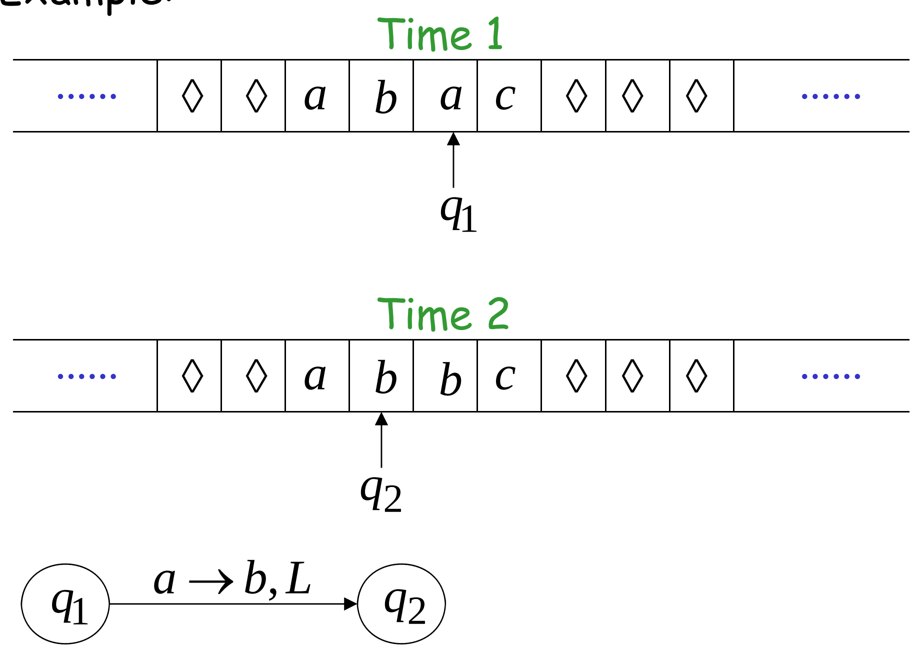


12



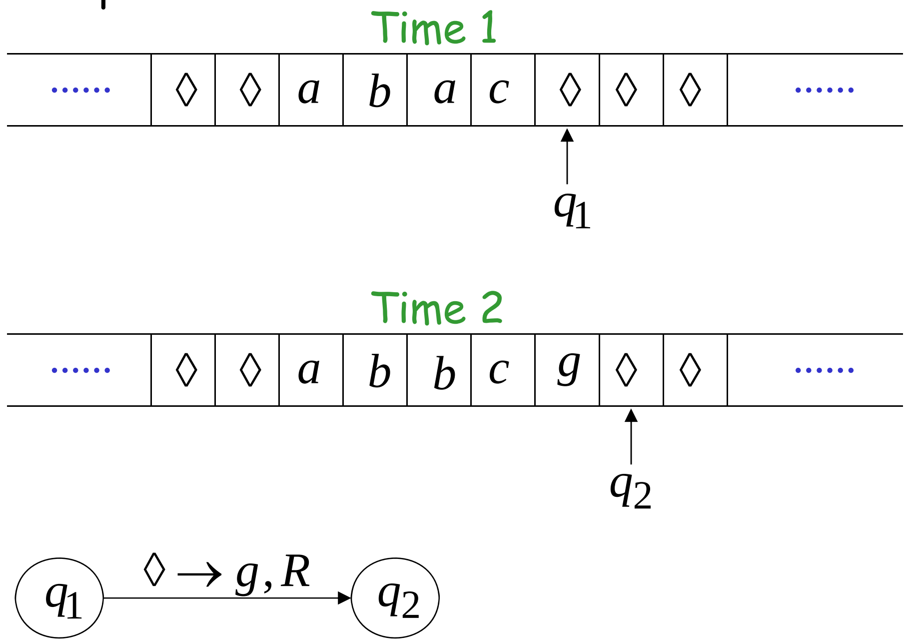
13

Example:



14

Example:

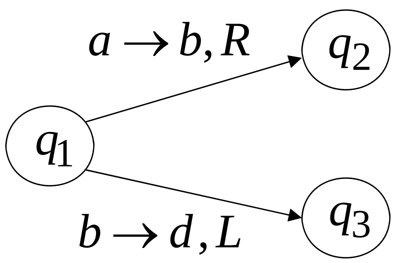


15

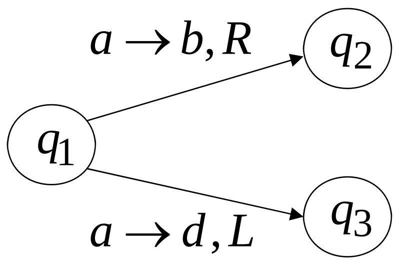
Determinism

Turing Machines are deterministic

Allowed



Not Allowed

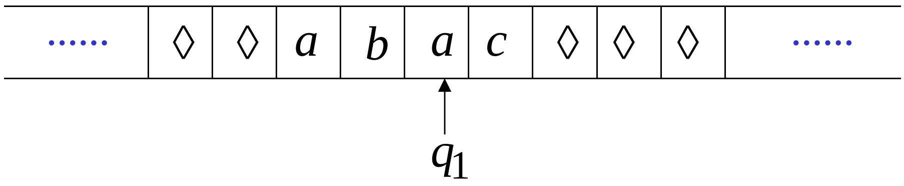


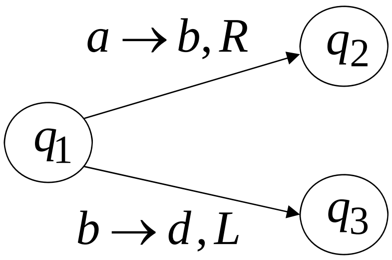
No lambda transitions allowed

16

Partial Transition Function

Example:





Allowed:

No transition
for input symbol c

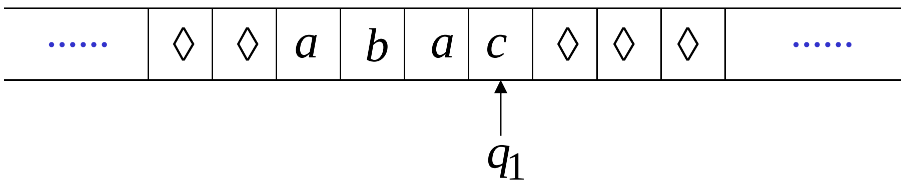
17

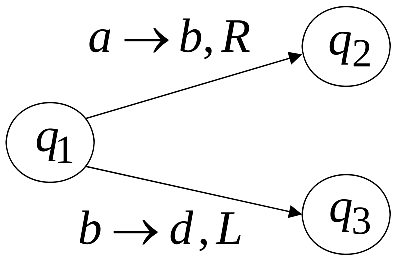
Halting

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

18

Example:



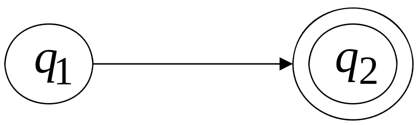


No possible transition

HALT!!!

19

Final States



Allowed

Not Allowed

- Final states have no outgoing transitions
- In a final state the machine halts

20

Acceptance

Accept Input $\Rightarrow$ If machine halts in a final state

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

or

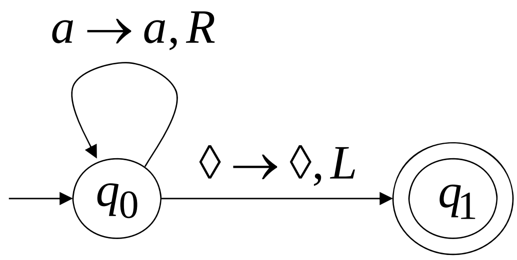
If machine enters an *infinite loop*

21

Turing Machine Example

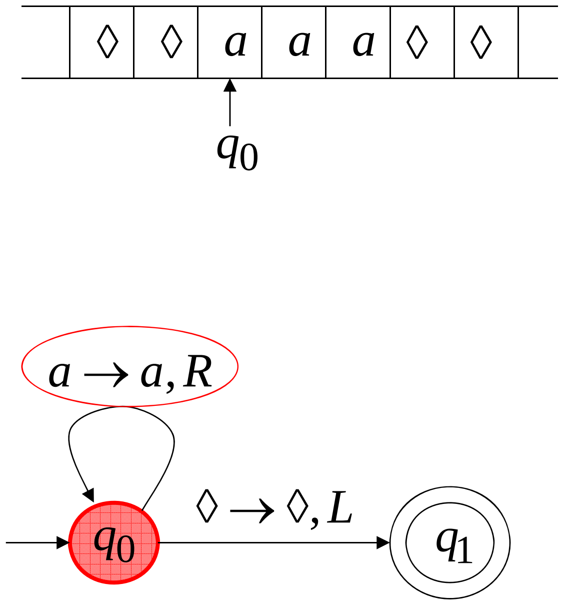
A Turing machine that accepts the language:

a^*



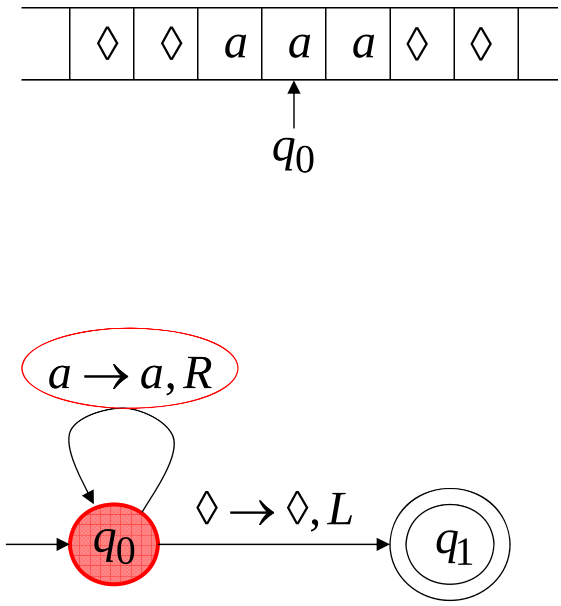
22

Time 0



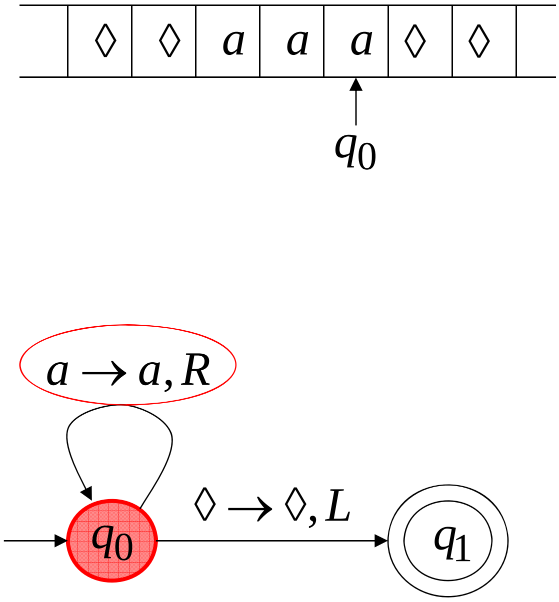
23

Time 1



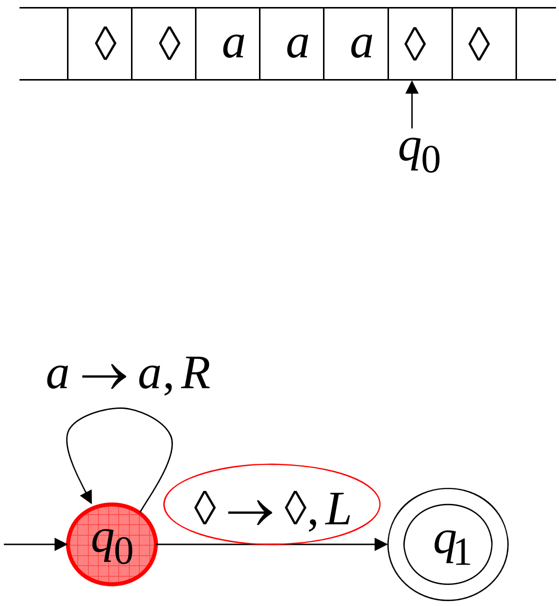
24

Time 2



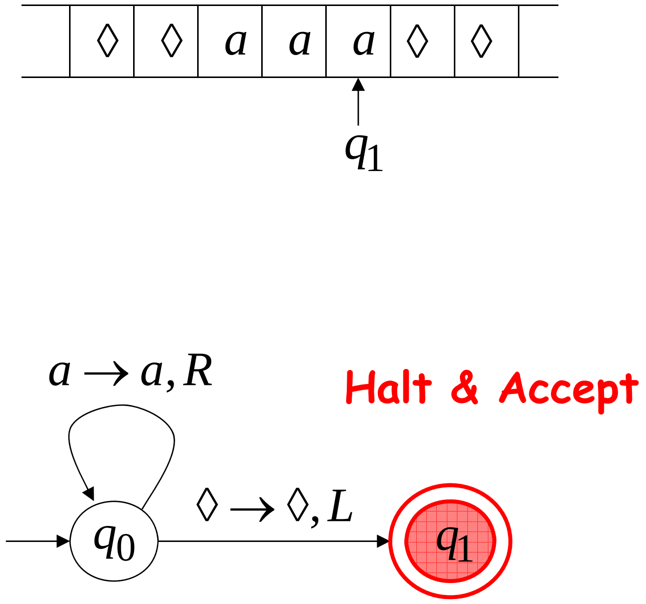
25

Time 3



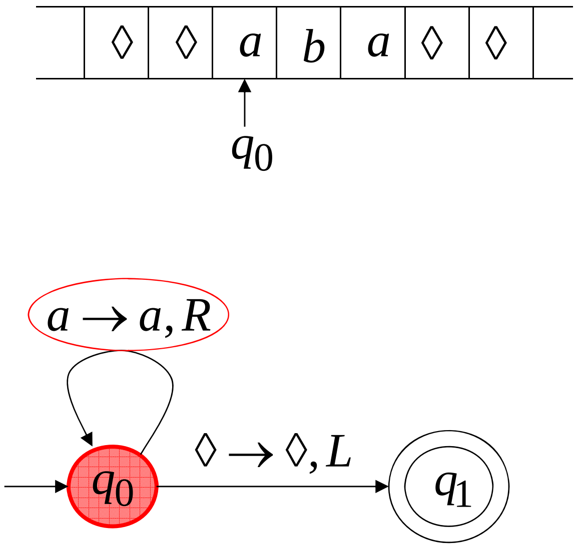
26

Time 4

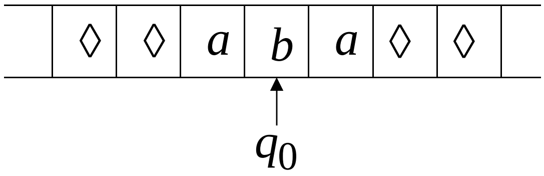


Rejection Example

Time 0

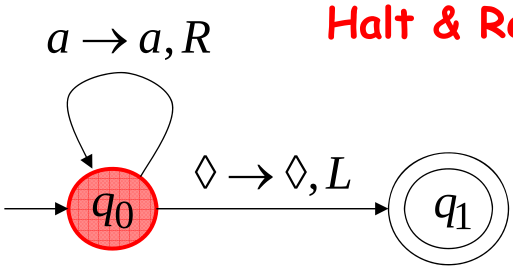


Time 1



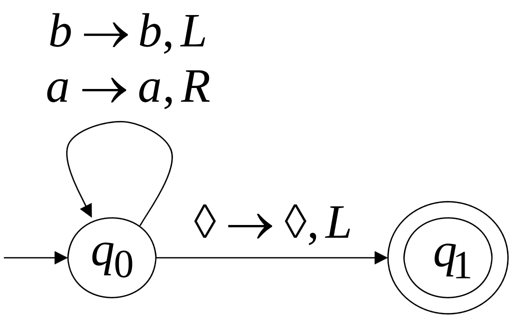
No possible Transition

Halt & Reject



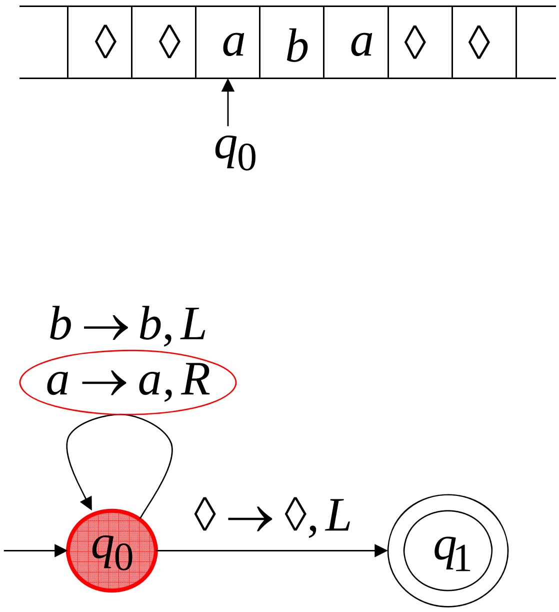
29

Infinite Loop Example



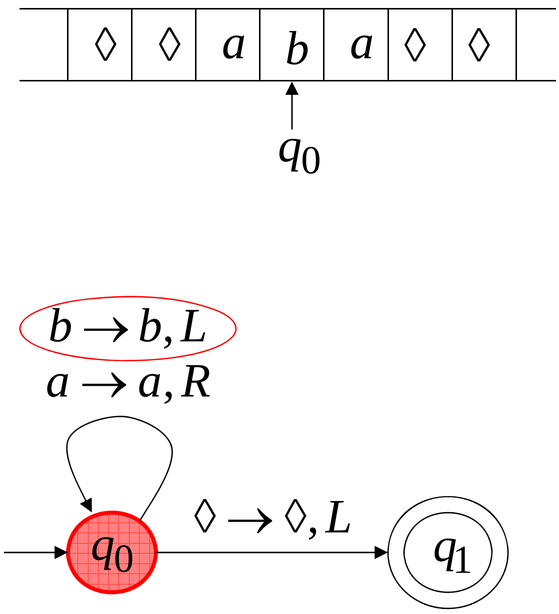
30

Time 0

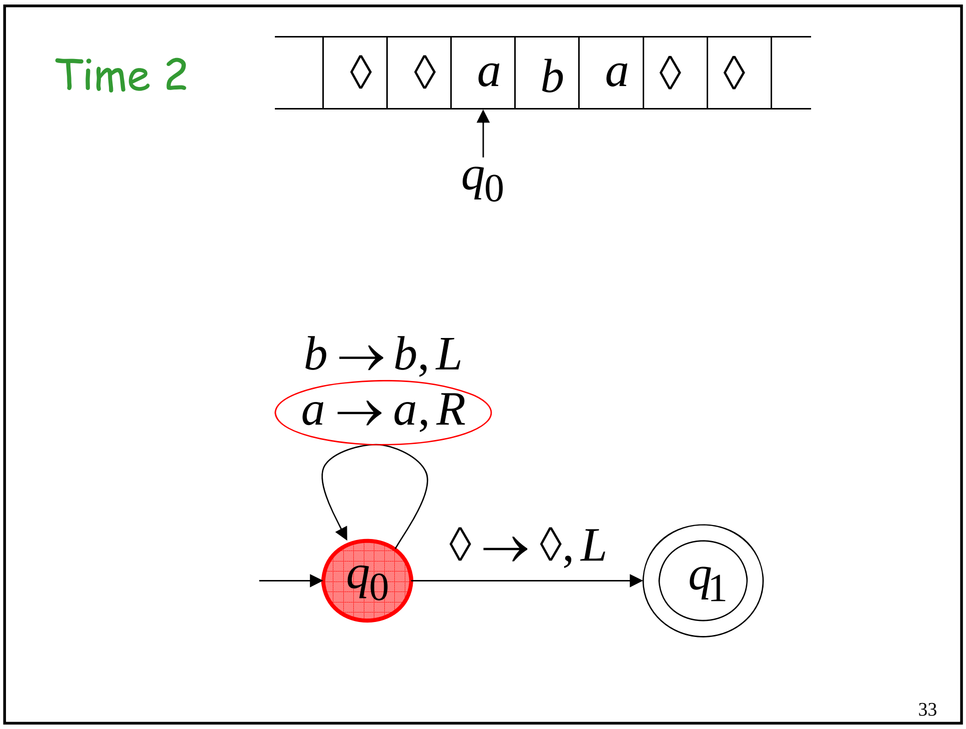


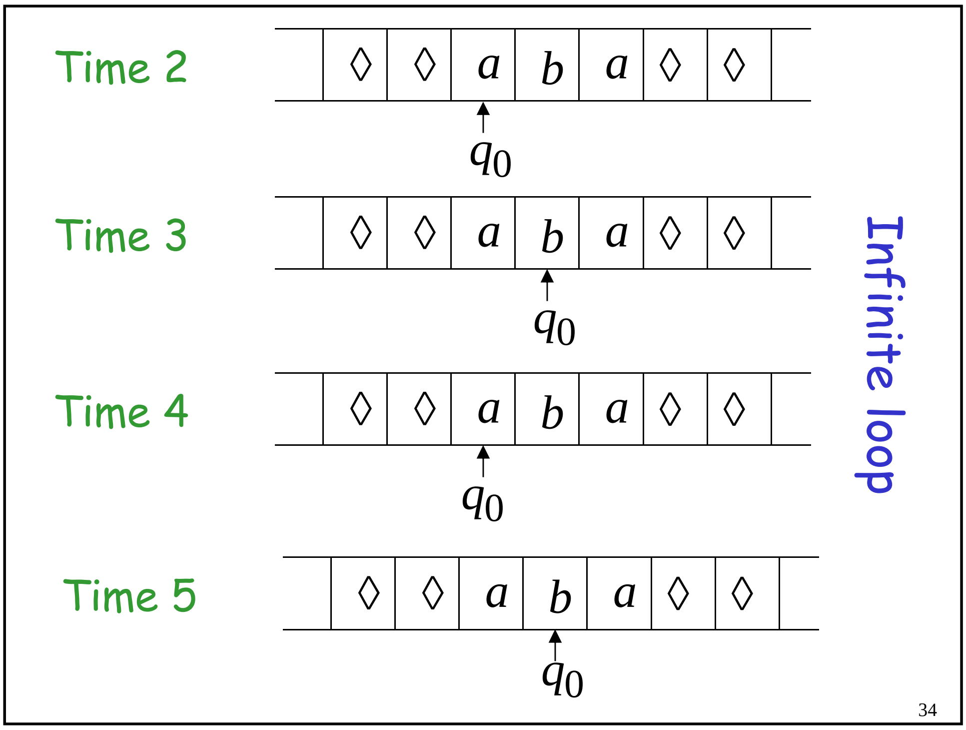
31

Time 1



32


Time 2
◊ ◊ a b a ◊ ◊
q0
b → b, L
a → a, R
◊ → ◊, L
q0
q1
33


Time 2
◊ ◊ a b a ◊ ◊
q0
Time 3
◊ ◊ a b a ◊ ◊
q0
Time 4
◊ ◊ a b a ◊ ◊
q0
Time 5
◊ ◊ a b a ◊ ◊
q0
Infinite loop
34

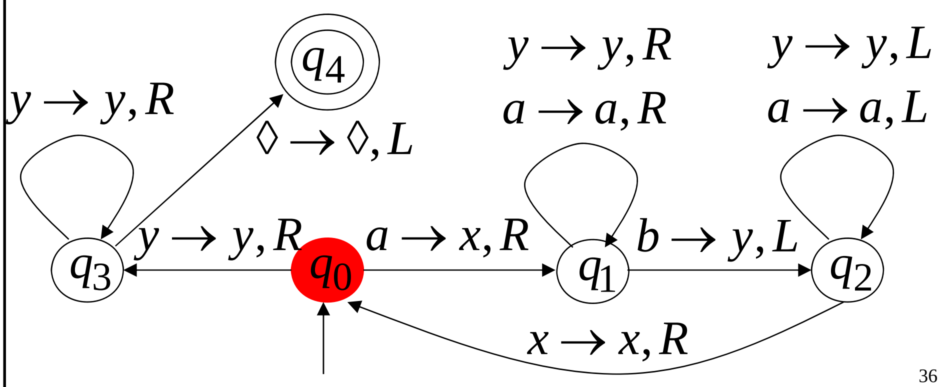
Because of the infinite loop:

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

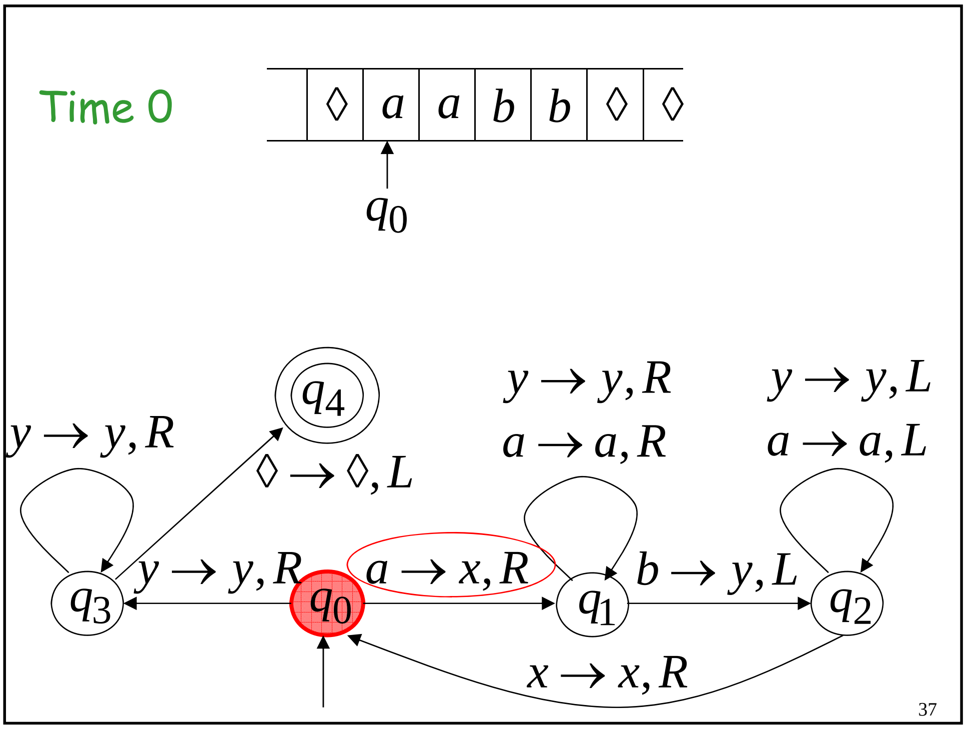
35

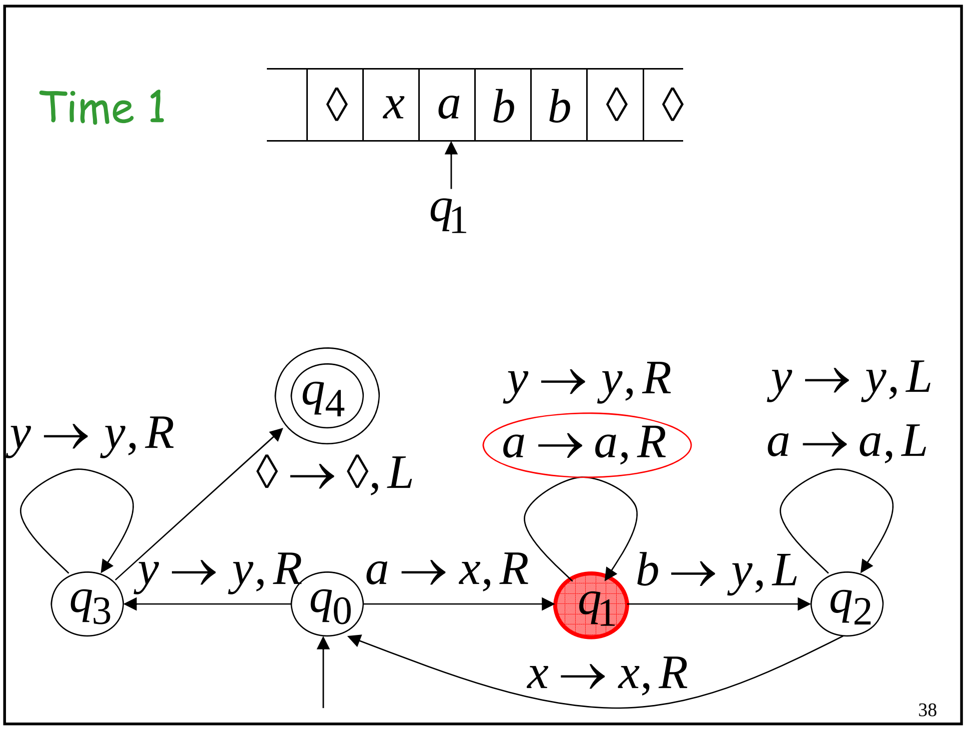
Another Turing Machine Example

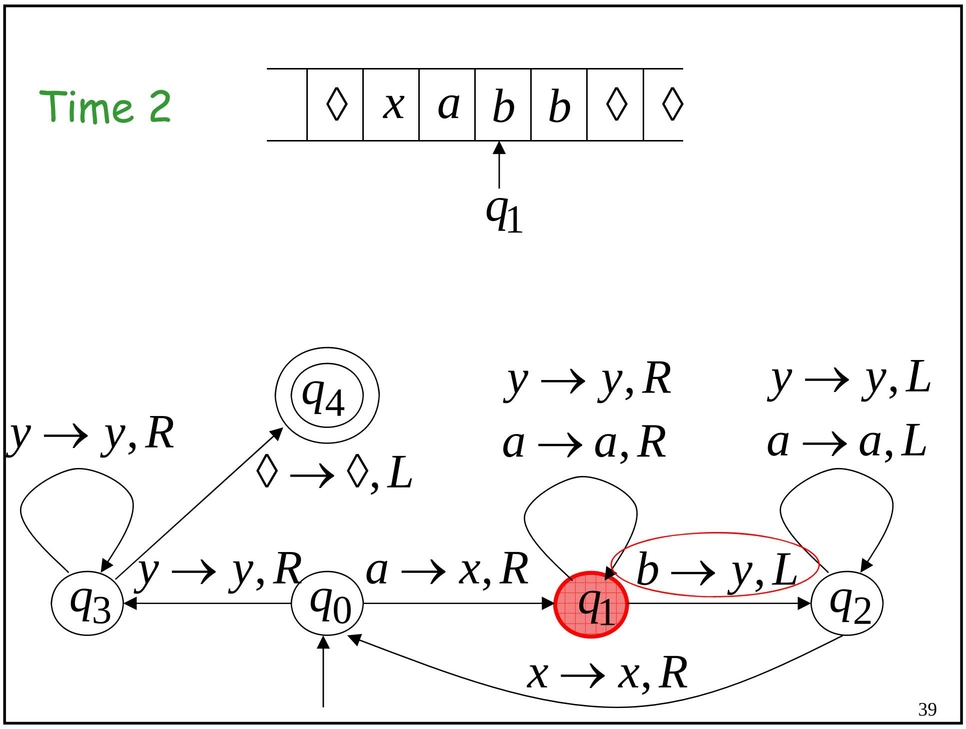
Turing machine for the language $\{a^n b^n\}$

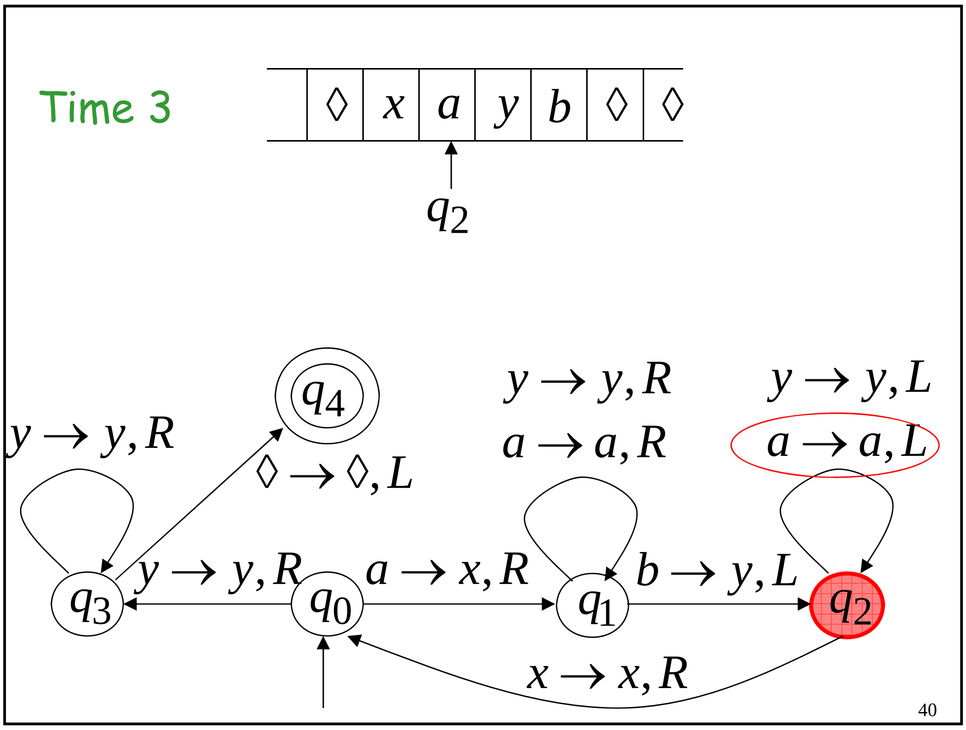


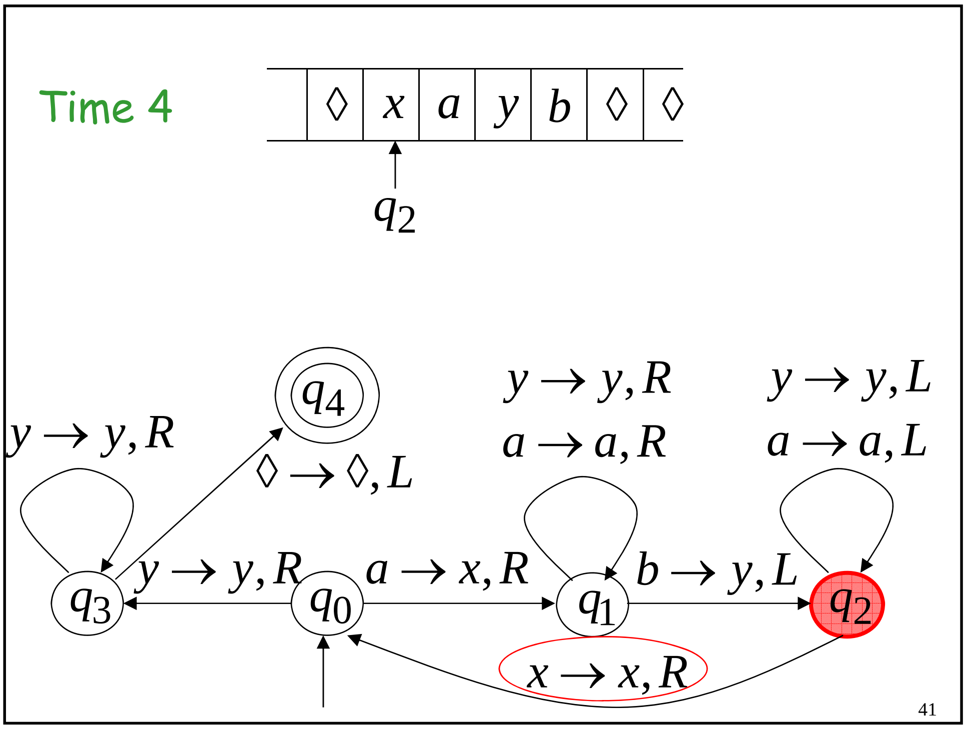
36

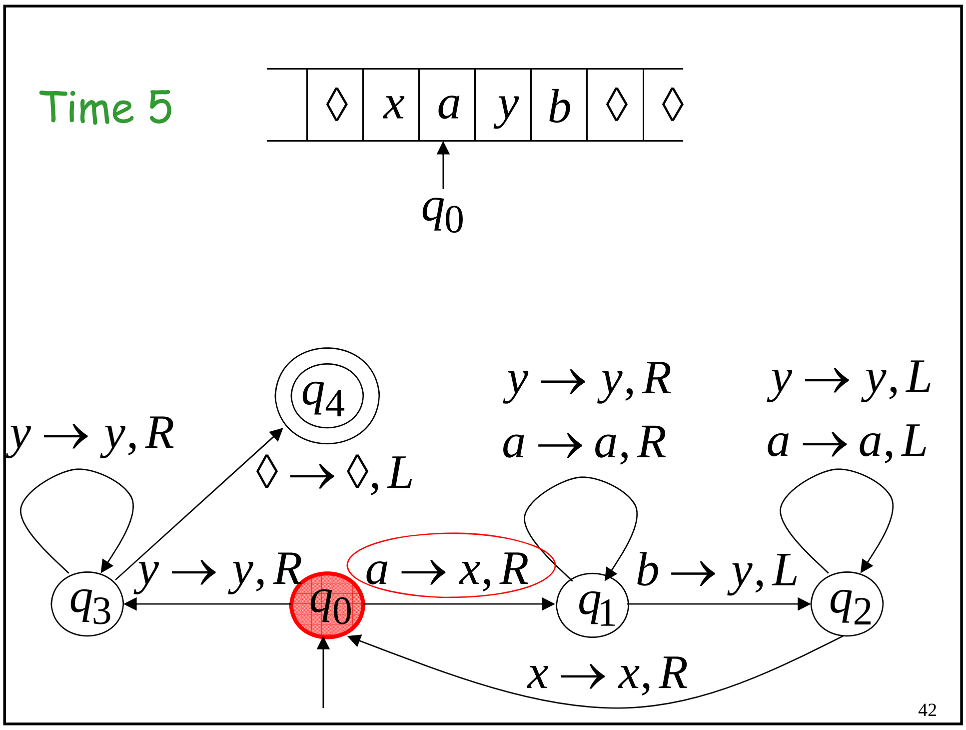

Time 0
◊
a
a
b
b
◊
◊
q_0
q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$
37


Time 1
◊
x
a
b
b
◊
◊
q_1
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
$x \to x, R$
38


Time 2
◊ x a b b ◊ ◊
q_1
$y \rightarrow y, R$
q_4
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$
39


Time 3
◊ x a y b ◊ ◊
q_2
$y \rightarrow y, R$
q_4
$\Diamond \rightarrow \Diamond, L$
$y \rightarrow y, R$
$a \rightarrow a, R$
$y \rightarrow y, L$
$a \rightarrow a, L$
$y \rightarrow y, R$
$a \rightarrow x, R$
$b \rightarrow y, L$
q_3
q_0
q_1
q_2
$x \rightarrow x, R$
40

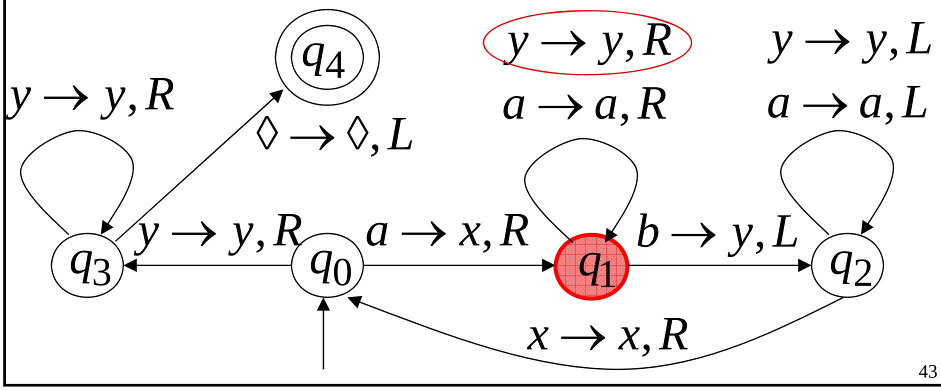

Time 4
◊ x a y b ◊ ◊
q2
q4
y → y, R
◊ → ◊, L
y → y, R
a → a, R
y → y, L
a → a, L
y → y, R
a → x, R
b → y, L
q3
q0
q1
q2
x → x, R
41


Time 5
◊ x a y b ◊ ◊
q0
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
42

Time 6

	◊	x	x	y	b	◊	◊

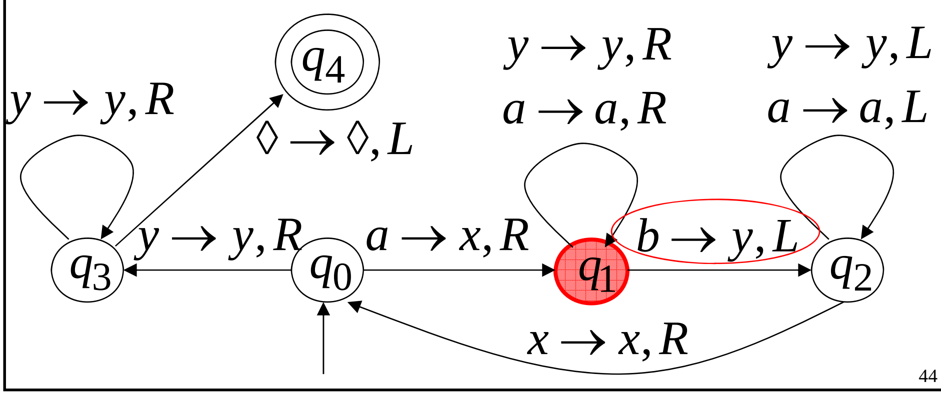
$\uparrow$
q_1

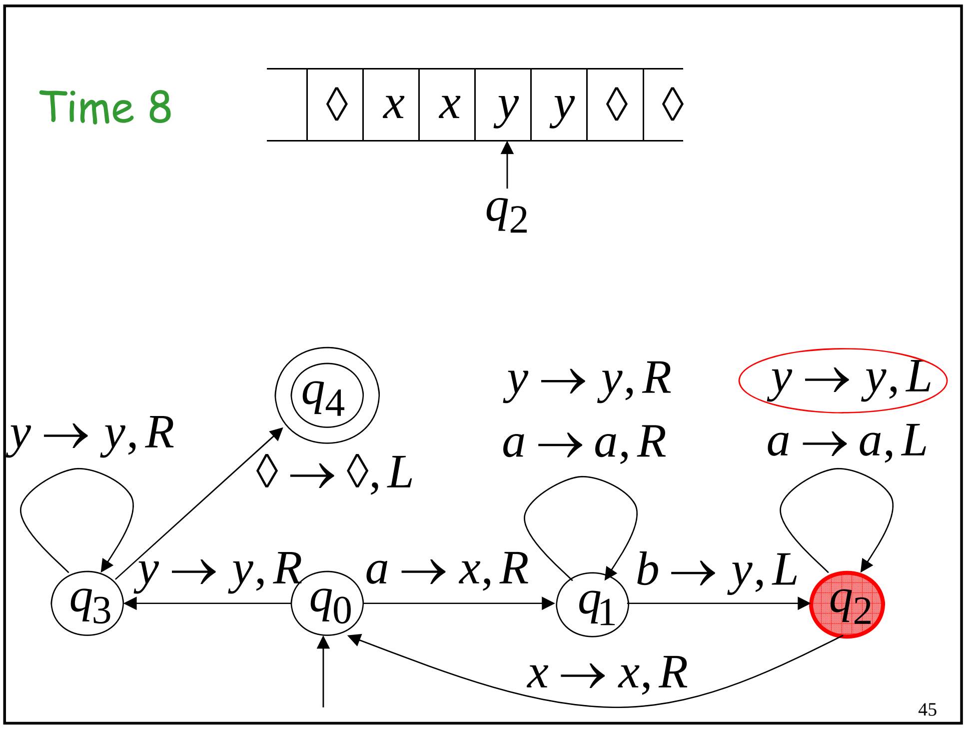


Time 7

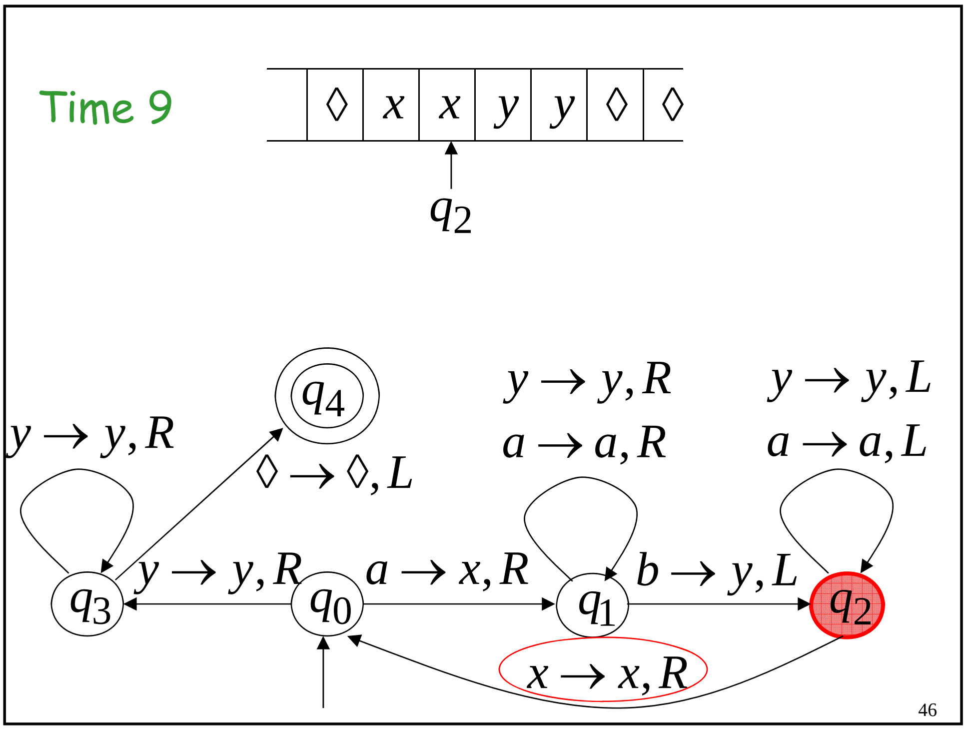
	◊	x	x	y	b	◊	◊

$\uparrow$
q_1

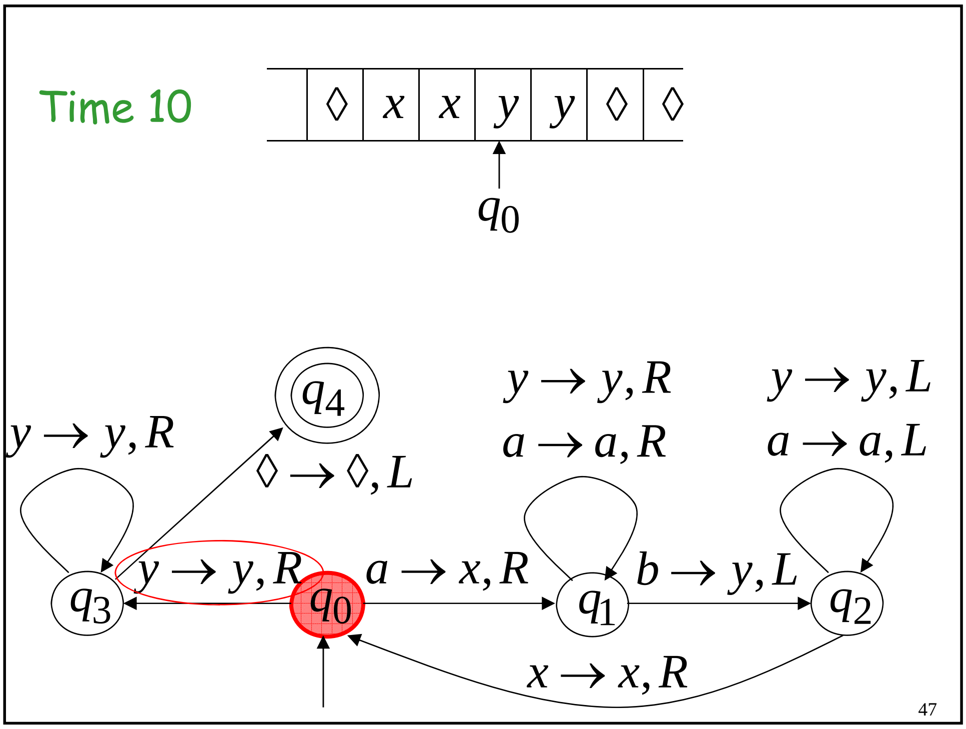


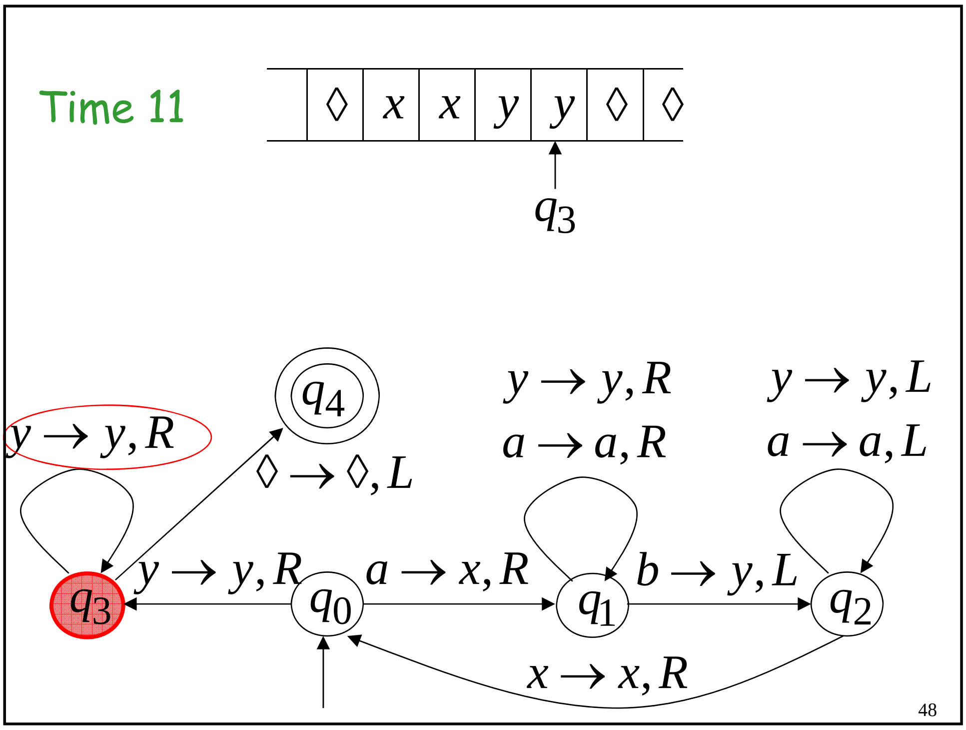


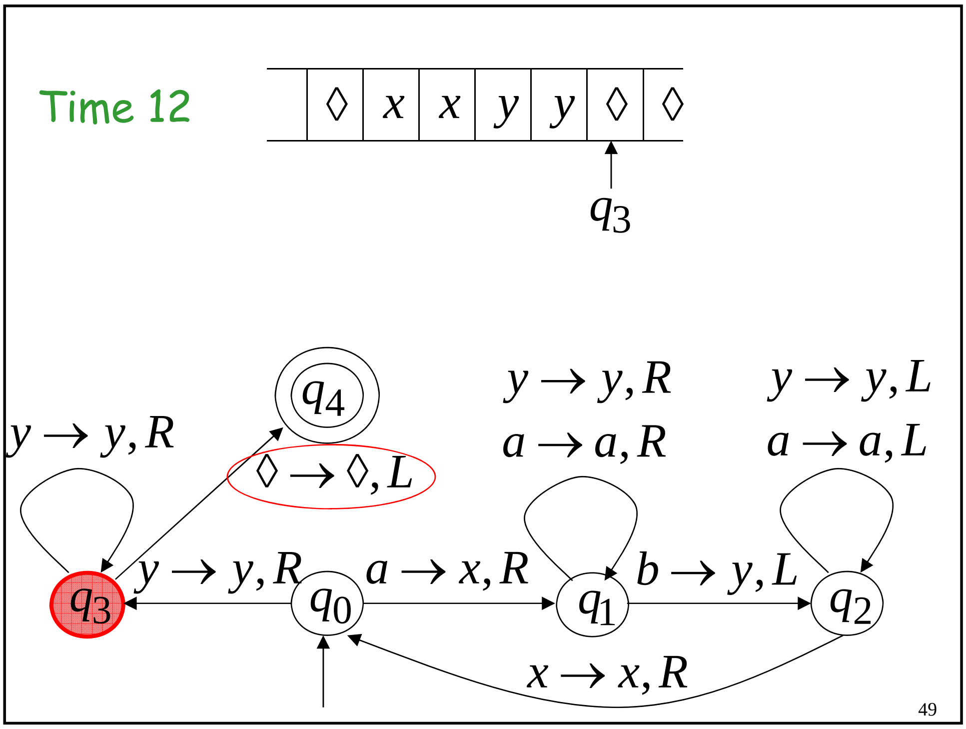
Time 8
◊ x x y y ◊ ◊
q_2
q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$
45

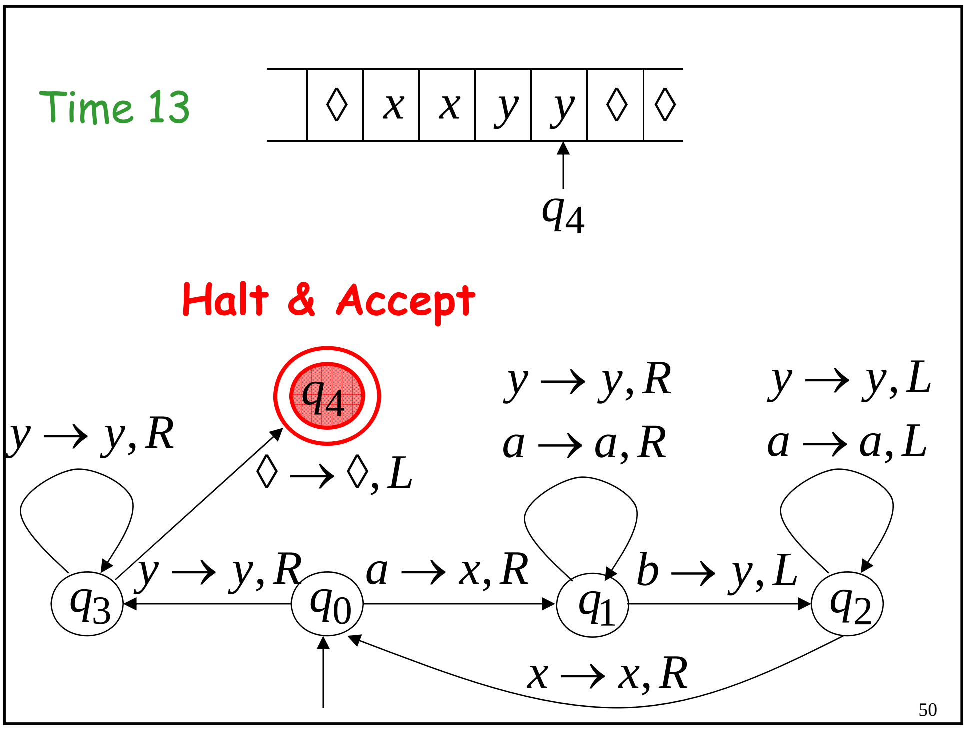


Time 9
◊ x x y y ◊ ◊
q_2
q_4
$y \to y, R$
$\Diamond \to \Diamond, L$
$y \to y, R$
$a \to a, R$
$y \to y, L$
$a \to a, L$
q_3
$y \to y, R$
q_0
$a \to x, R$
q_1
$b \to y, L$
q_2
$x \to x, R$
46


Time 10
◊ x x y y ◊ ◊
q0
q4
y → y, R
a → a, R
y → y, L
a → a, L
y → y, R
◊ → ◊, L
q3
y → y, R
q0
a → x, R
q1
b → y, L
q2
x → x, R
47


Time 11
◊ x x y y ◊ ◊
q3
q4
y → y, R
a → a, R
y → y, L
a → a, L
y → y, R
◊ → ◊, L
q3
y → y, R
q0
a → x, R
q1
b → y, L
q2
x → x, R
48


Time 12
◊ x x y y ◊ ◊
q3
q4
y → y, R
◊ → ◊, L
y → y, R
a → a, R
y → y, L
a → a, L
q3
y → y, R
q0
a → x, R
q1
b → y, L
q2
x → x, R
49


Time 13
◊ x x y y ◊ ◊
q4
Halt & Accept
q4
y → y, R
◊ → ◊, L
y → y, R
a → a, R
y → y, L
a → a, L
q3
y → y, R
q0
a → x, R
q1
b → y, L
q2
x → x, R
50

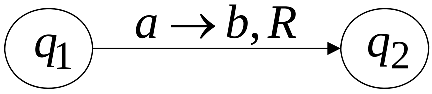
Observation:

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

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

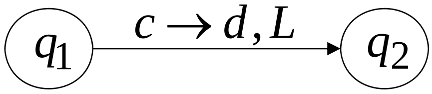
Formal Definitions
for
Turing Machines

Transition Function



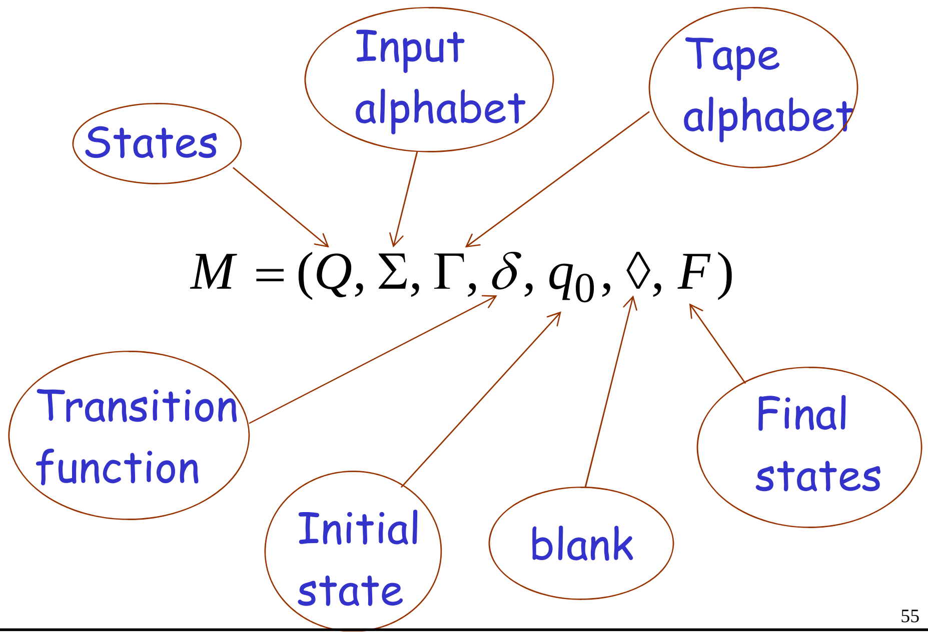
$$\delta(q_1, a) = (q_2, b, R)$$

Transition Function



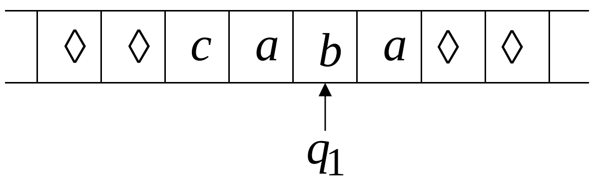
$$\delta(q_1, c) = (q_2, d, L)$$

Turing Machine:



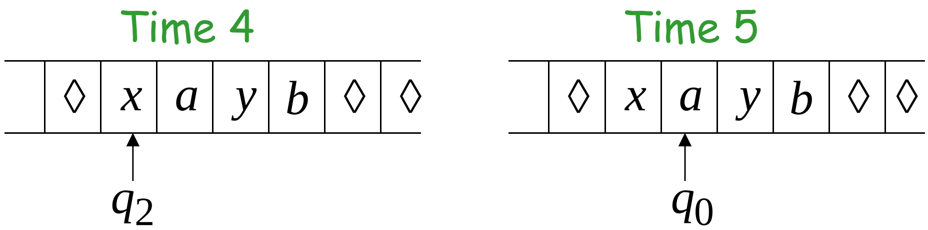
55

Configuration

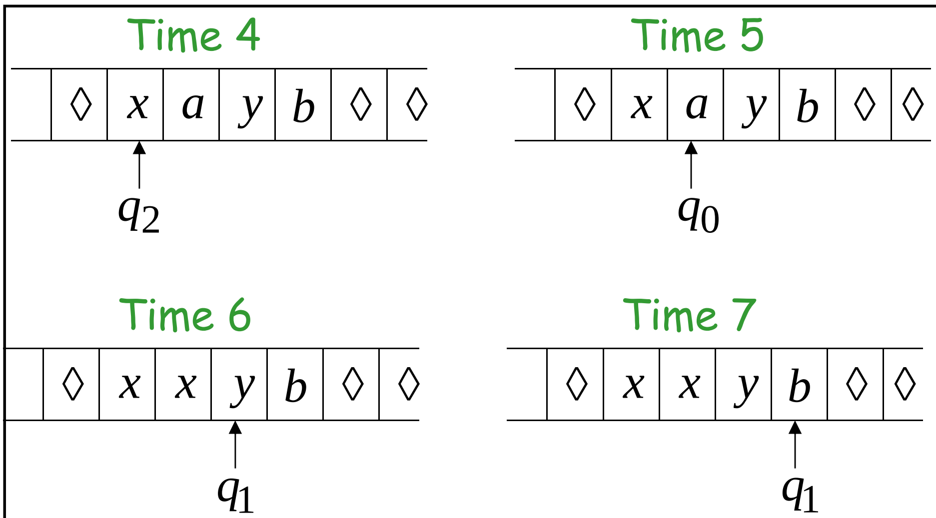


Instantaneous description: $ca \; q_1 \; ba$

56



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



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

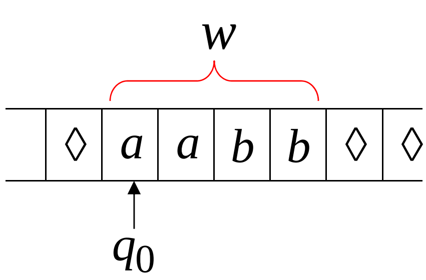
$$q_2\ xayb \succ x\ q_0\ ayb \succ xx\ q_1\ yb \succ xxy\ q_1\ b$$

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

59

Initial configuration: $q_0\ w$

Input string



60

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

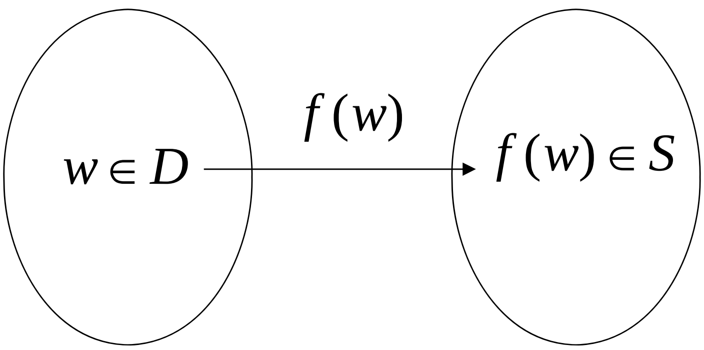
Computing Functions
with
Turing Machines

63

A function $f(w)$ has:

Domain: D

Result Region: S



64

A function may have many parameters:

Example: Addition function

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

Integer Domain

Decimal: 5

Binary: 101

Unary: 11111

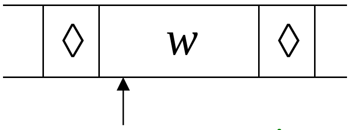
We prefer **unary** representation:

easier to manipulate with Turing machines

Definition:

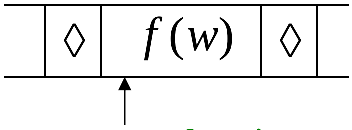
A function f is computable if
there is a Turing Machine M such that:

Initial configuration



q_0 initial state

Final configuration



q_f final state

For all $w \in D$ Domain

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

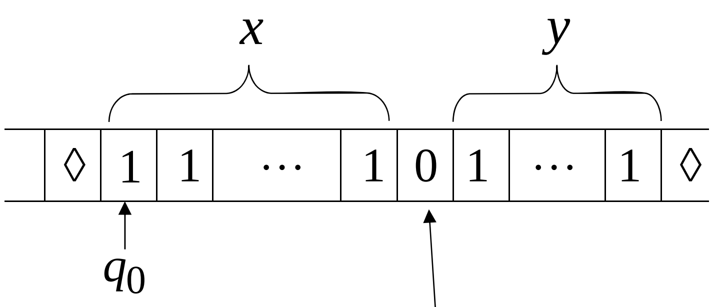
x, y are integers

Turing Machine:

Input string: $x0y$ unary

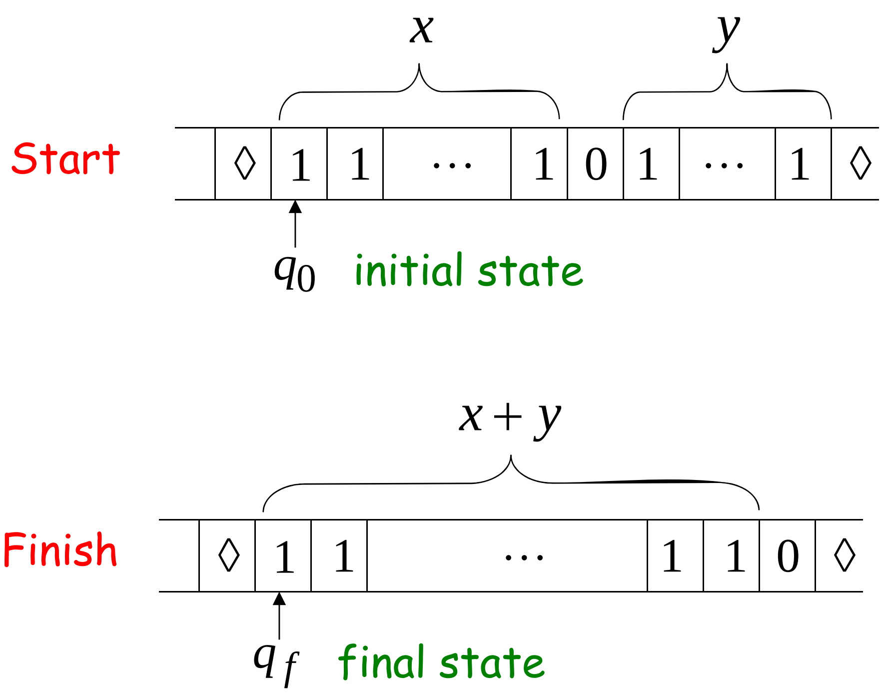
Output string: $xy0$ unary

Start



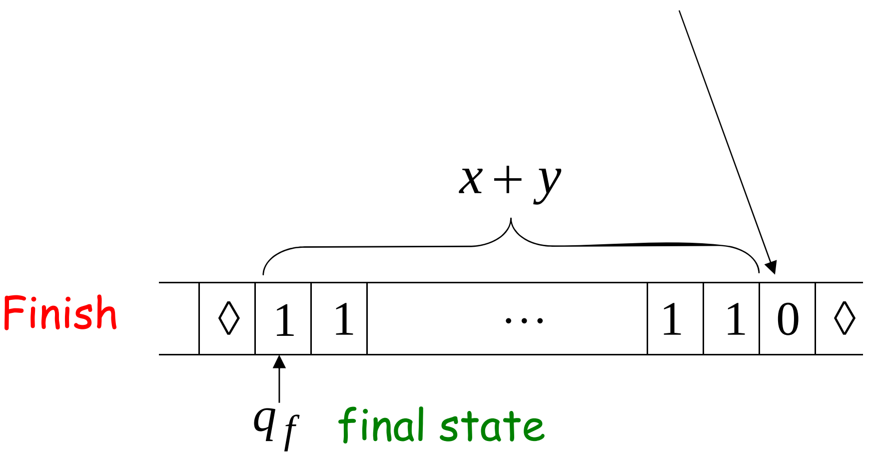
initial state

The 0 is the delimiter that
separates the two numbers



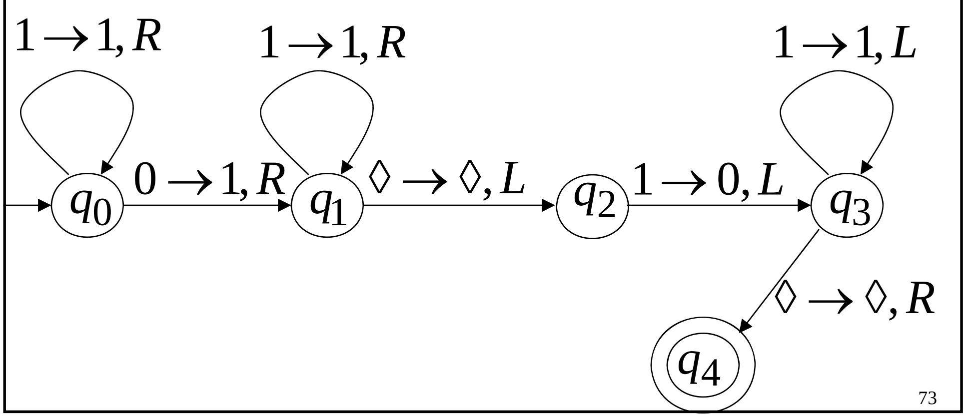
71

The 0 helps when we use
the result for other operations



72

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

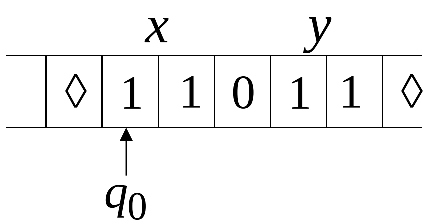


Execution Example:

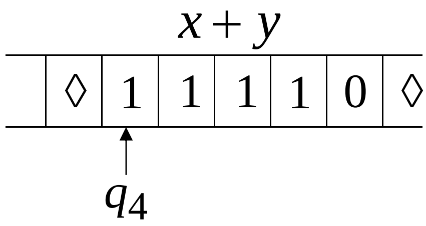
$x = 11$ (2)

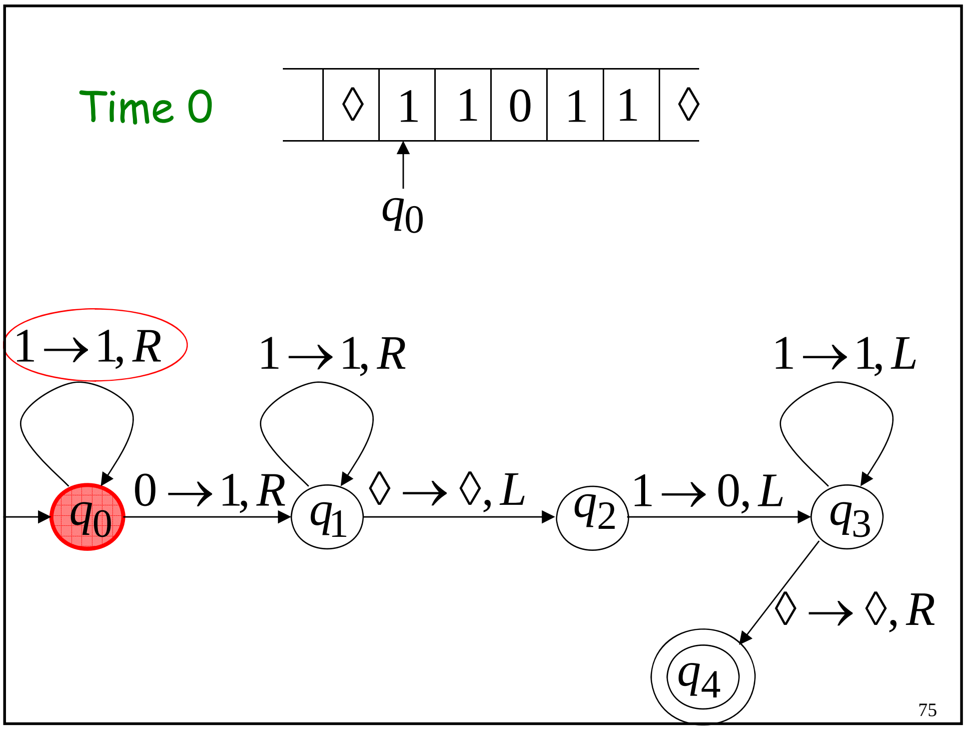
$y = 11$ (2)

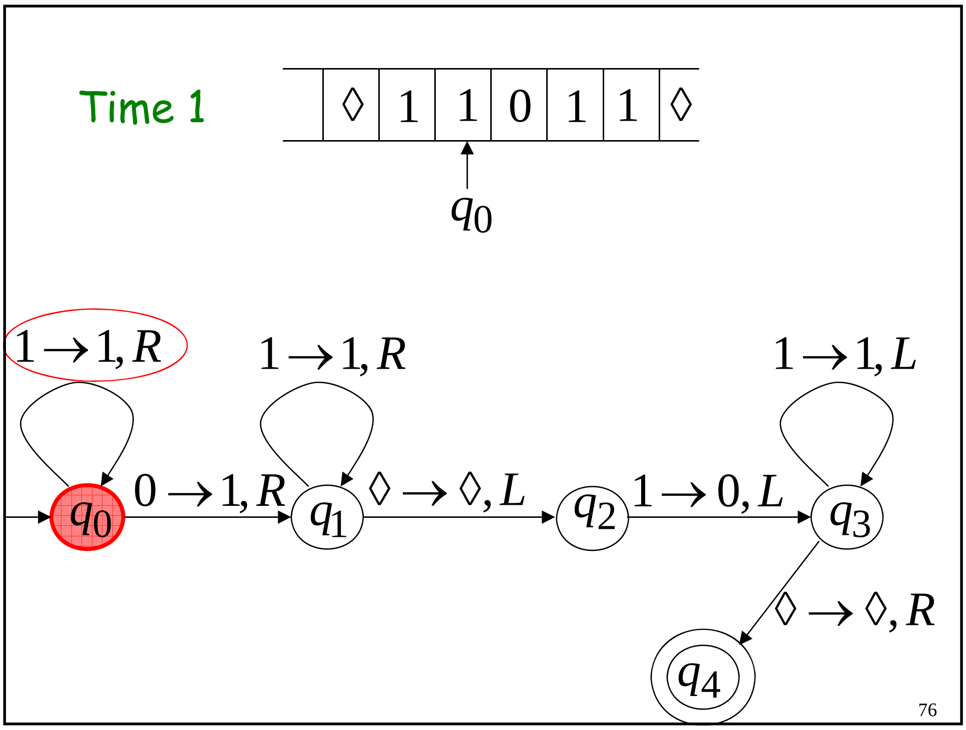
Time 0

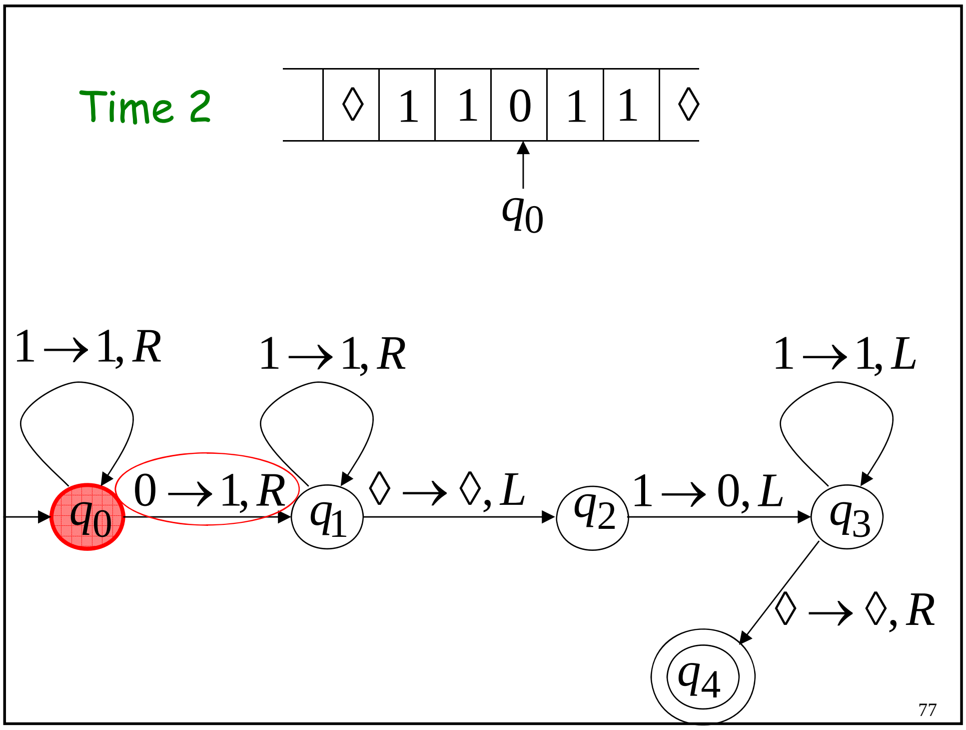


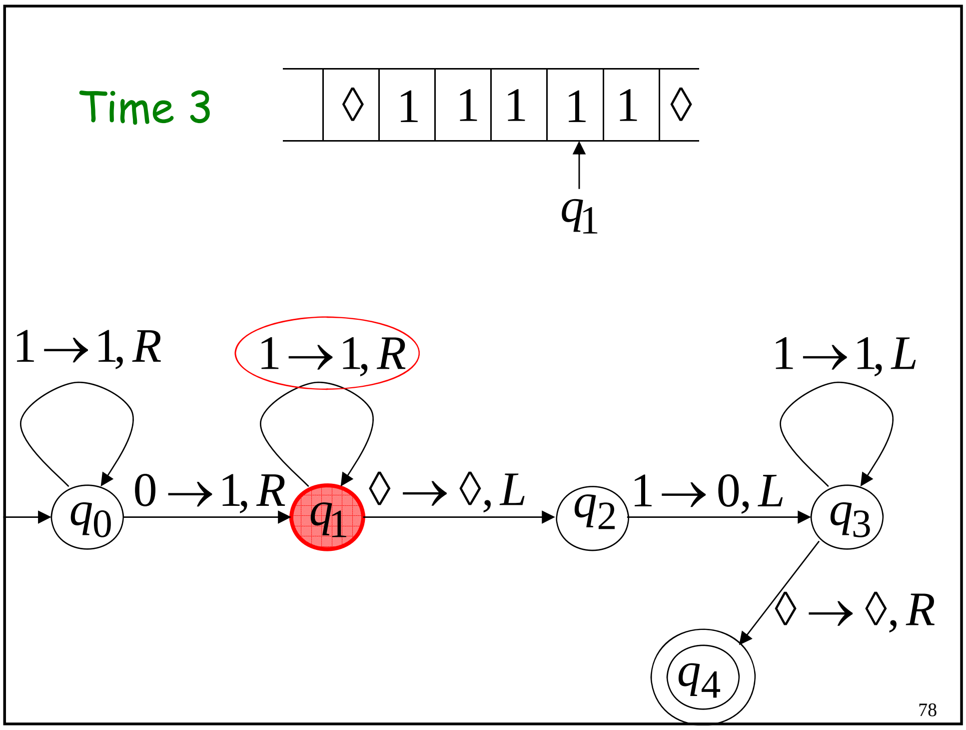
Final Result

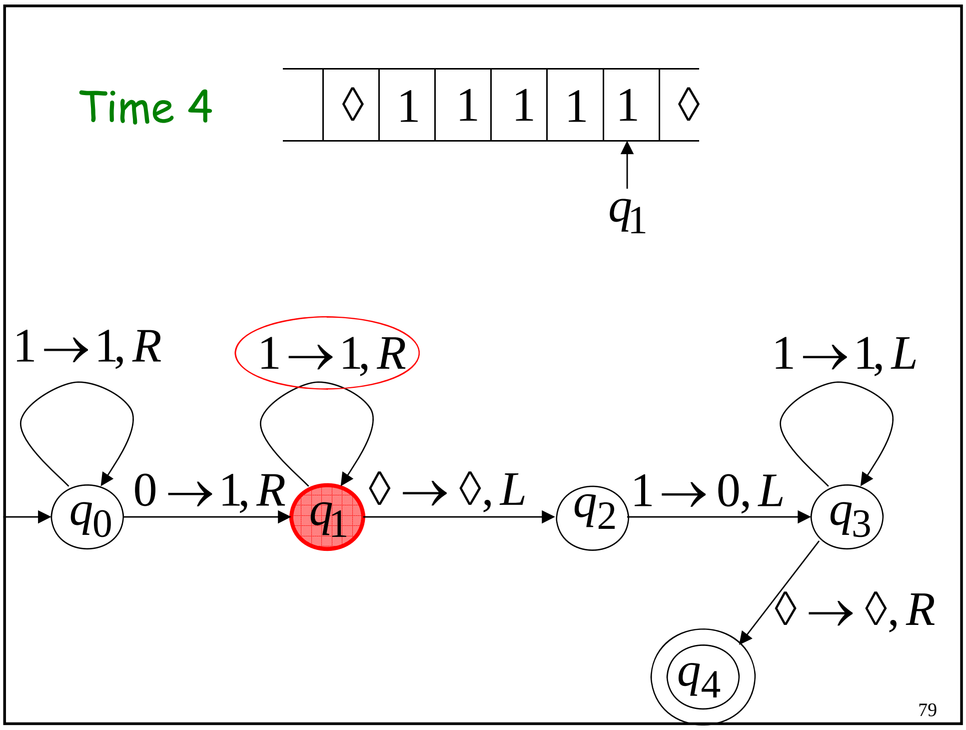


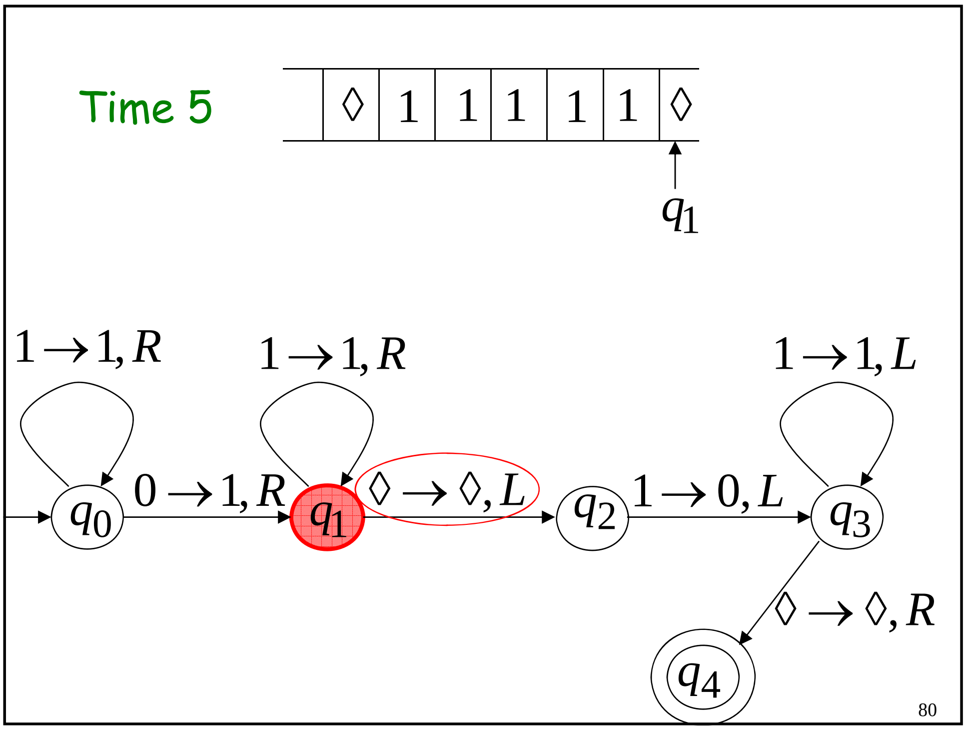

Time 0
◊ 1 1 0 1 1 ◊
q_0
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
$\lozenge \rightarrow \lozenge, R$
q_0
q_1
q_2
q_3
q_4
75

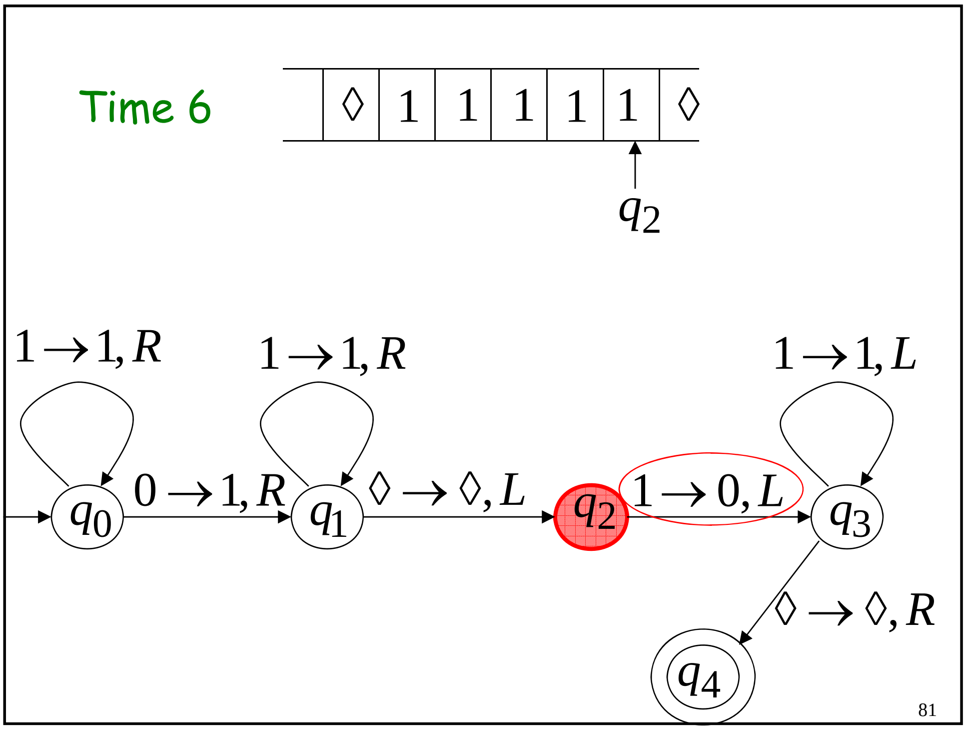

Time 1
◊ 1 1 0 1 1 ◊
q_0
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
$0 \rightarrow 1, R$
$\lozenge \rightarrow \lozenge, L$
$1 \rightarrow 0, L$
$\lozenge \rightarrow \lozenge, R$
q_0
q_1
q_2
q_3
q_4
76

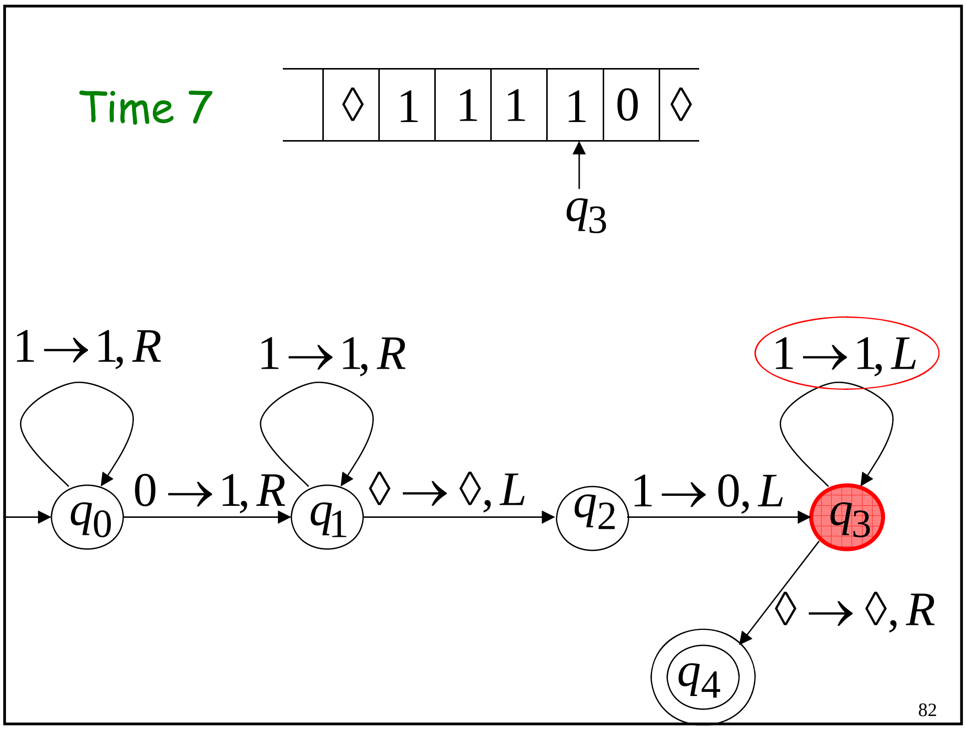

Time 2
◊ 1 1 0 1 1 ◊
q0
1 → 1, R
1 → 1, R
1 → 1, L
0 → 1, R
◊ → ◊, L
1 → 0, L
◊ → ◊, R
q0
q1
q2
q3
q4
77

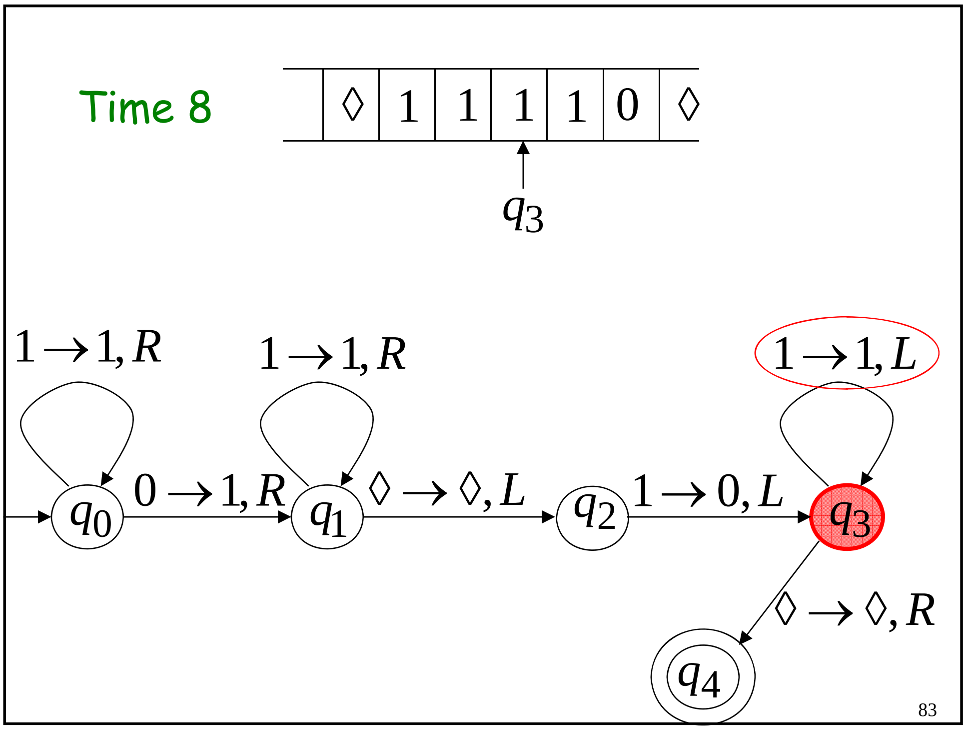

Time 3
◊ 1 1 1 1 1 ◊
q1
1 → 1, R
1 → 1, R
1 → 1, L
0 → 1, R
◊ → ◊, L
1 → 0, L
◊ → ◊, R
q0
q1
q2
q3
q4
78

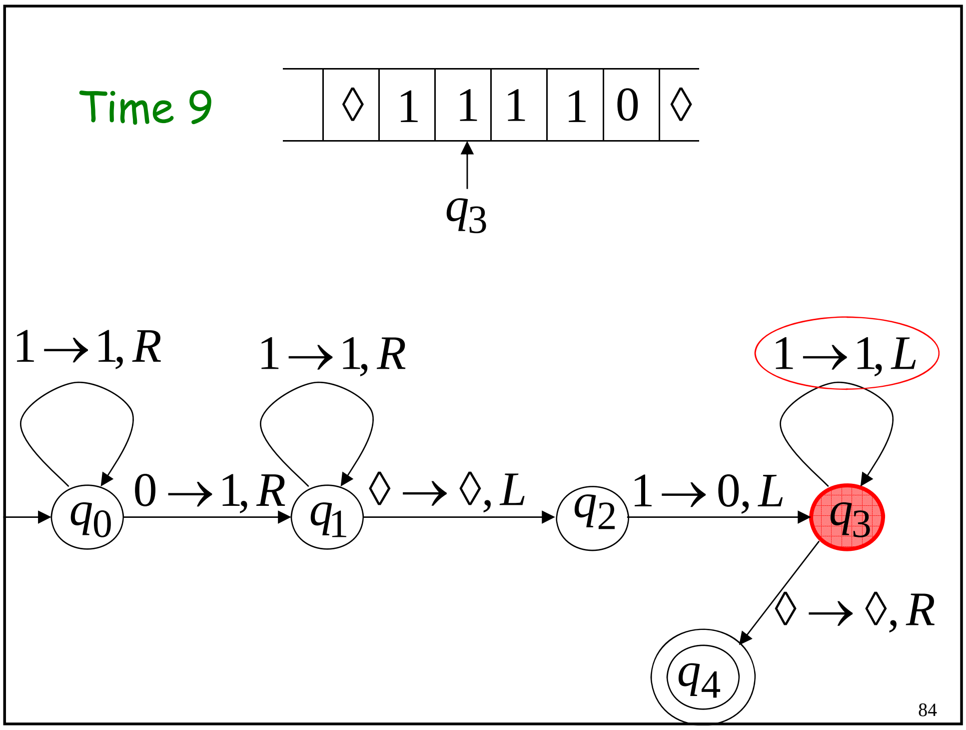

Time 4
◊
1
1
1
1
1
◊
q_1
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\Diamond \rightarrow \Diamond, L$
q_2
$1 \rightarrow 0, L$
q_3
$\Diamond \rightarrow \Diamond, R$
q_4
79

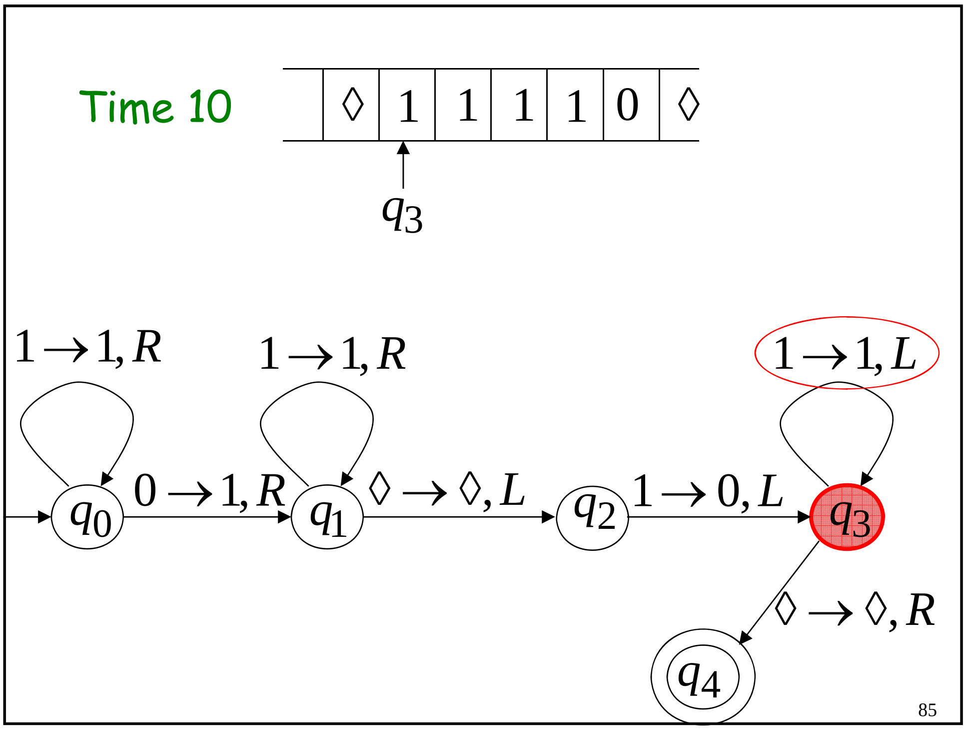

Time 5
◊
1
1
1
1
1
◊
q_1
$1 \rightarrow 1, R$
$1 \rightarrow 1, R$
$1 \rightarrow 1, L$
q_0
$0 \rightarrow 1, R$
q_1
$\Diamond \rightarrow \Diamond, L$
q_2
$1 \rightarrow 0, L$
q_3
$\Diamond \rightarrow \Diamond, R$
q_4
80

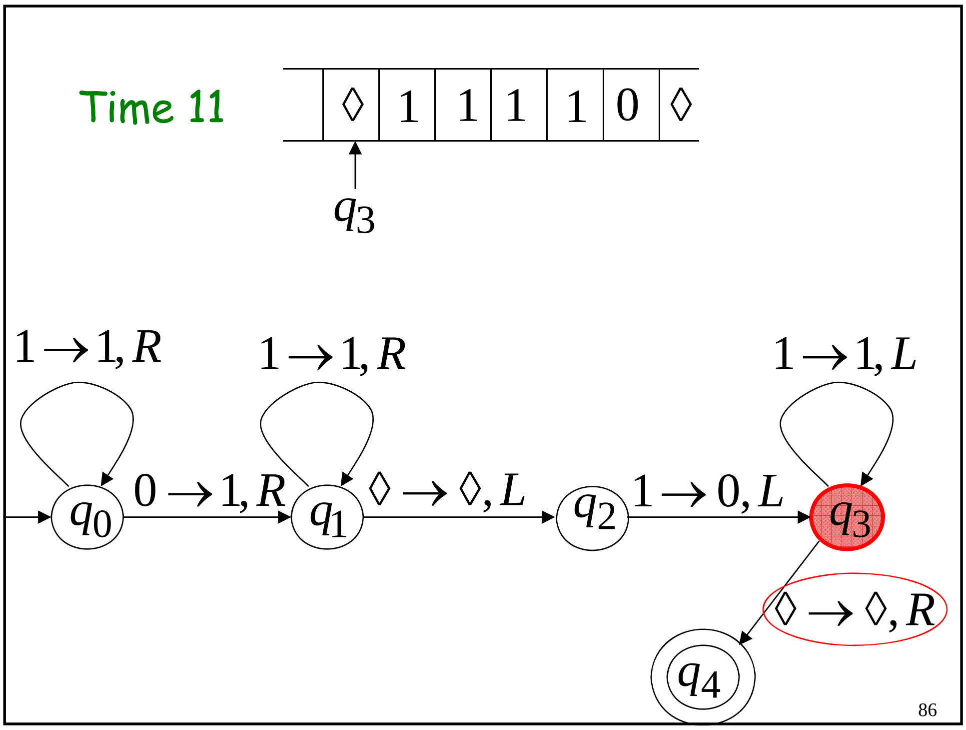

Time 6
◊ 1 1 1 1 1 ◊
q2
1 → 1, R
1 → 1, R
1 → 1, L
q0
0 → 1, R
q1
◊ → ◊, L
q2
1 → 0, L
q3
◊ → ◊, R
q4
81


Time 7
◊ 1 1 1 1 0 ◊
q3
1 → 1, R
1 → 1, R
1 → 1, L
q0
0 → 1, R
q1
◊ → ◊, L
q2
1 → 0, L
q3
◊ → ◊, R
q4
82

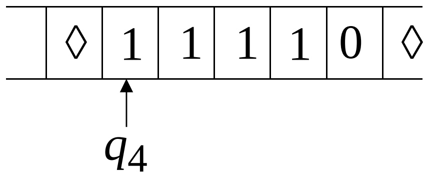

Time 8
◊ 1 1 1 1 0 ◊
q_3
$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
q_0
$0 \to 1, R$
q_1
$◊ \to ◊, L$
q_2
$1 \to 0, L$
q_3
$◊ \to ◊, R$
q_4
83

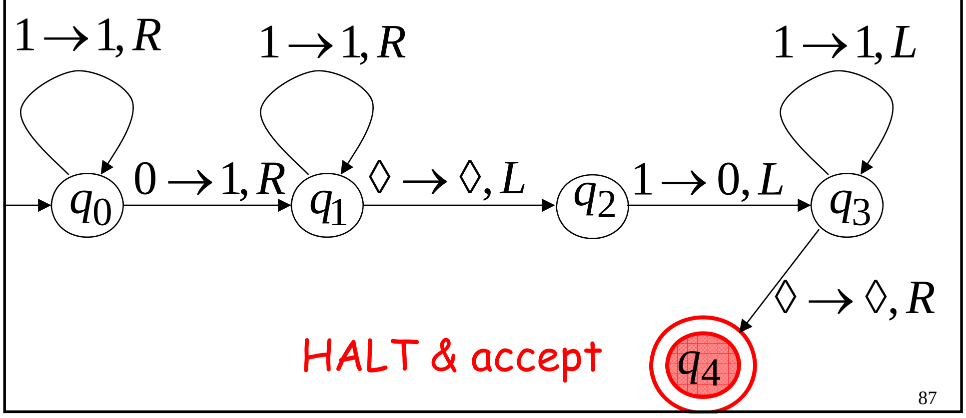

Time 9
◊ 1 1 1 1 0 ◊
q_3
$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
q_0
$0 \to 1, R$
q_1
$◊ \to ◊, L$
q_2
$1 \to 0, L$
q_3
$◊ \to ◊, R$
q_4
84


Time 10
◊ 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
85


Time 11
◊ 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 12





Another Example

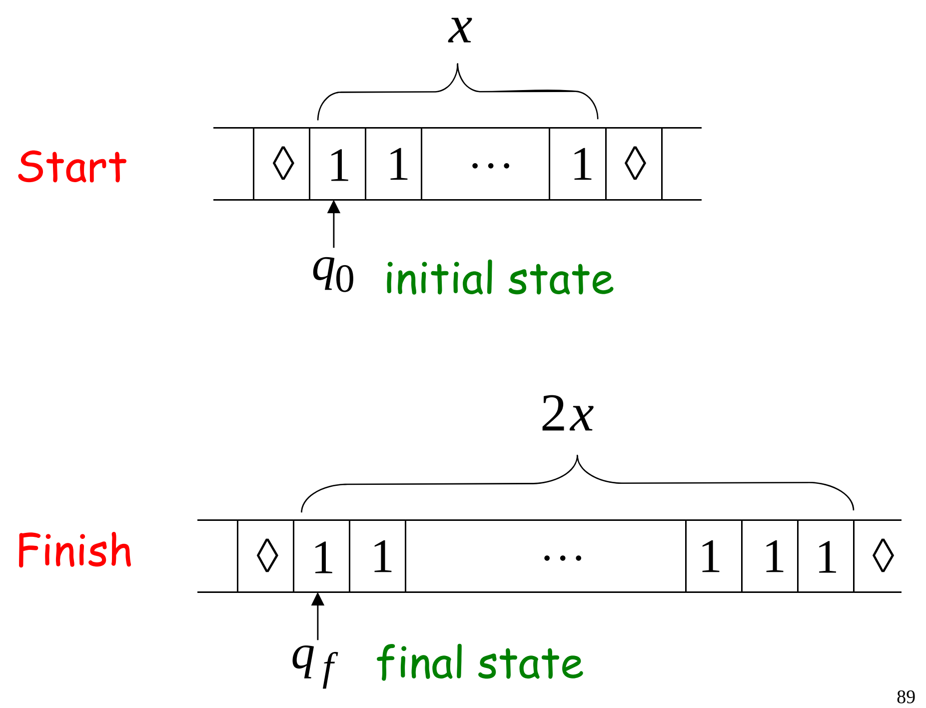
The function $f(x) = 2x$ is computable

x is integer

Turing Machine:

Input string: x unary

Output string: xx unary



89

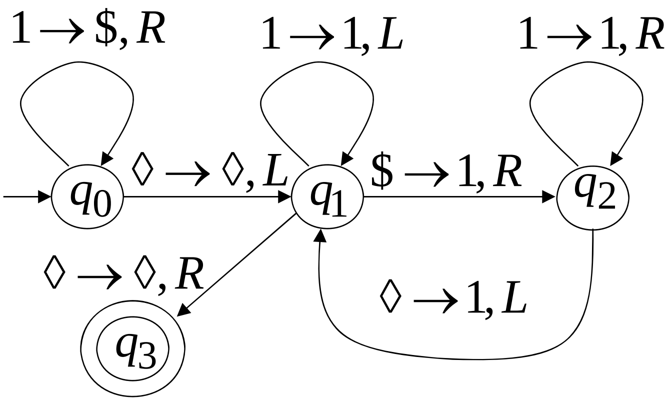
Turing Machine Pseudocode for $f(x) = 2x$

- Replace every 1 with $
- Repeat:
    - Find rightmost $, replace it with 1
    - Go to right end, insert 1

Until no more $ remain

90

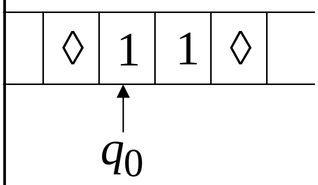
Turing Machine for $f(x) = 2x$



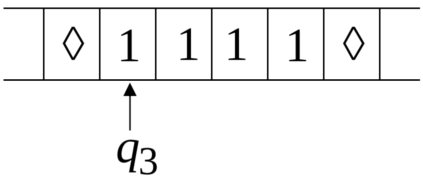
91

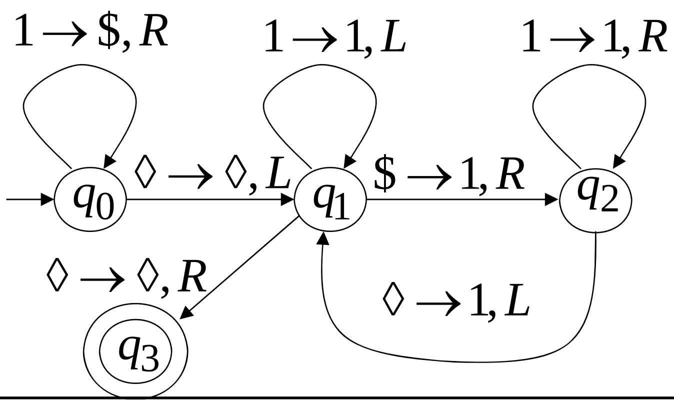
Example

Start



Finish





92

Another Example

The function is computable

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

Turing Machine for

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

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

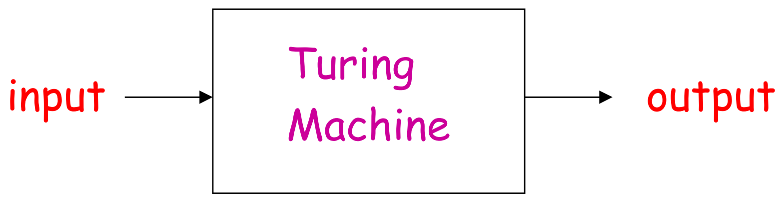
  erase tape, write 1 $(x > y)$

  else

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

Combining Turing Machines

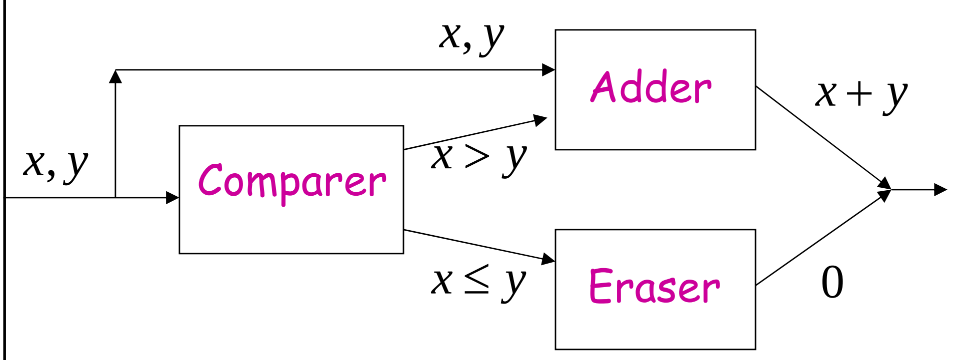
Block Diagram



97

Example:

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



98