Computing Functions with Turing Machines

A function $f(w)$ has:

Domain D $w \in D$

Result Region S $f(w) \in S$

Integer Domain:

Unary: 11111

Binary: 101

Decimal: 5

We prefer Unary representation:

Easier to manipulate

A function may have many parameters:

Example:

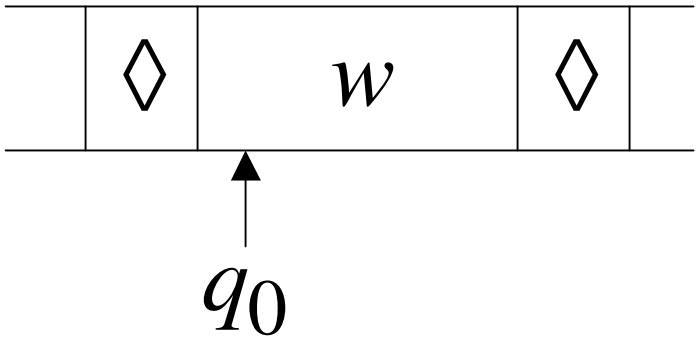
Addition function

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

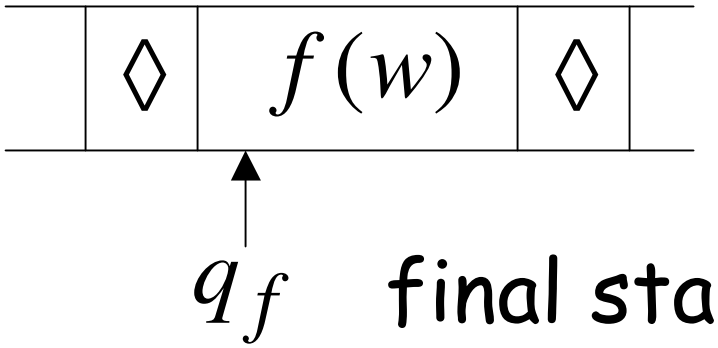
x, y are integers

Definition:

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



Initial Configuration



final state

Final configuration

$w \in D$ 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

$w \in D$ Domain

Example

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

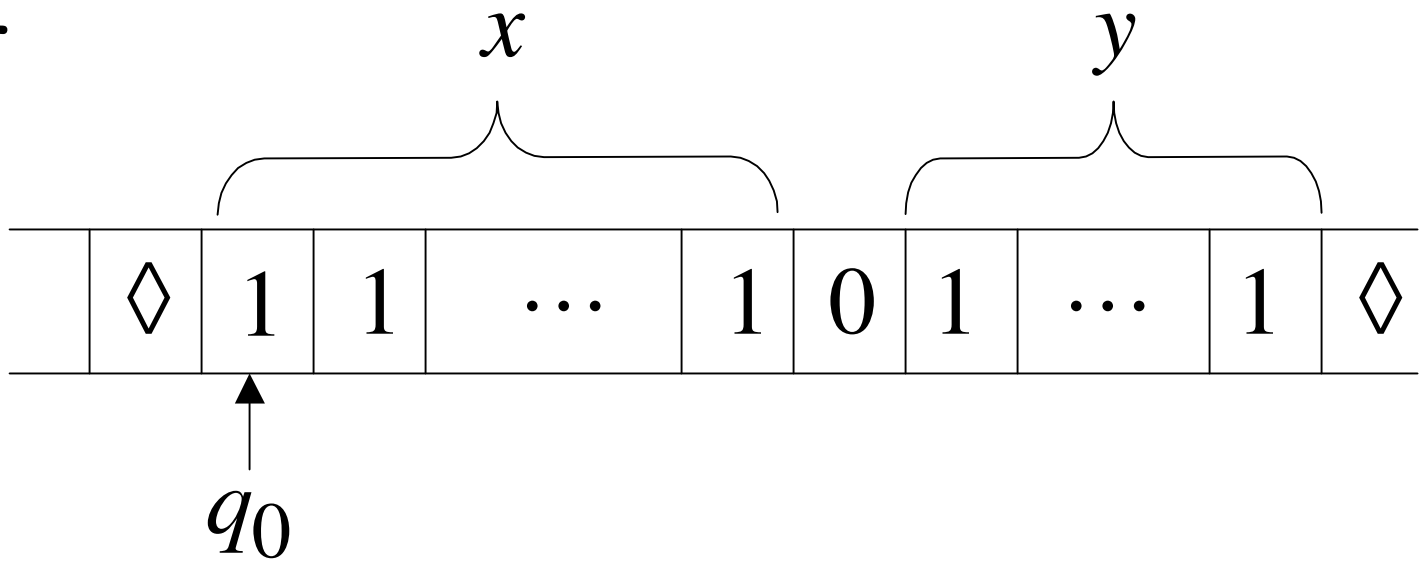
x, y are integers

Turing Machine:

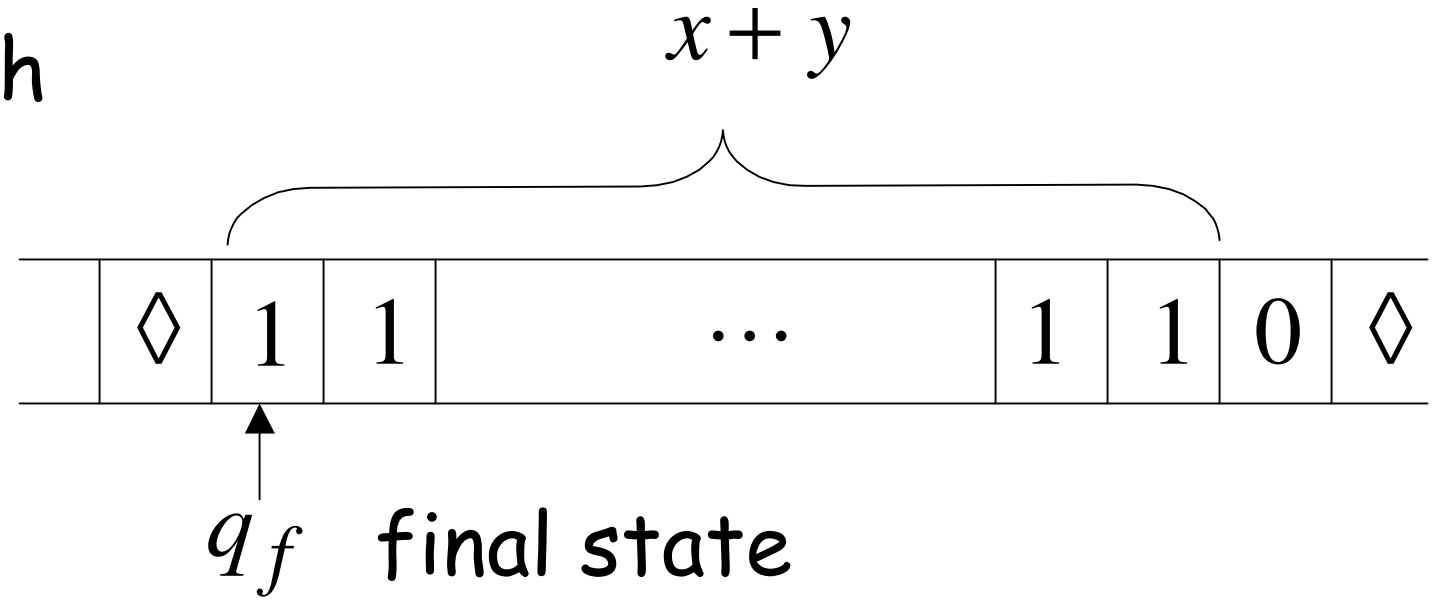
Input string: $x0y$ unary

Output string: $xy0$ unary

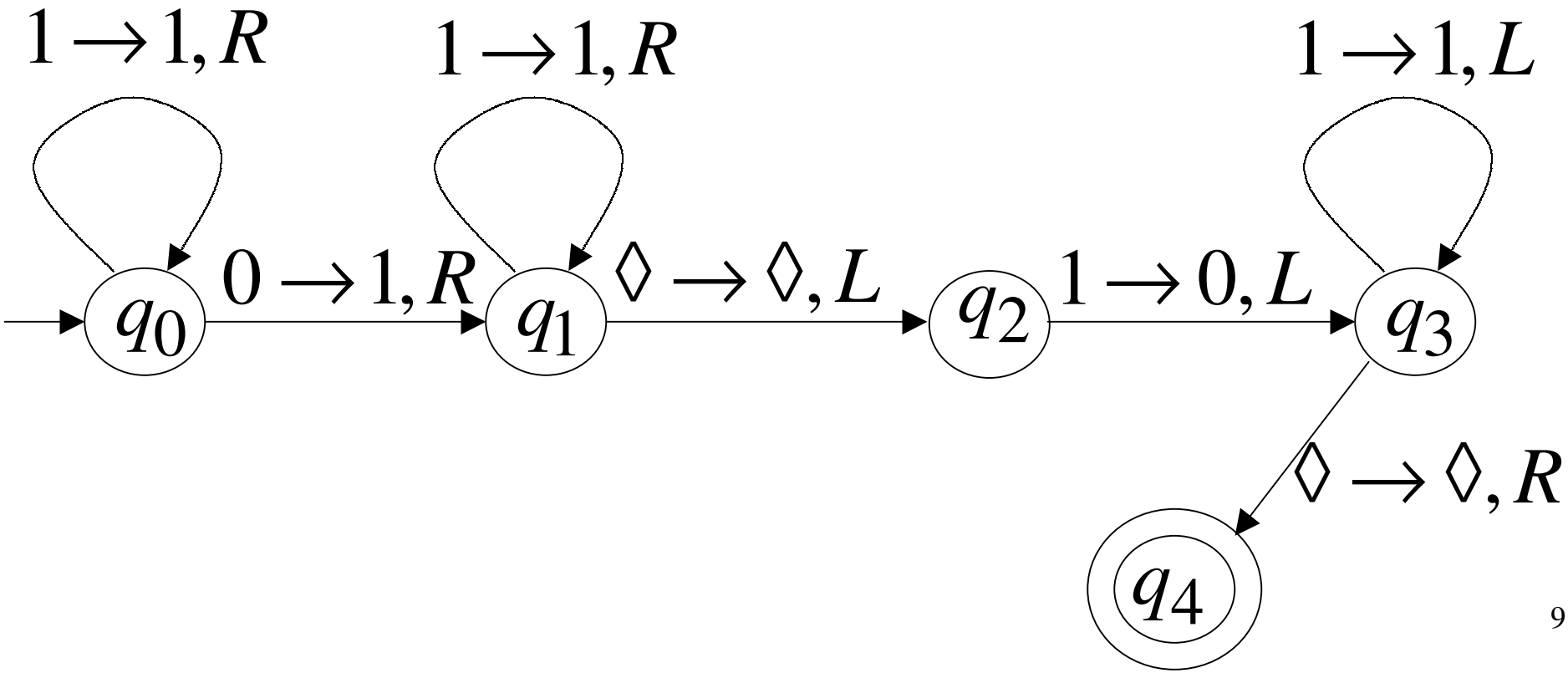
Start



Finish



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

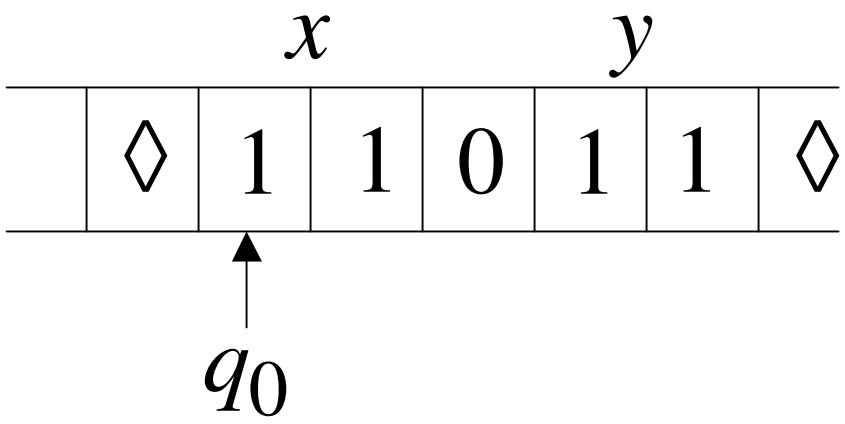


Execution Example:

$x = 11$ (2)

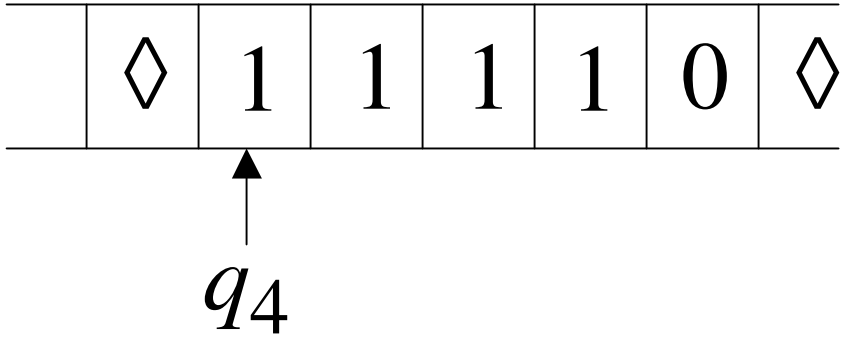
$y = 11$ (2)

Time 0

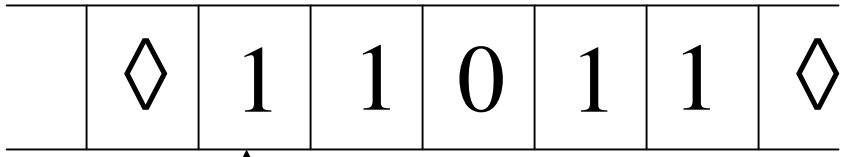


Final Result

$x + y$

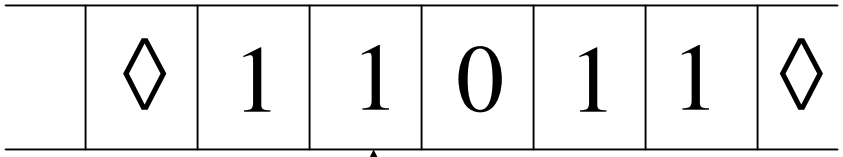


Time 0

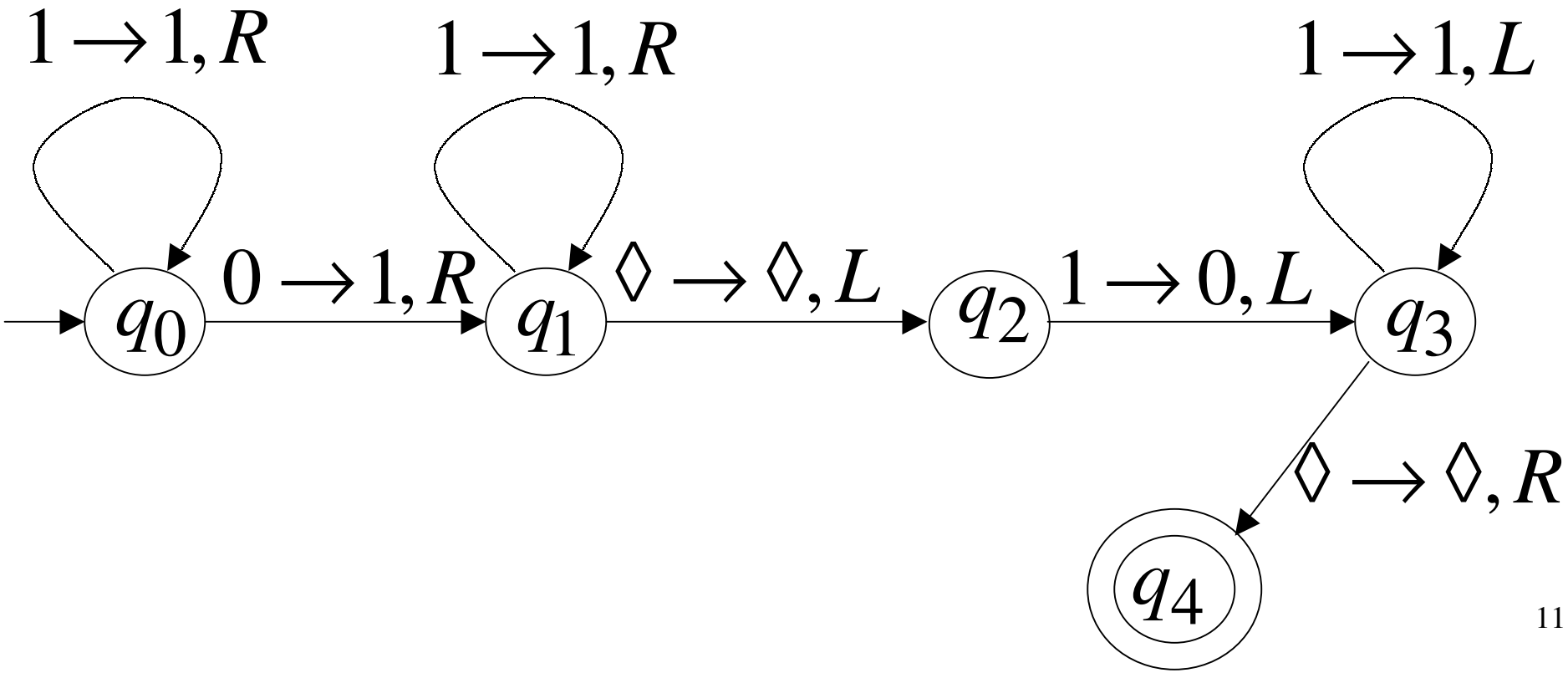


q_0

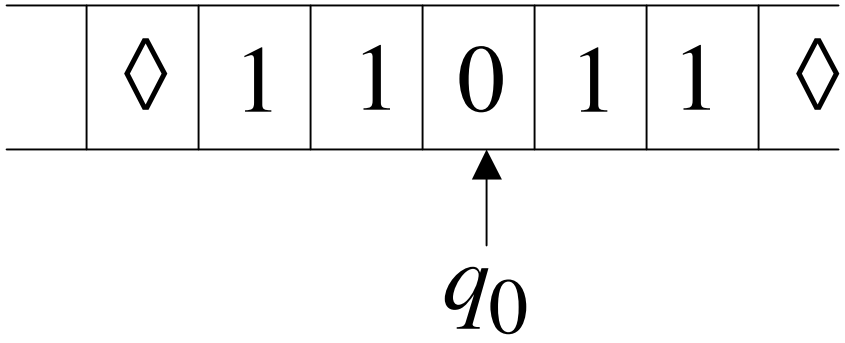
Time 1



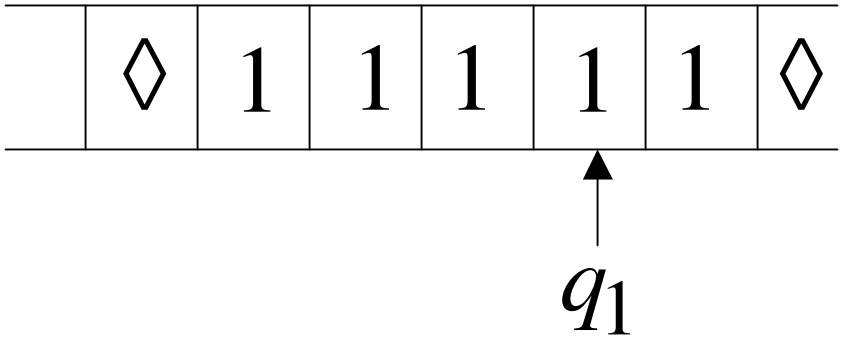
q_0

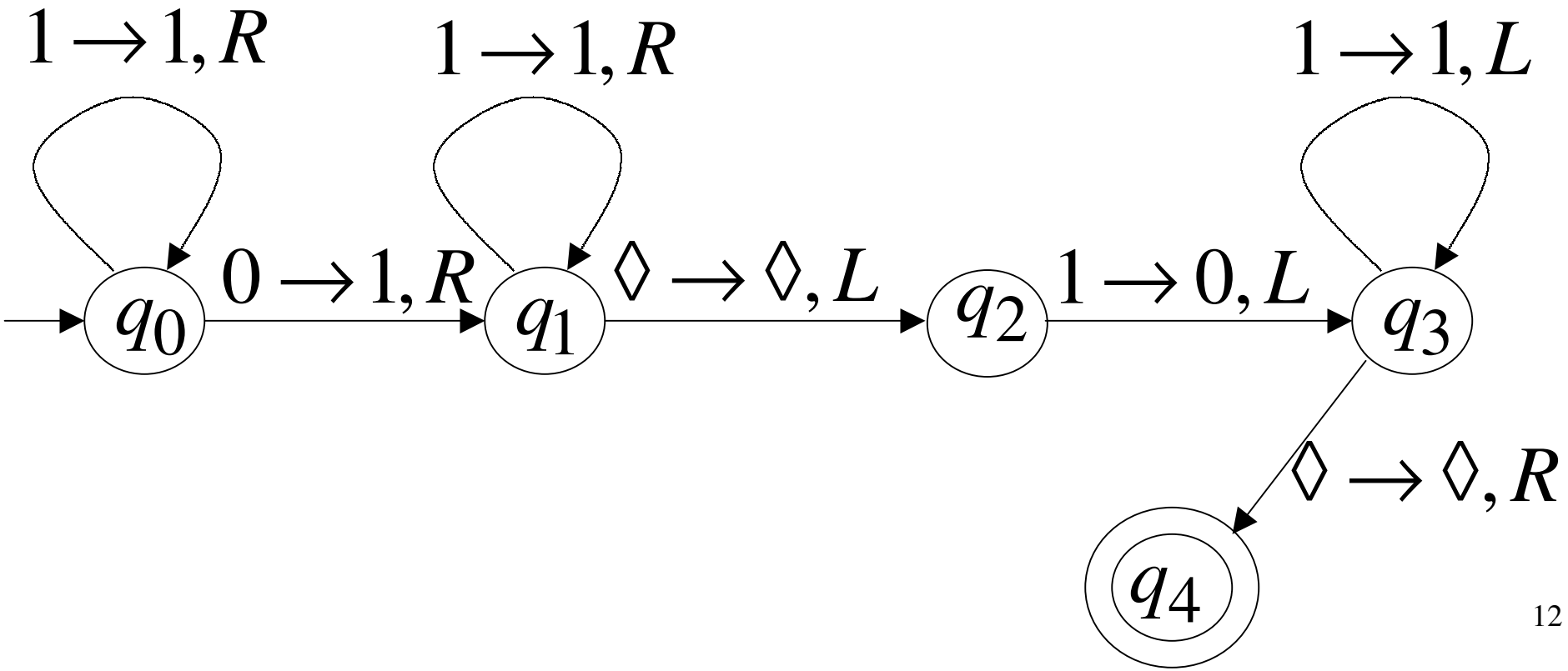


Time 2

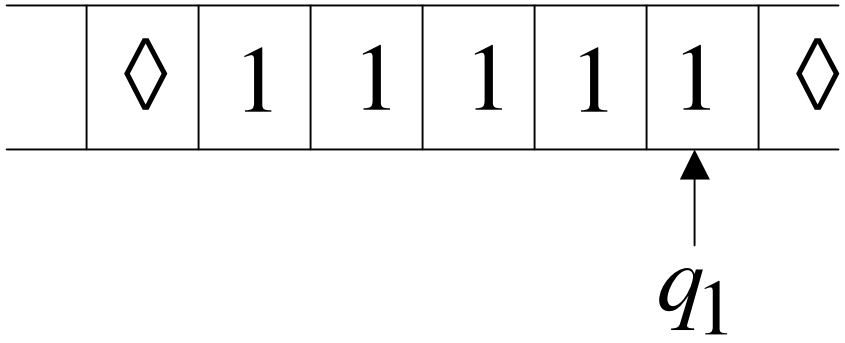

◊ 1 1 0 1 1 ◊
q_0

Time 3

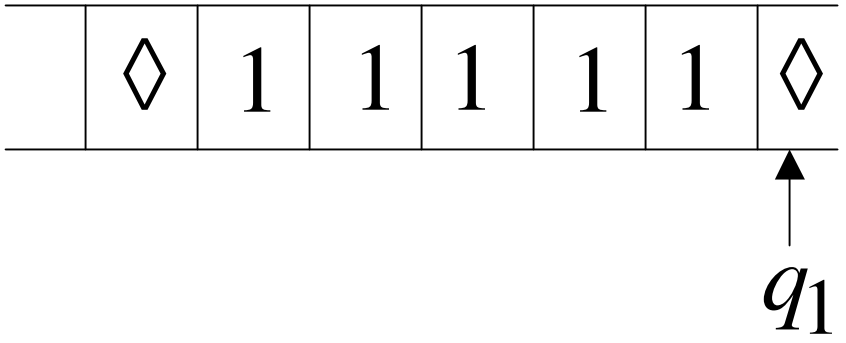

◊ 1 1 1 1 1 ◊
q_1

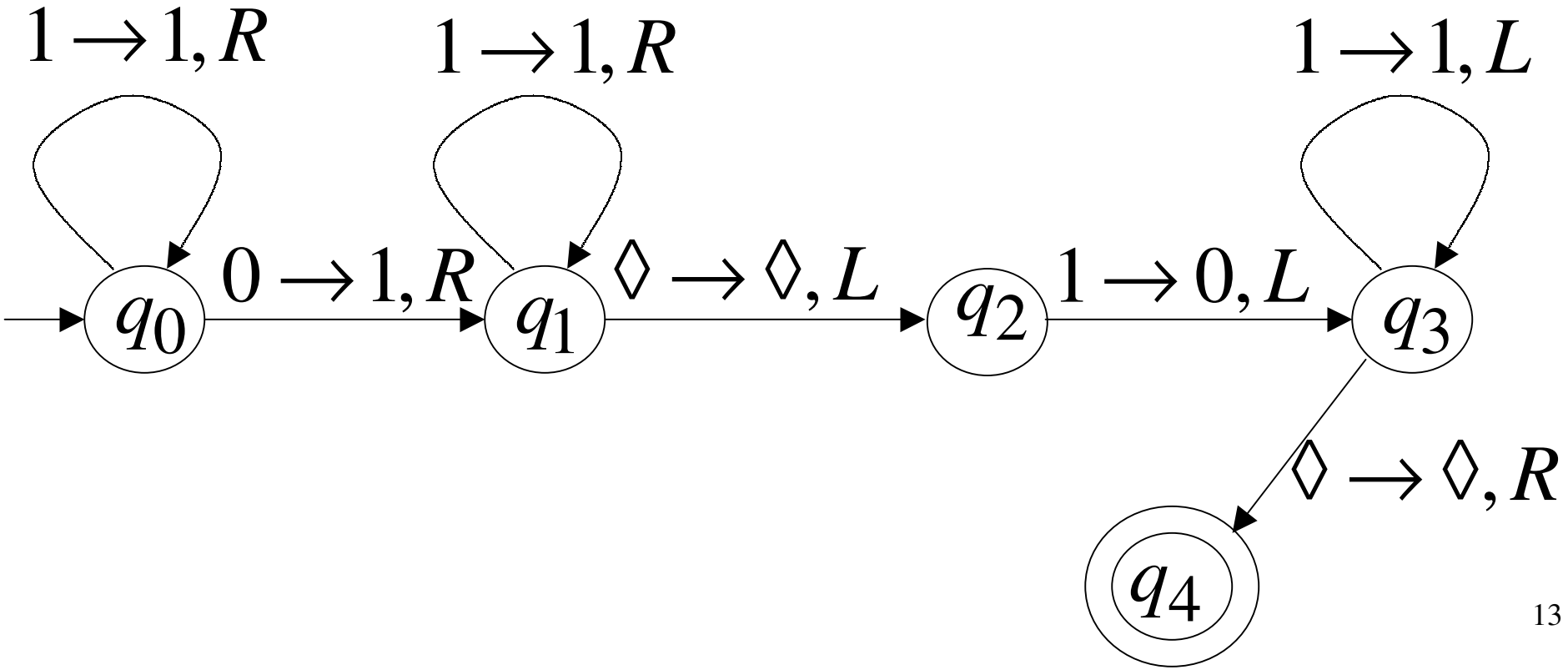

$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
$0 \to 1, R$
$\Diamond \to \Diamond, L$
$1 \to 0, L$
$\Diamond \to \Diamond, R$
q_0
q_1
q_2
q_3
q_4

Time 4

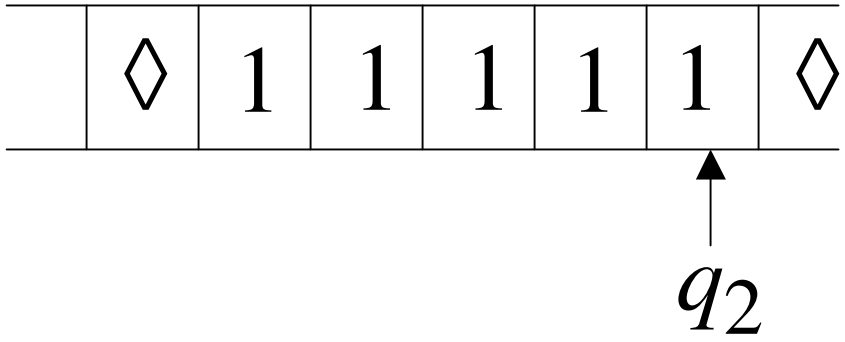

◊ 1 1 1 1 1 ◊
q_1

Time 5

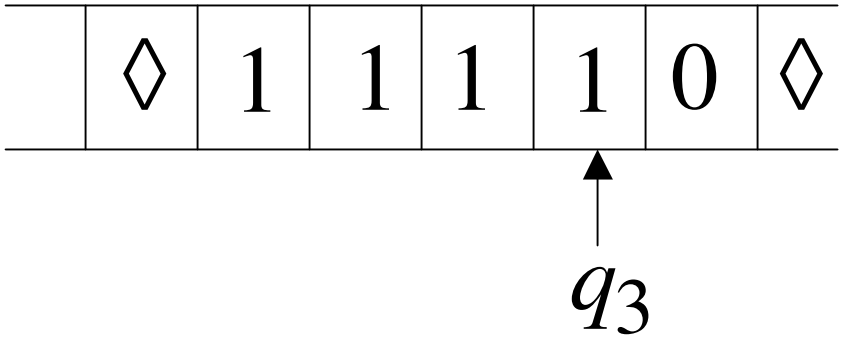

◊ 1 1 1 1 1 ◊
q_1

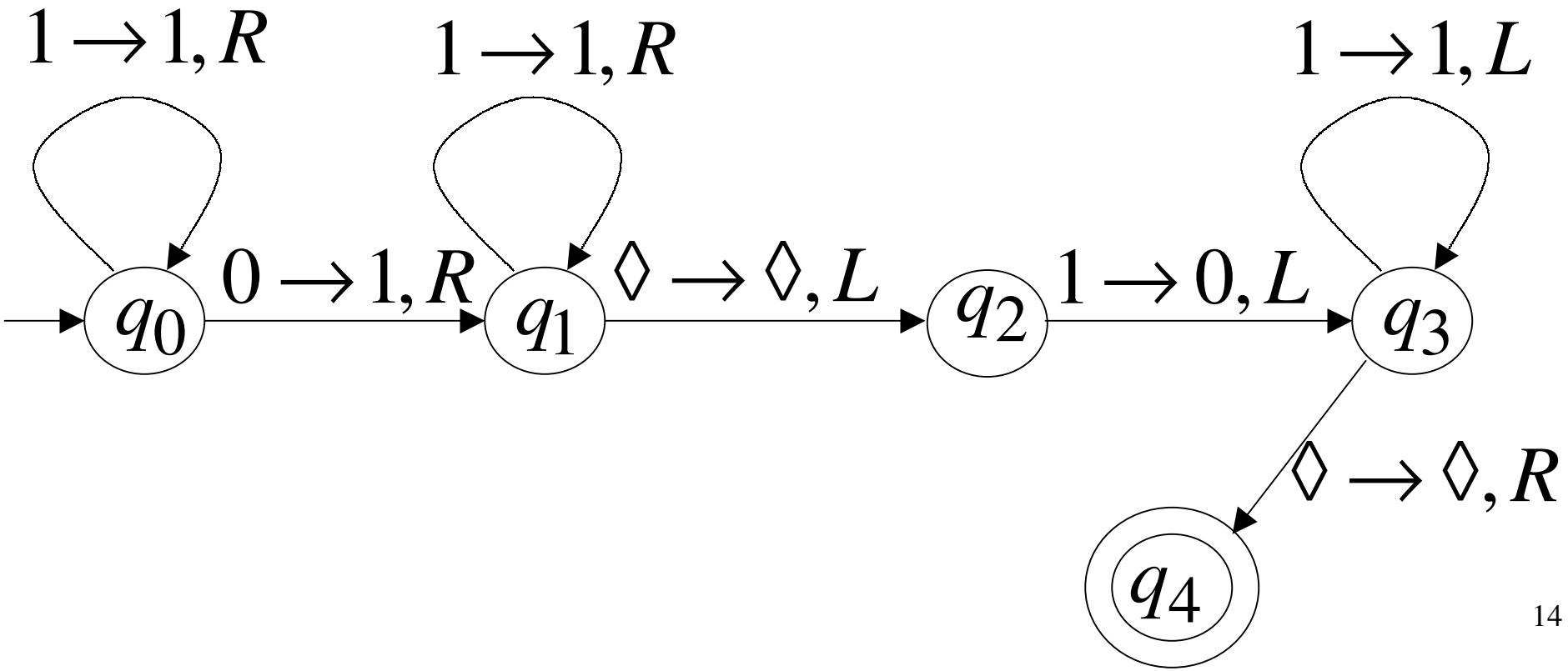

$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
$0 \to 1, R$
$\Diamond \to \Diamond, L$
$1 \to 0, L$
$\Diamond \to \Diamond, R$
q_0
q_1
q_2
q_3
q_4

Time 6

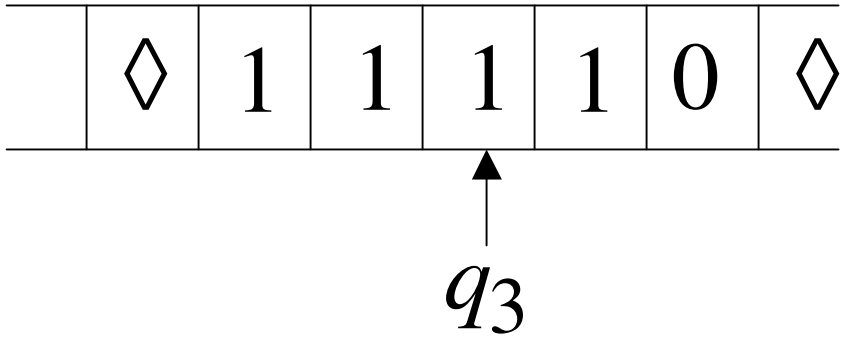

◊ 1 1 1 1 1 ◊
q_2

Time 7

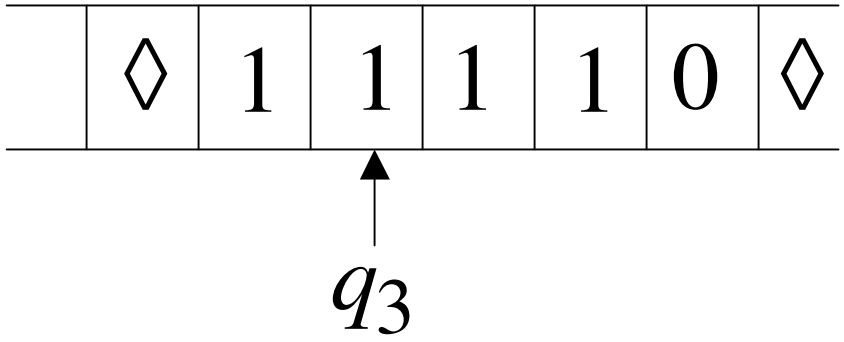

◊ 1 1 1 1 0 ◊
q_3

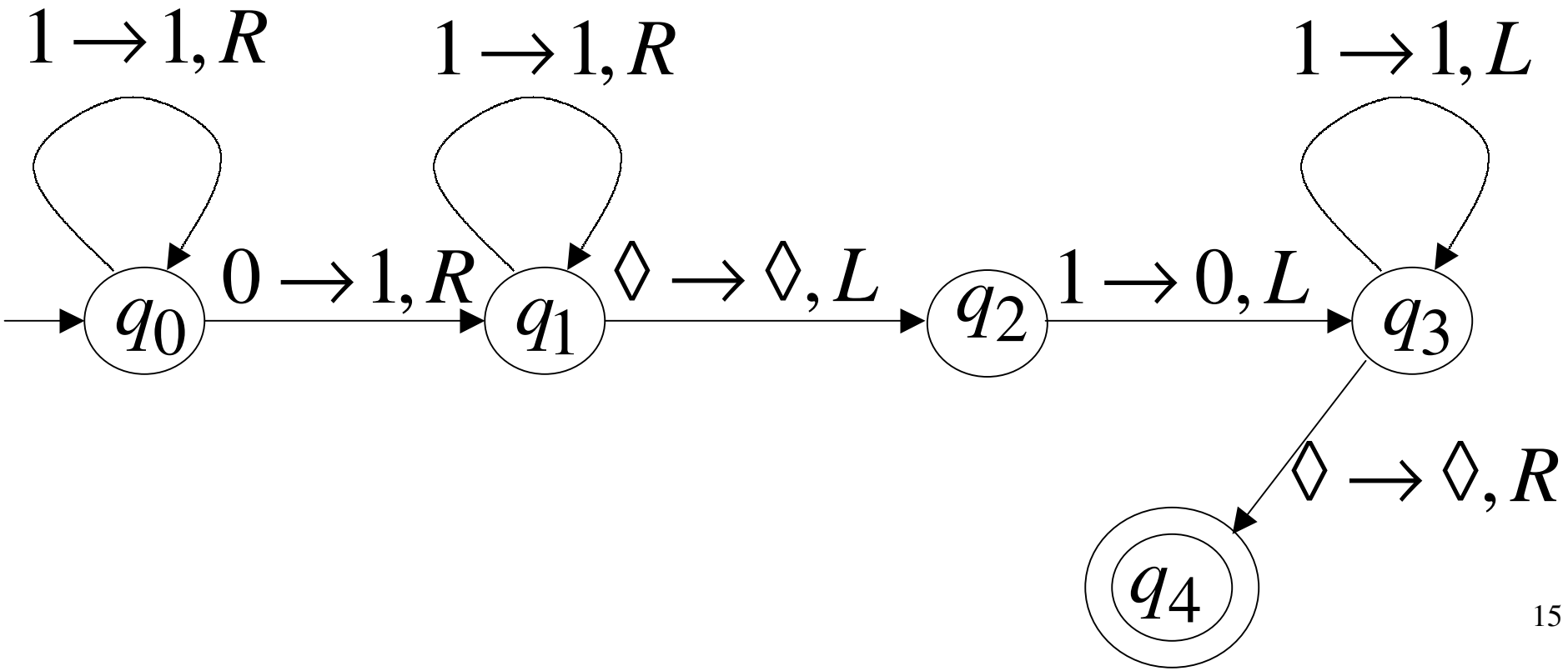

$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
$0 \to 1, R$
$\Diamond \to \Diamond, L$
$1 \to 0, L$
$\Diamond \to \Diamond, R$
q_0
q_1
q_2
q_3
q_4

Time 8

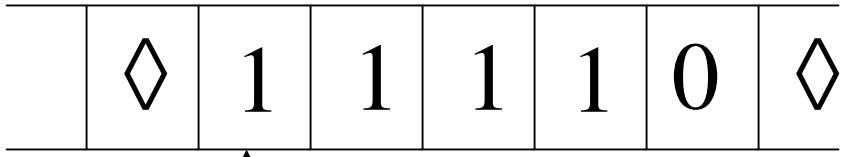

◊ 1 1 1 1 0 ◊
q_3

Time 9


◊ 1 1 1 1 0 ◊
q_3

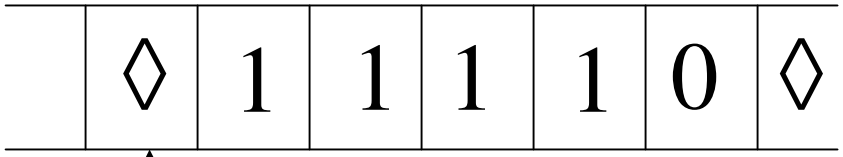

$1 \to 1, R$
$1 \to 1, R$
$1 \to 1, L$
$0 \to 1, R$
$\Diamond \to \Diamond, L$
$1 \to 0, L$
$\Diamond \to \Diamond, R$
q_0
q_1
q_2
q_3
q_4

Time 10

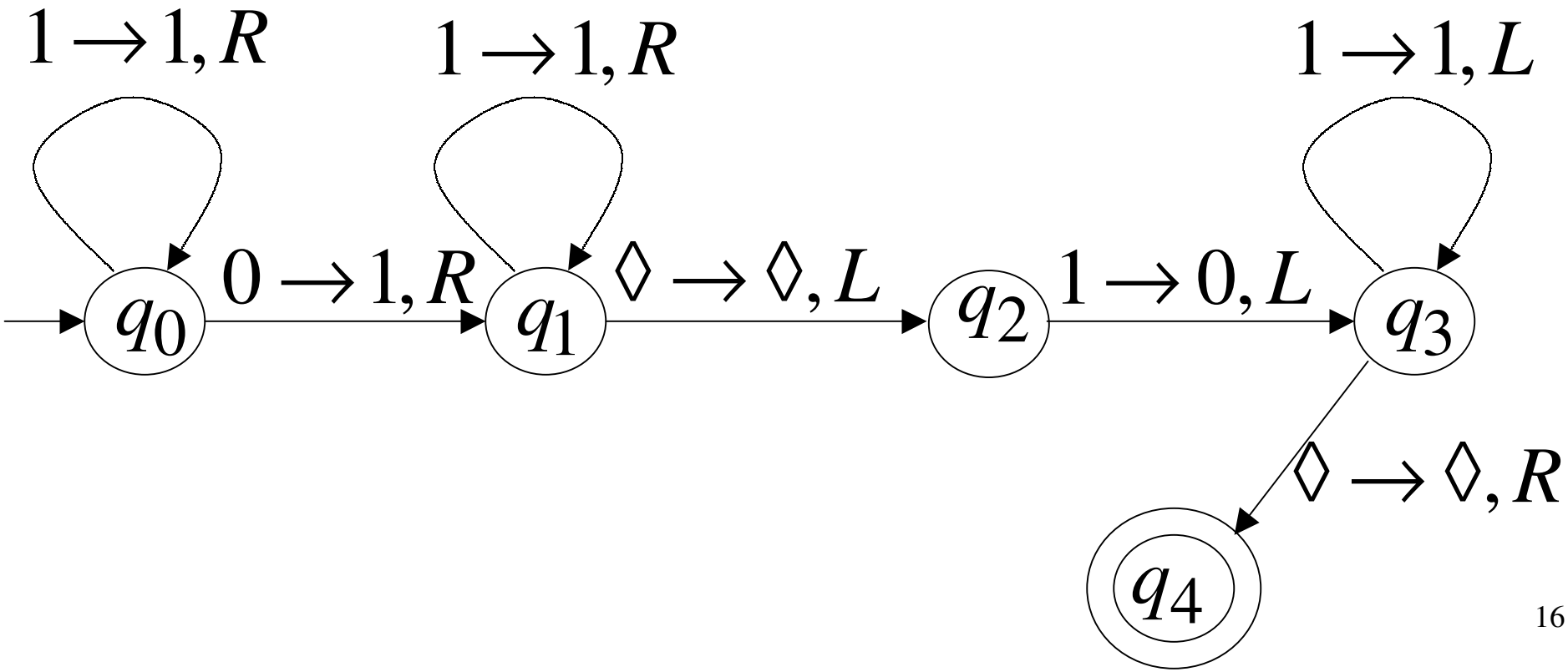


q_3

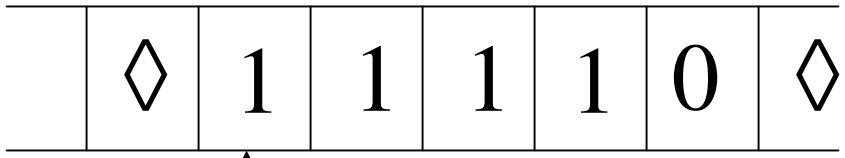
Time 11



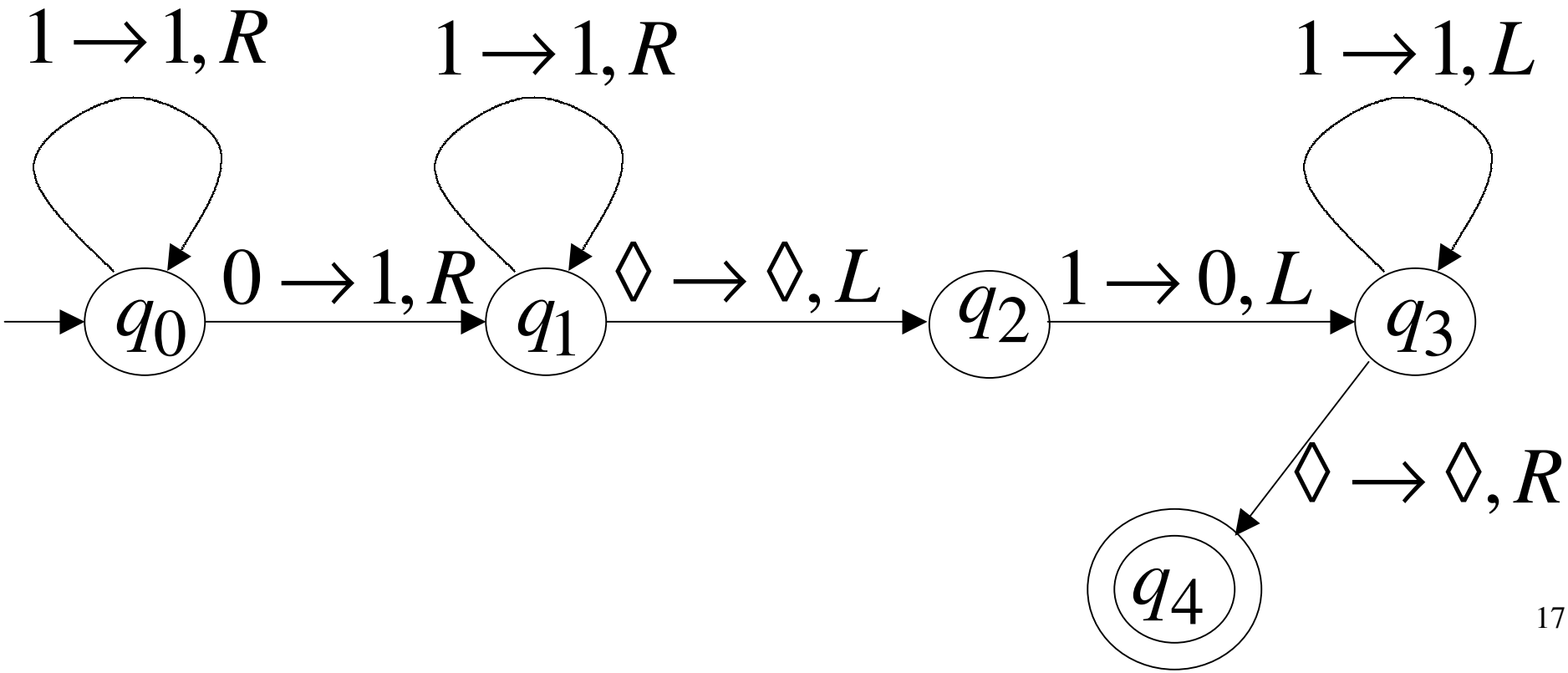
q_3



Time 12



q_4 HALT & accept



Another Example

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

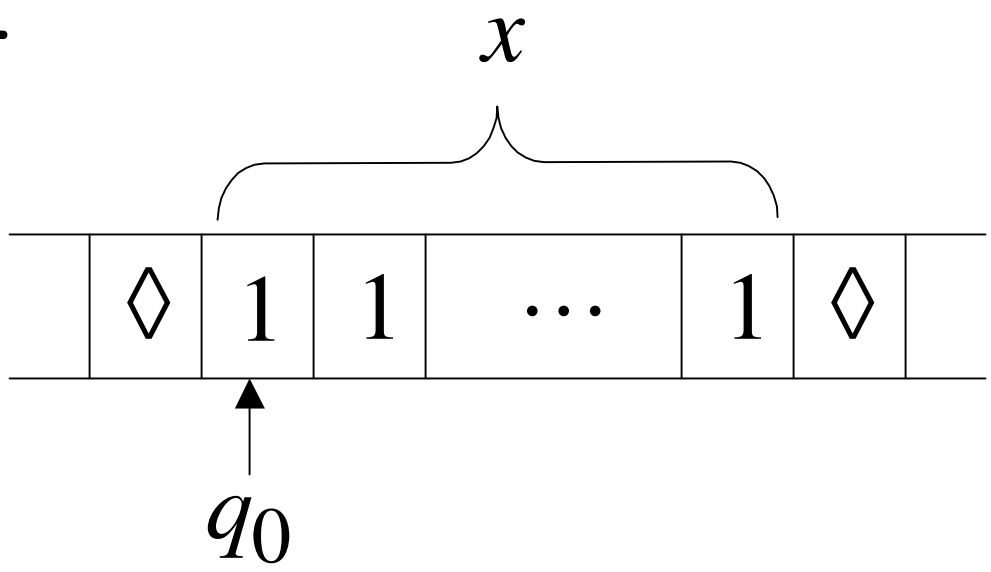
x is integer

Turing Machine:

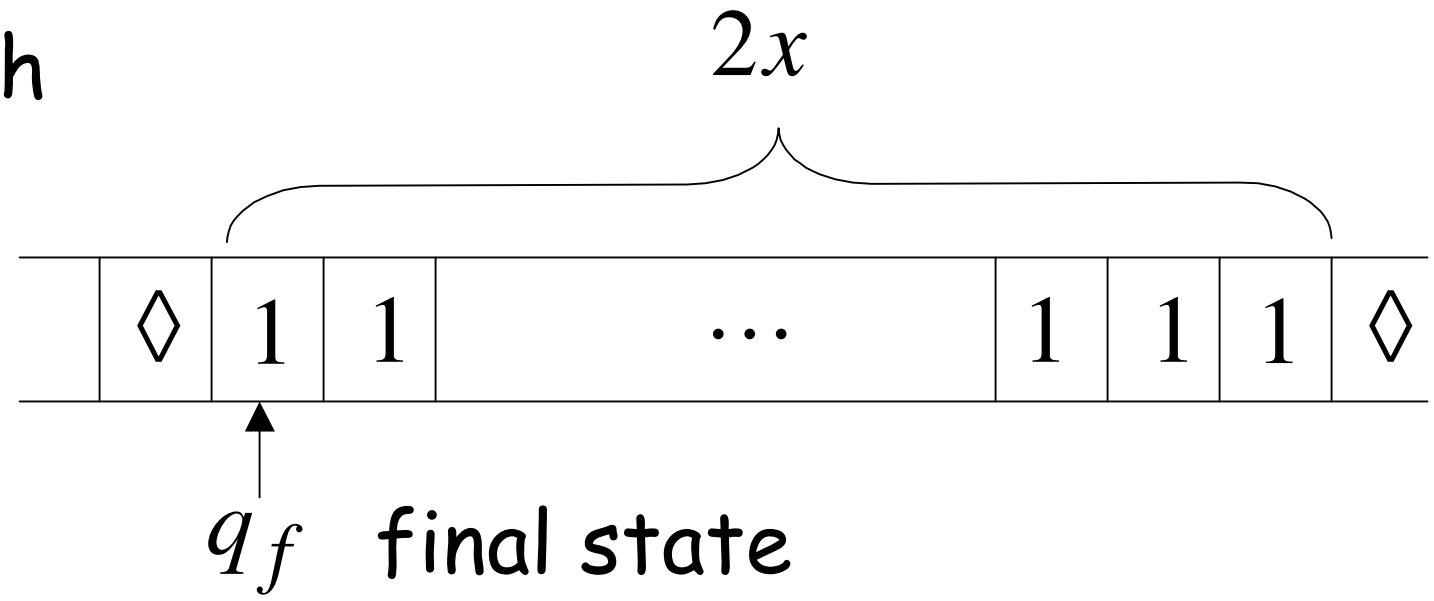
Input string: x unary

Output string: xx unary

Start



Finish



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

1. Replace every 1 with $

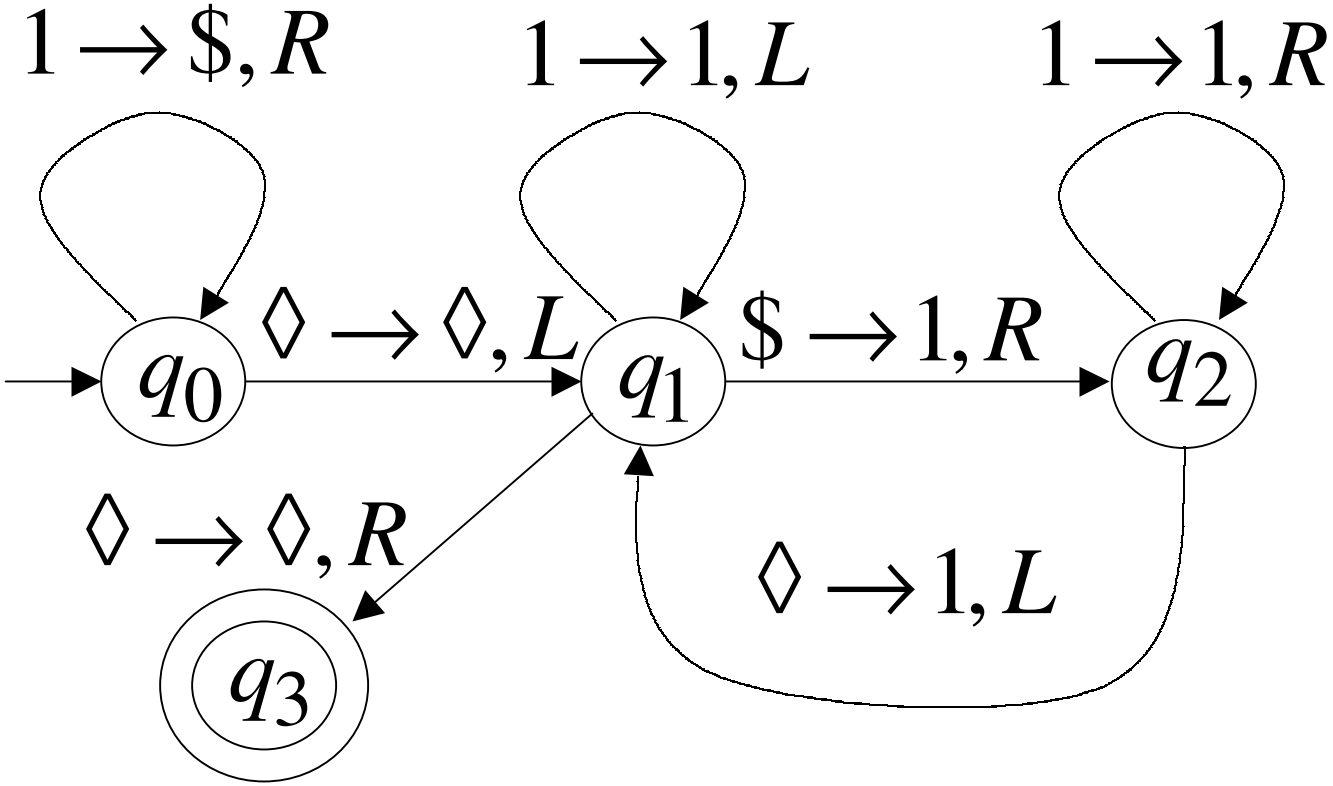
Repeat:

2. Find rightmost $, replace it with 1

3. Go to right end, insert 1

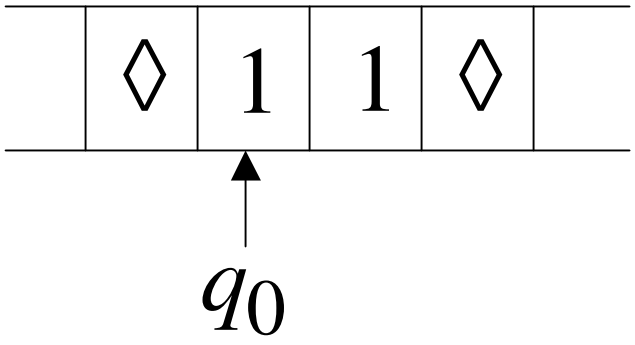
Until no more $ remain

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

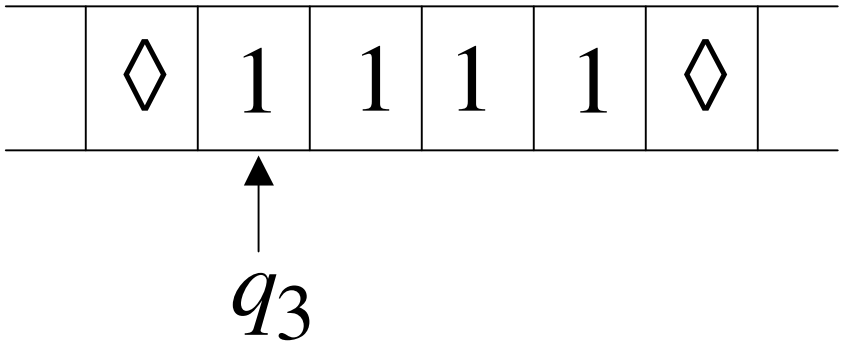


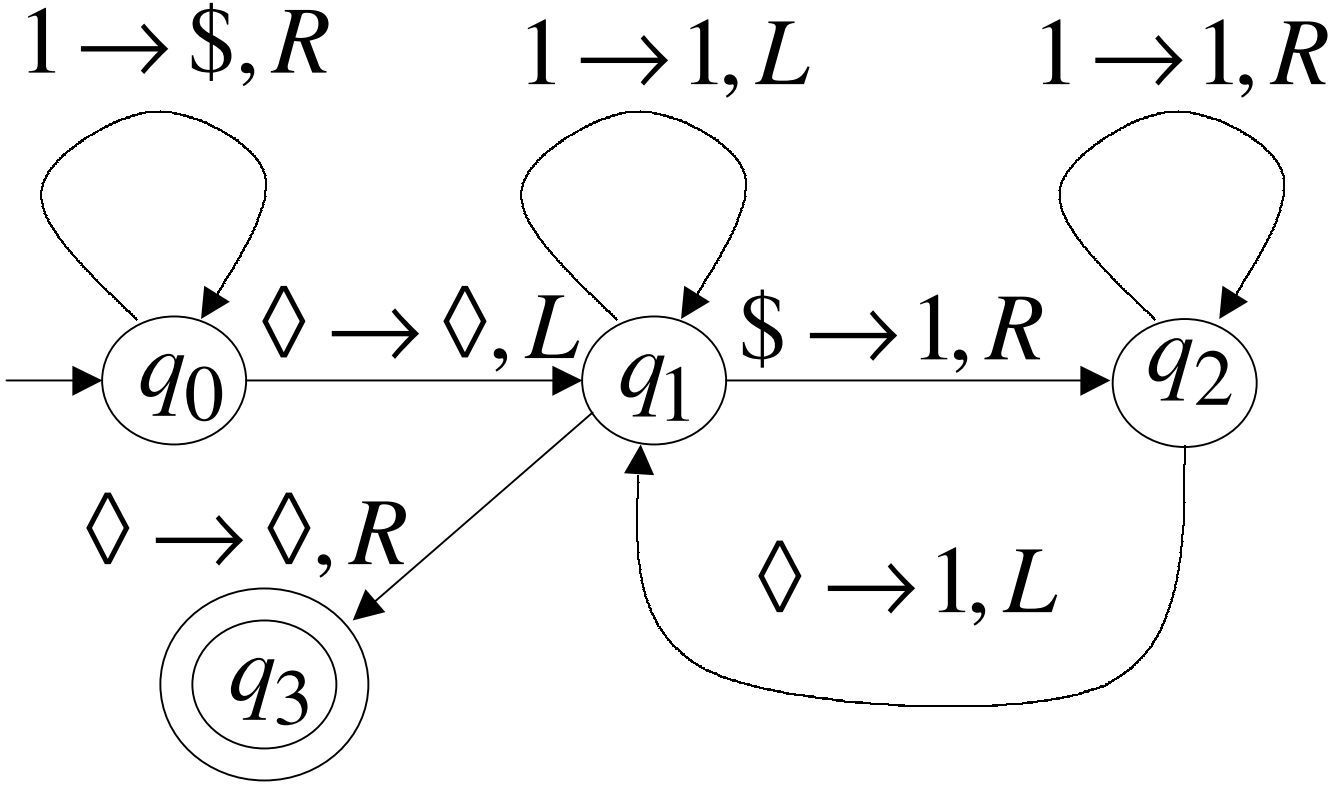
Example

Start



Finish





Another Example

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

is computable

Turing Machine for

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

Input: $x0y$

Output: 1 or 0

Turing Machine Pseudocode:

1. Repeat

   Match a 1 from x with a 1 from y

   Until all x or y has been matched

2. If a 1 from x is not matched

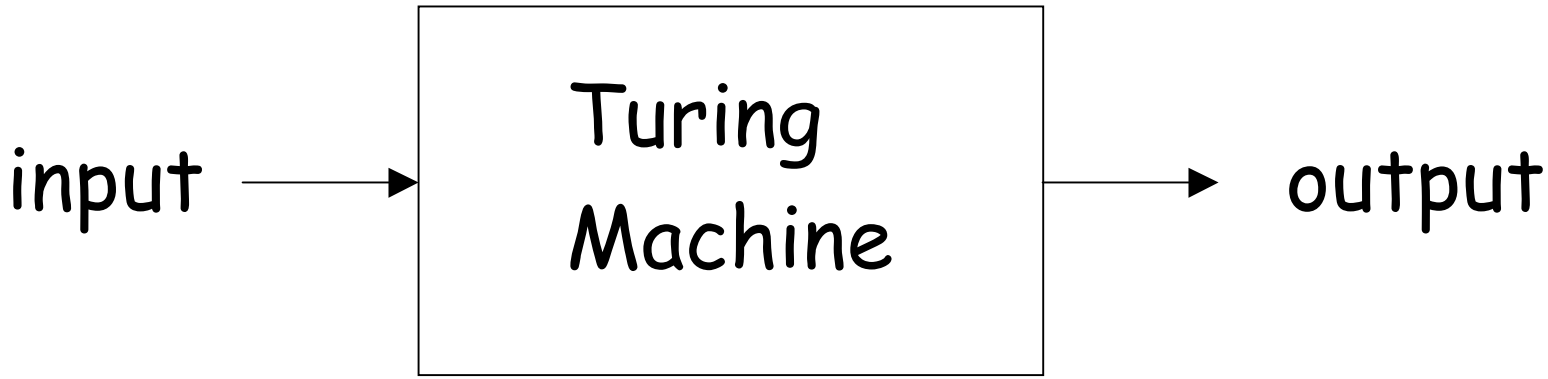
   erase tape, write 1

   else

   erase tape, write 0

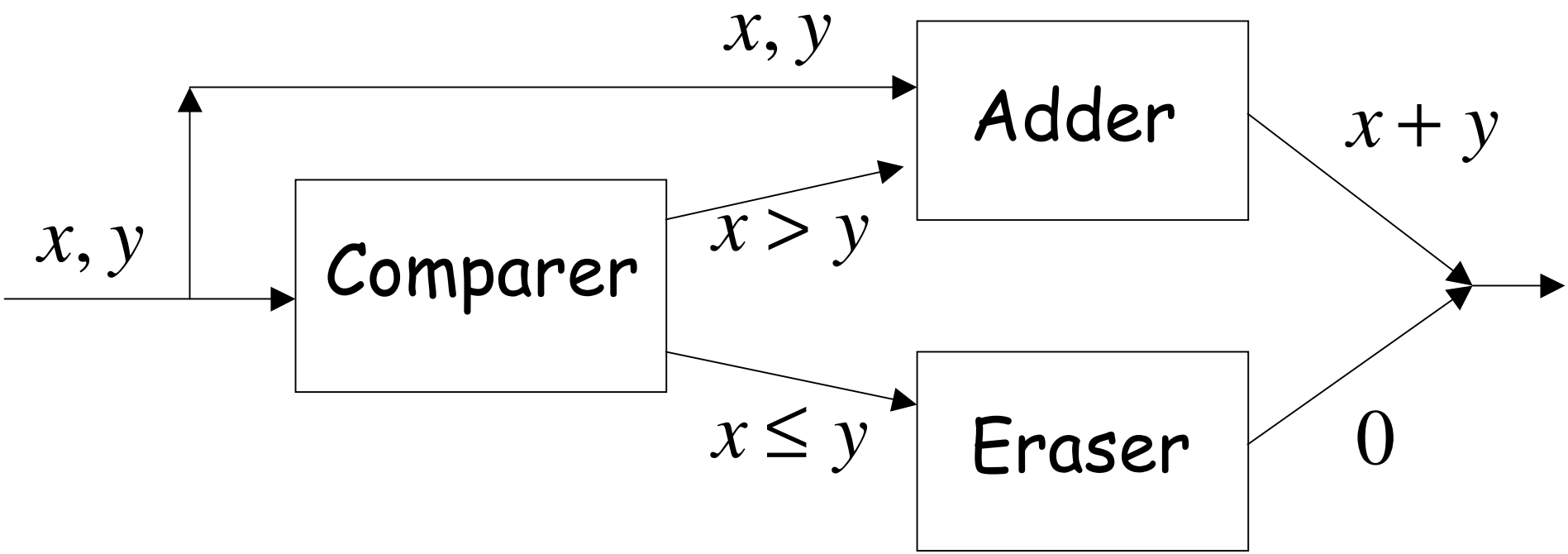
Combining Turing Machines

Block Diagram



Example:

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



Turing's Thesis

Do Turing machines have the same power with a digital computer?

Do Turing machines have the same power with a digital computer?

Intuitive answer: Yes

There is no formal answer

Turing's thesis:

Any computation carried out
by mechanical means
can be performed by Turing Machine

(1930)

Computer Science Law:

A computation is mechanical
if and only if
it can be performed by a Turing Machine

There is no known model of computation
more powerful than Turing Machines

Definition of Algorithm:

An algorithm for function $f(w)$

is a

Turing Machine which computes $f(w)$

Algorithms are Turing Machines

When we say:

There exists an algorithm

We mean:

There exists a Turing Machine