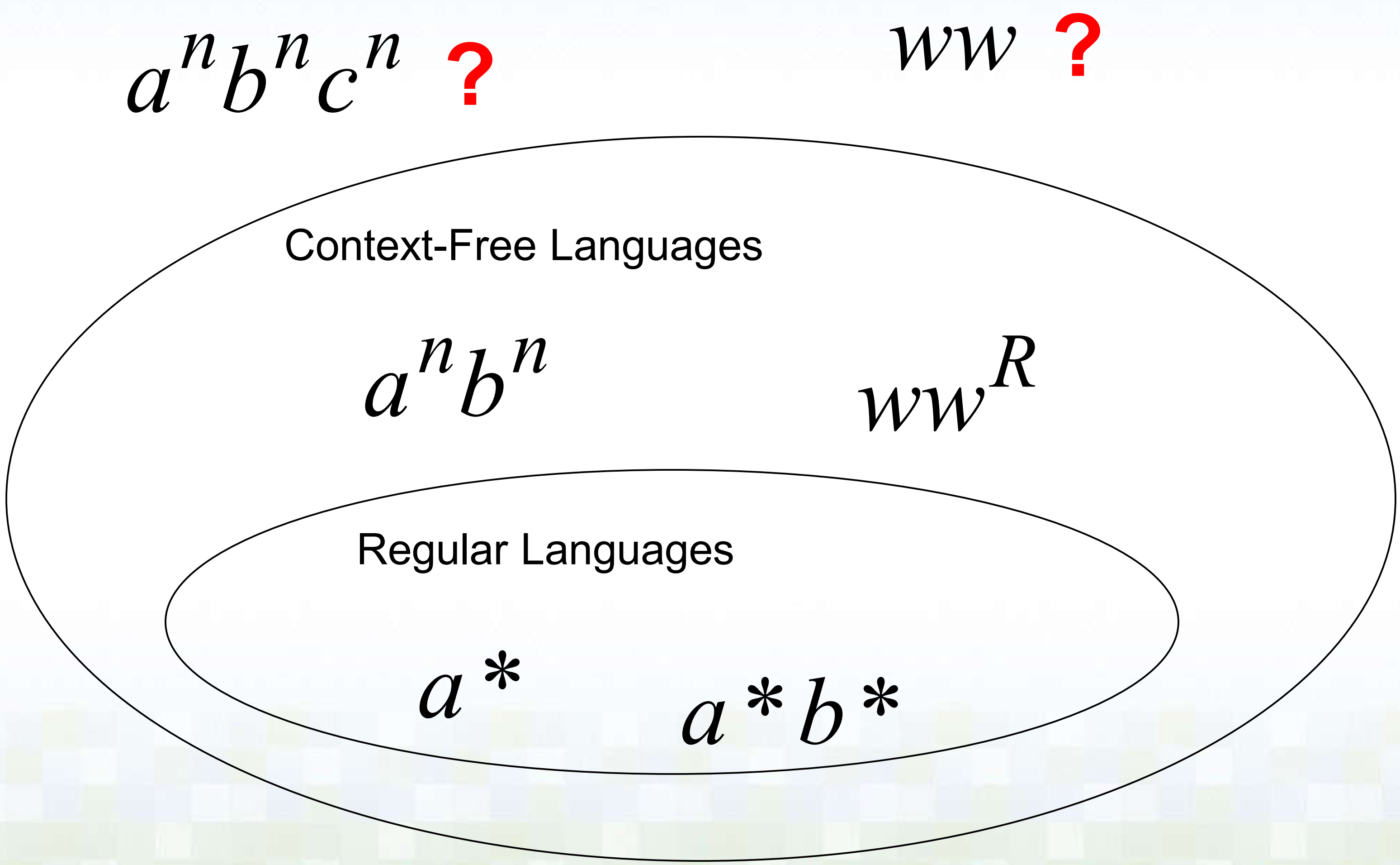
CS372
FORMAL LANGUAGES & THE THEORY OF COMPUTATION

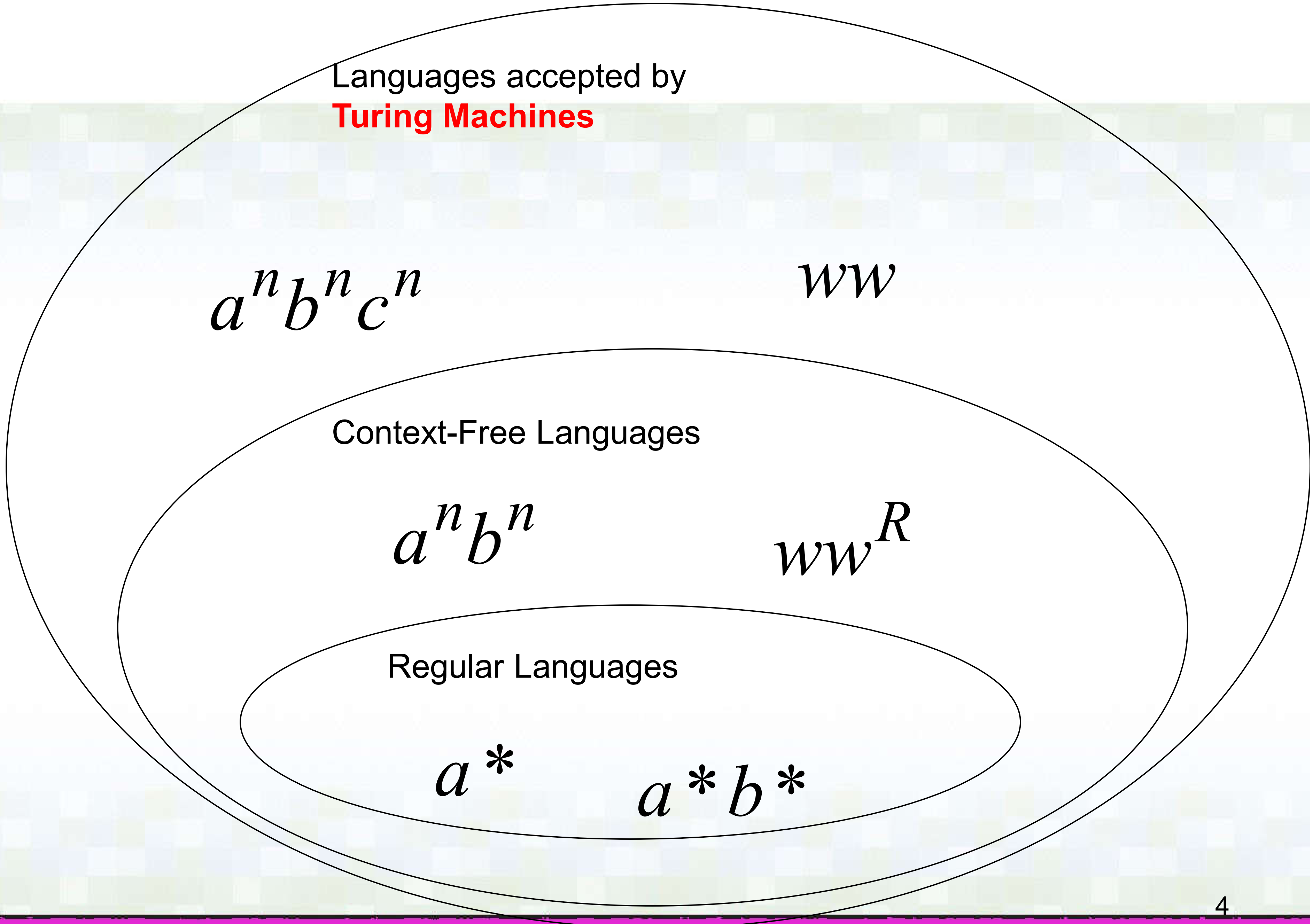
Dr.Nguyen Thi Thu Huong
Phone: +84 24 38696121, Mobi: +84 903253796
Email: huongnt@soict.hust.edu.vn,
huong.nguyenthithu@hust.edu.vn

Unit 7

Turing Machines

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^*

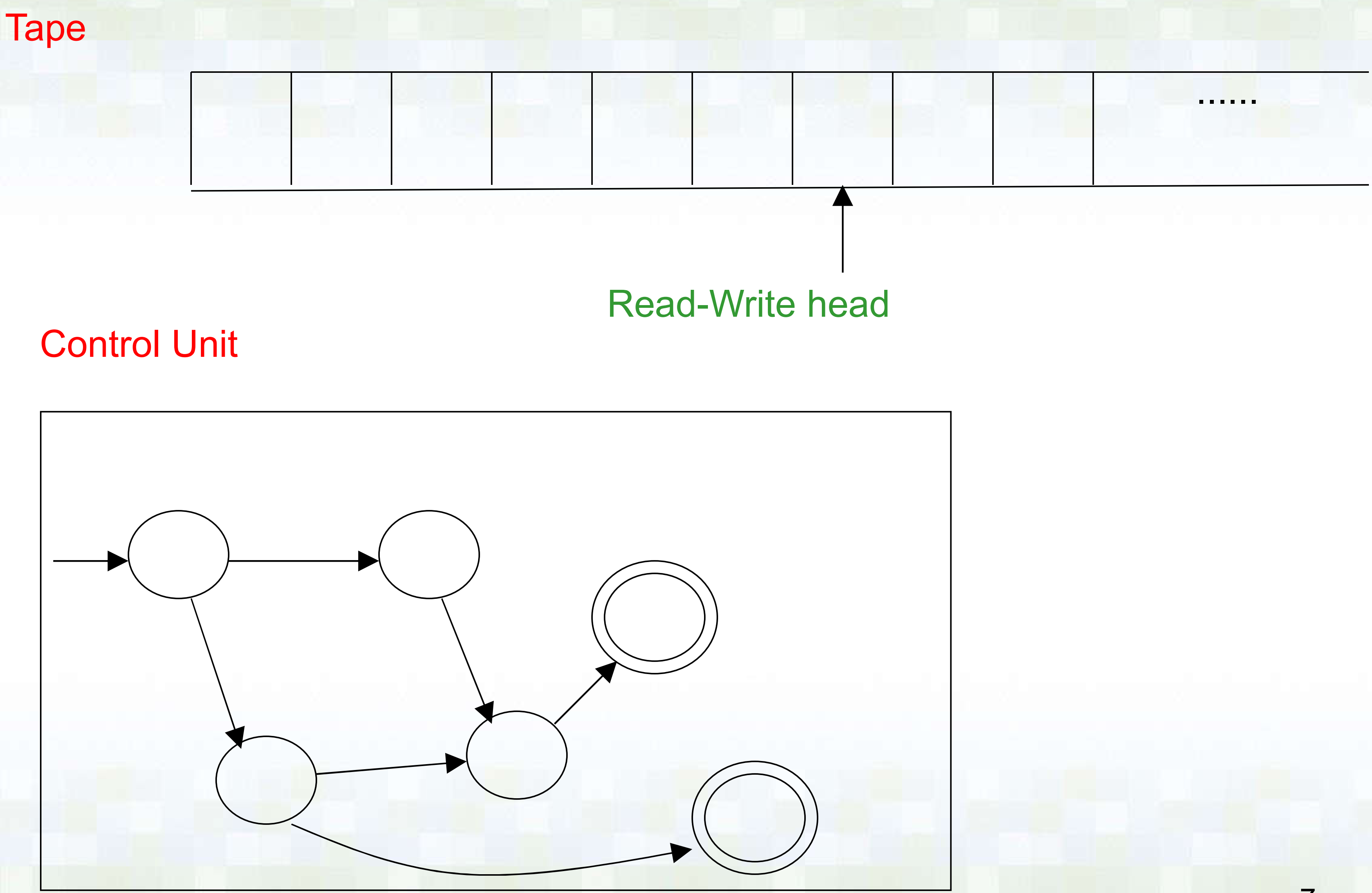
Turing machines

- Informal description of a Turing Machine
- Formal definition of a Turing machine
- Configuration
- Decidable languages

Informal Description

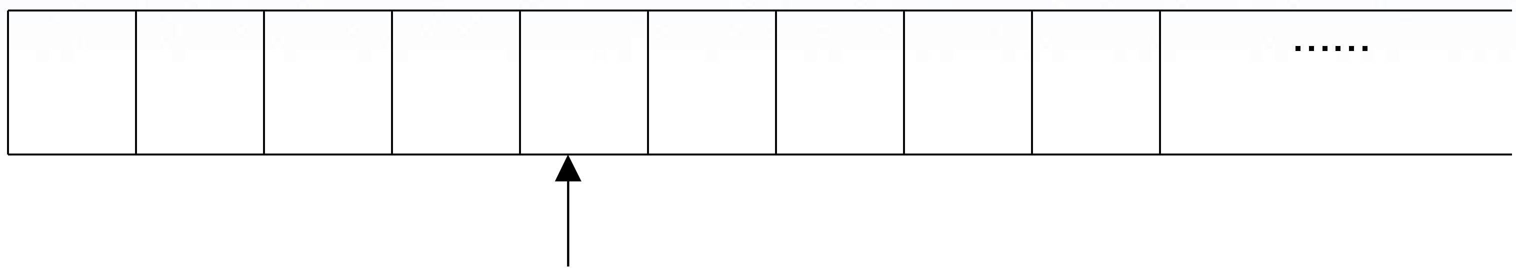
- Components
- The tape
- State
- Transition

Components of a Turing Machine



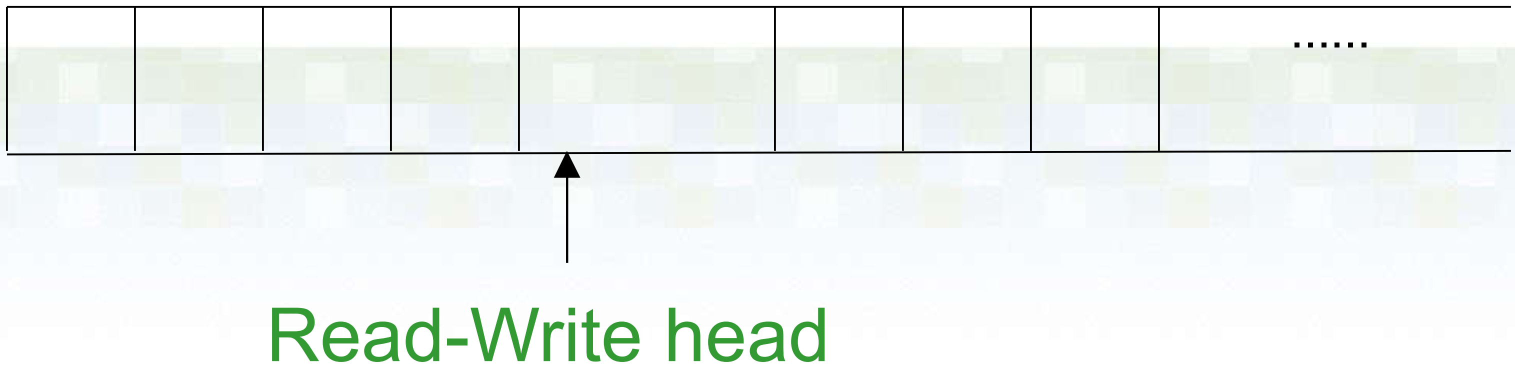
The Tape

No boundaries -- infinite length



Read-Write head

The head moves Left or Right

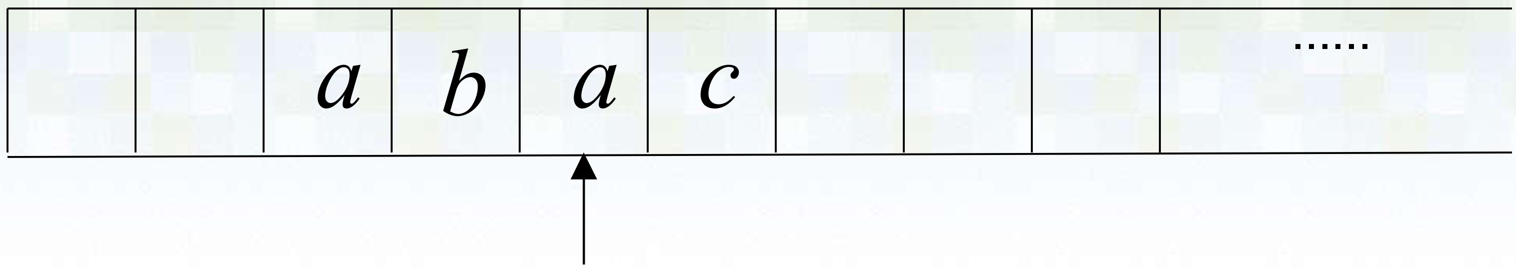


The head at each transition (time step):

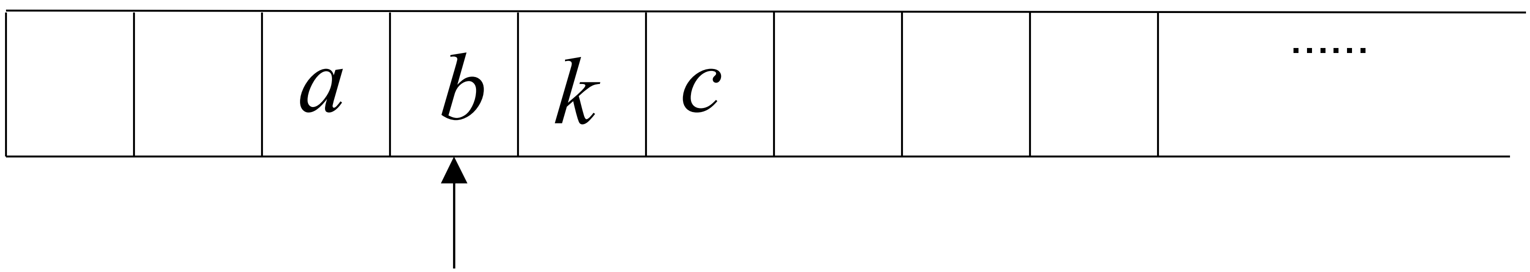
1. Reads a symbol
2. Writes a symbol
3. Moves Left or Right

Example:

Time 0



Time 1

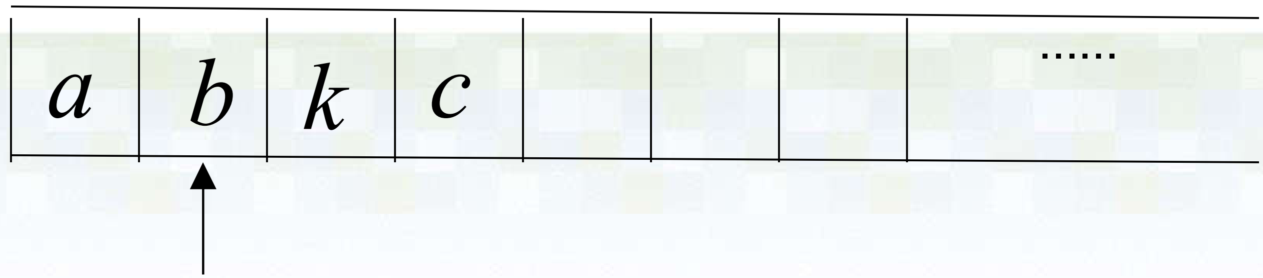


1. Reads a

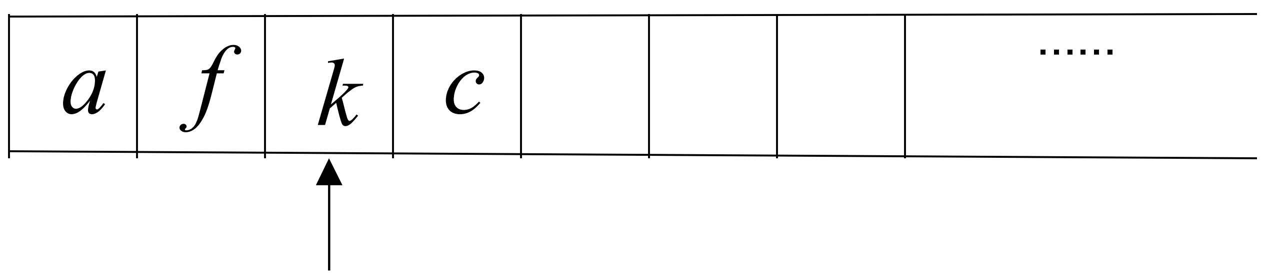
2. Writes k

3. Moves Left

Time 1

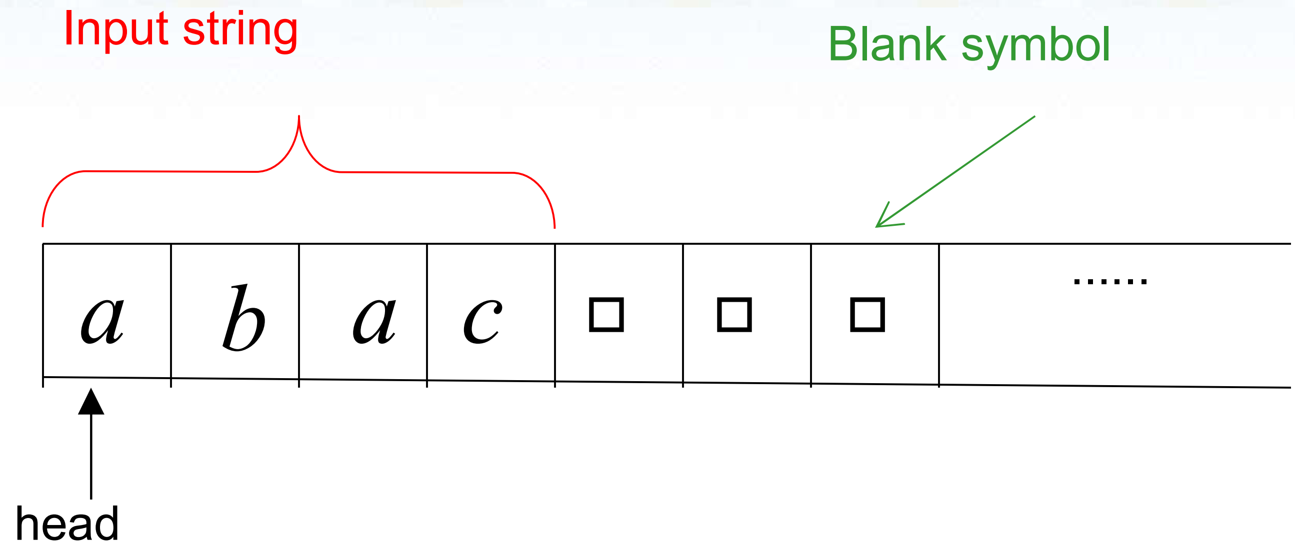


Time 2



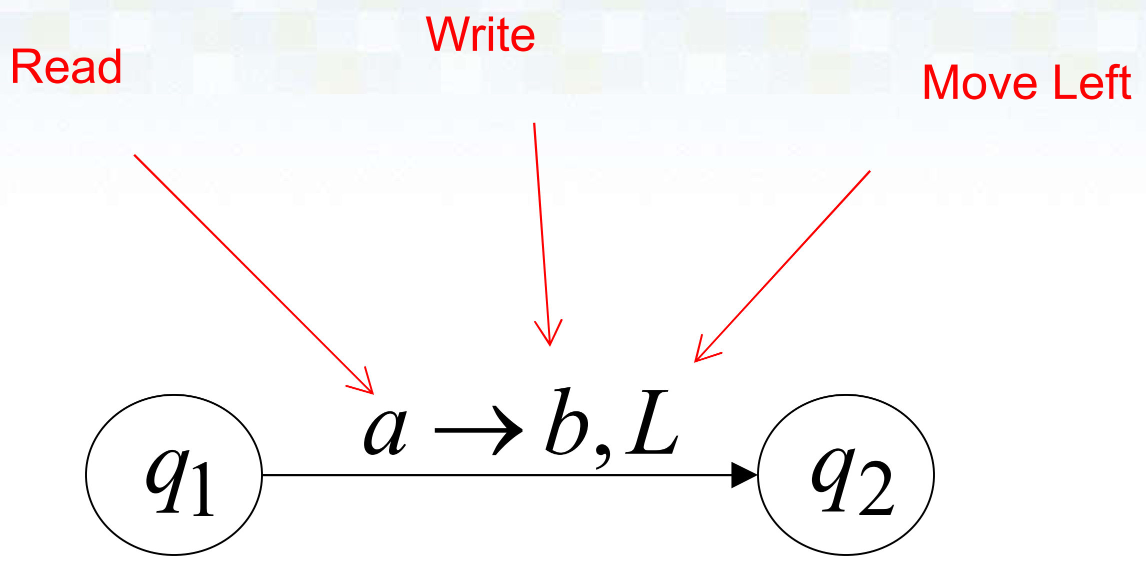
1. Reads b
2. Writes f
3. Moves Right

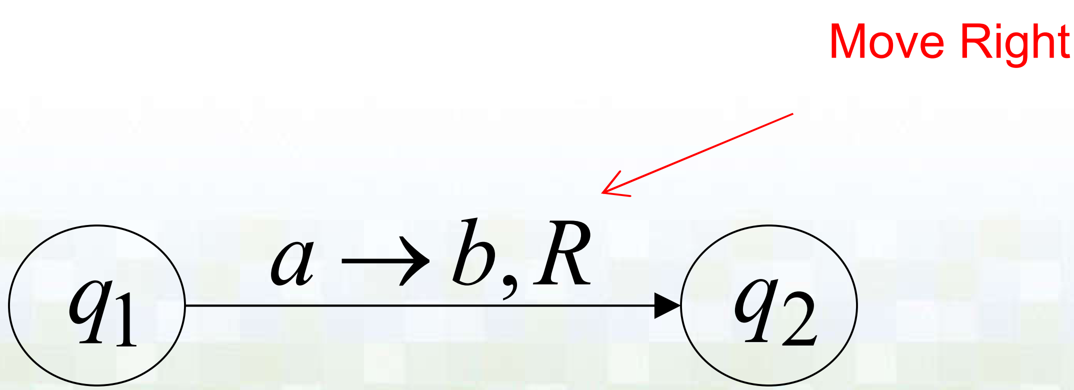
The Input String



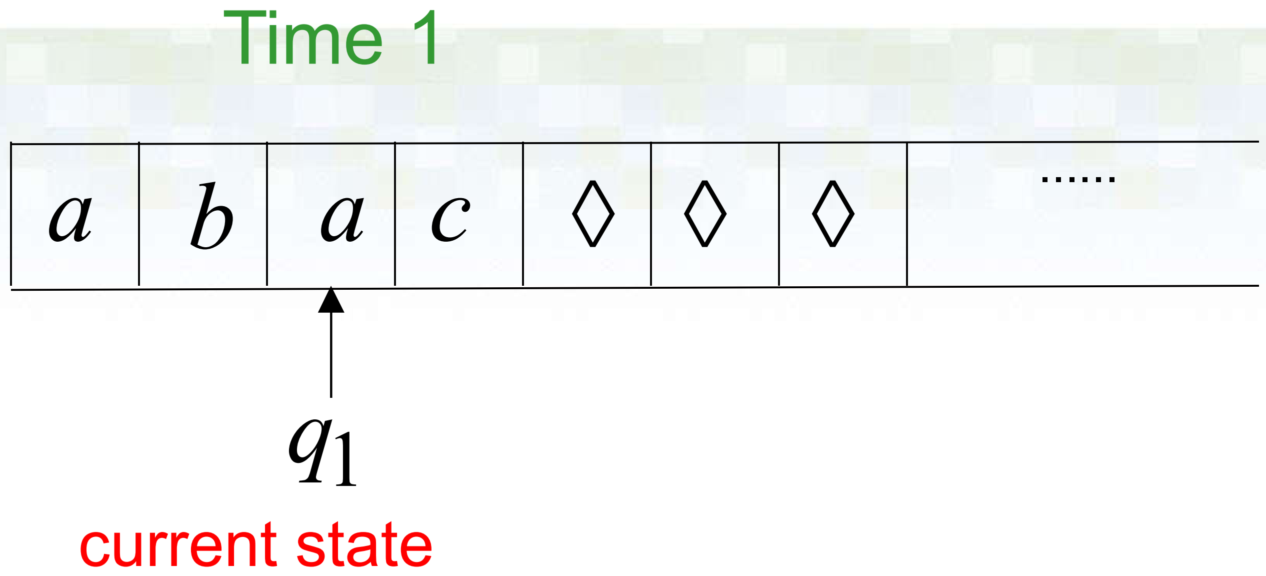
Head starts at the leftmost position
of the input string

States & Transitions

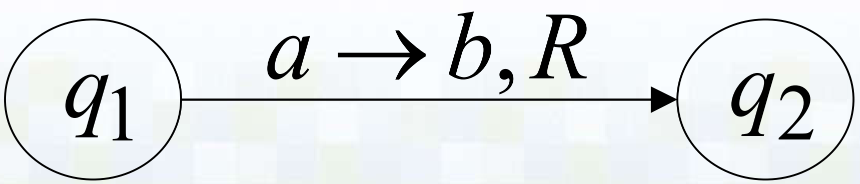




Example:

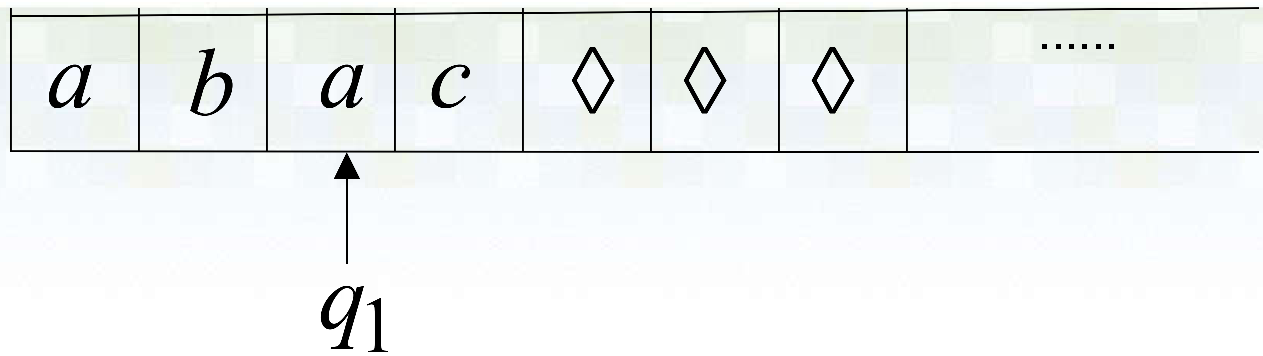


Time 1
a b a c $\Diamond$ $\Diamond$ $\Diamond$ ......
q_1
current state



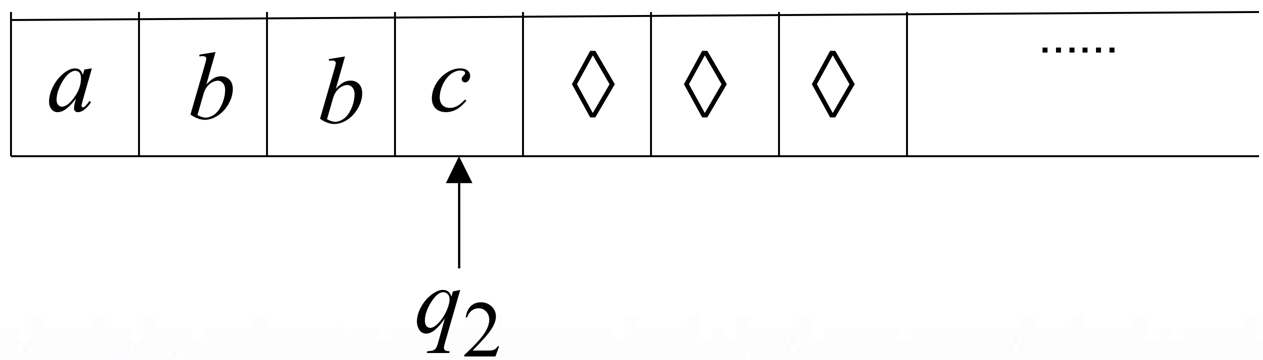
q_1 $\xrightarrow{a \rightarrow b, R}$ q_2

Time 1

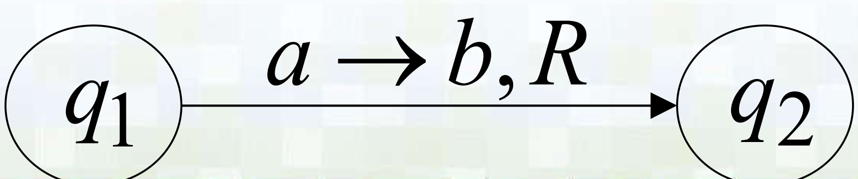


a b a c $\Diamond$ $\Diamond$ $\Diamond$ ......
q_1

Time 2



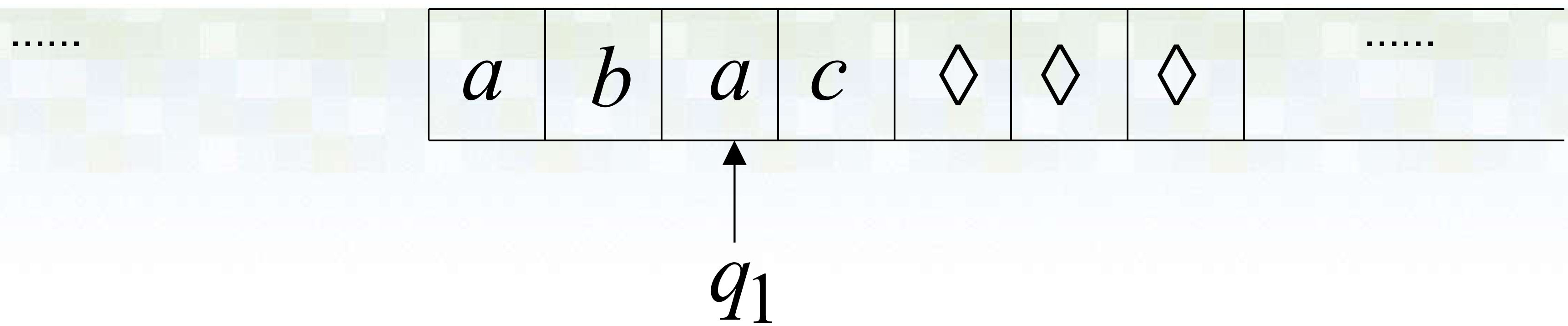
a b b c $\Diamond$ $\Diamond$ $\Diamond$ ......
q_2



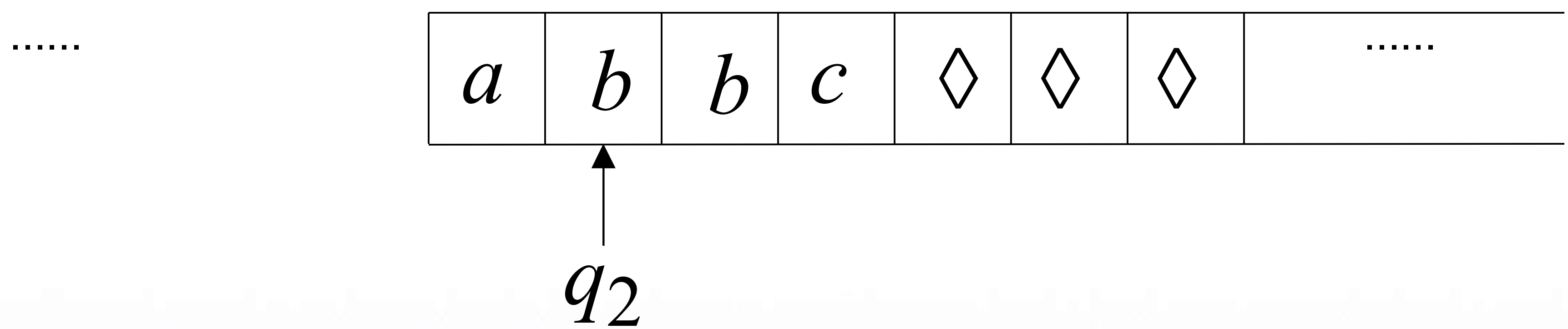
q_1
$a \rightarrow b, R$
q_2

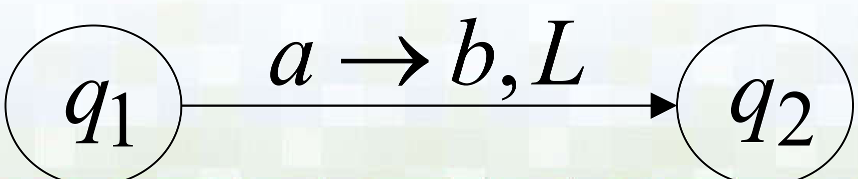
Example:

Time 1



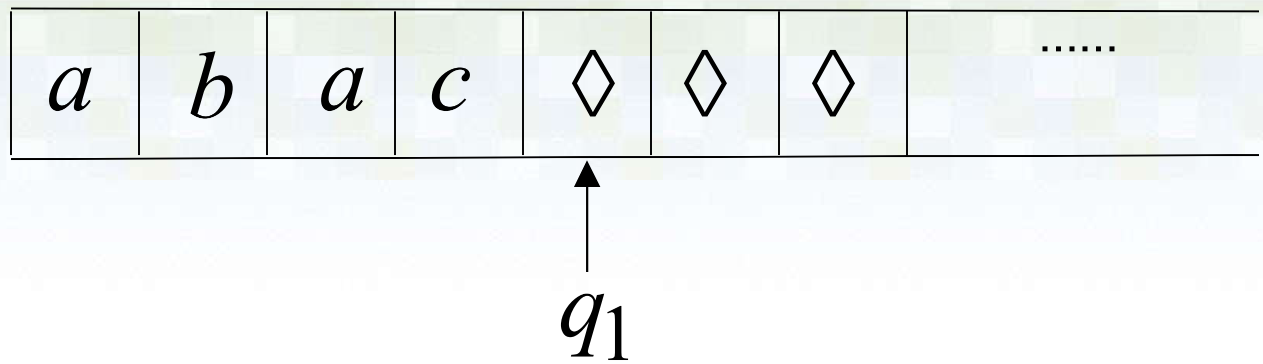
Time 2



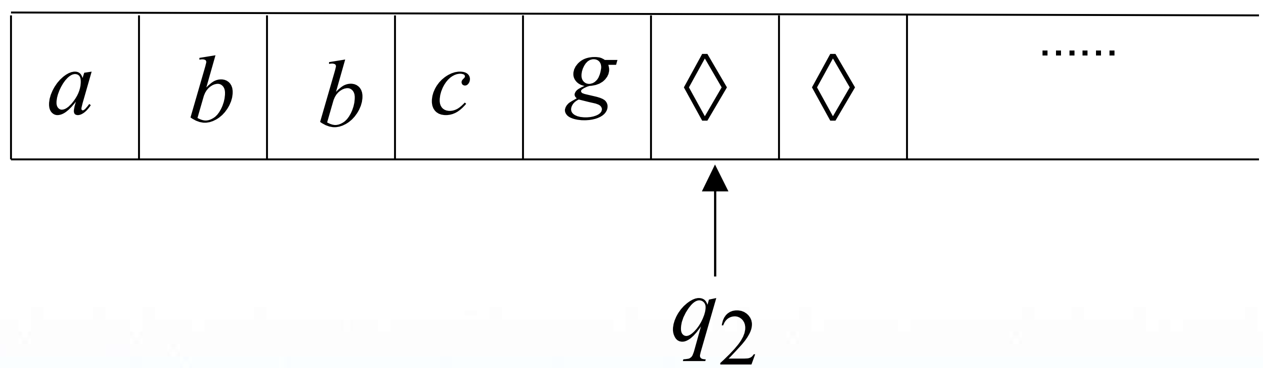


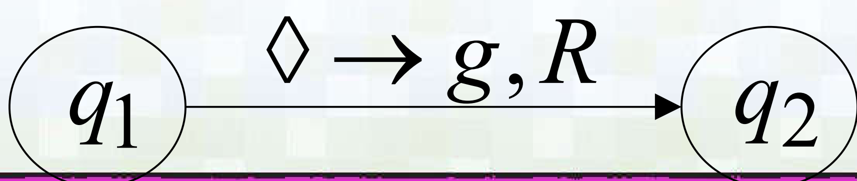
Example:

Time 1



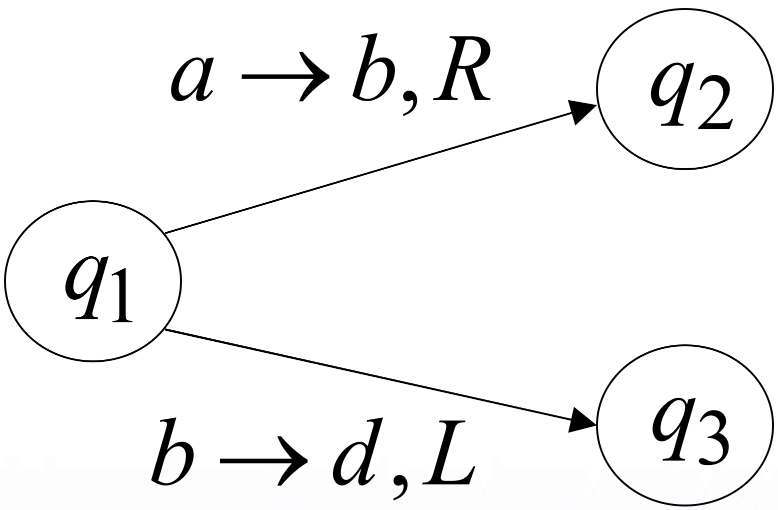
Time 2



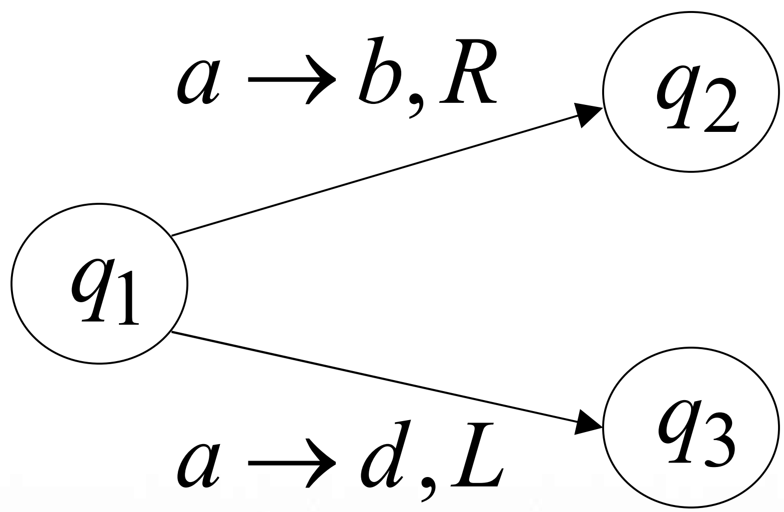


Determinism

Allowed



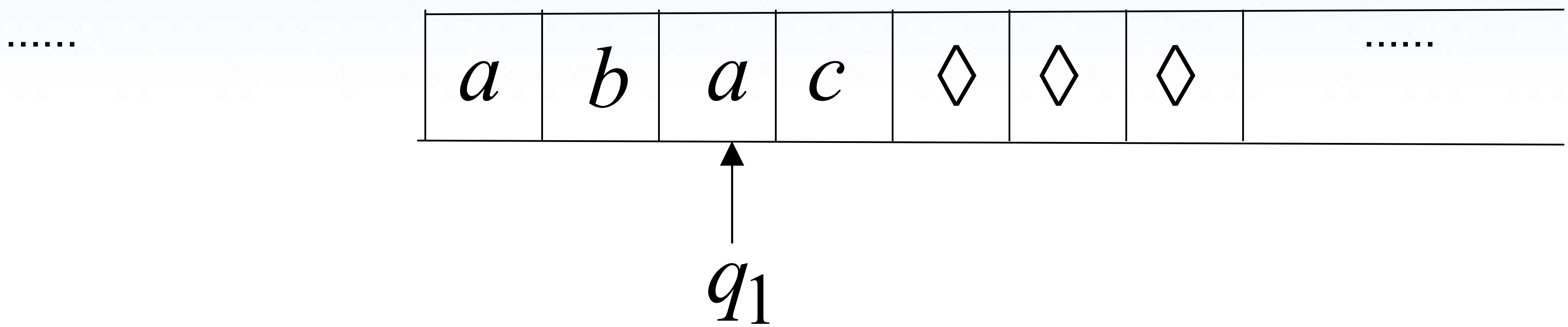
Not Allowed

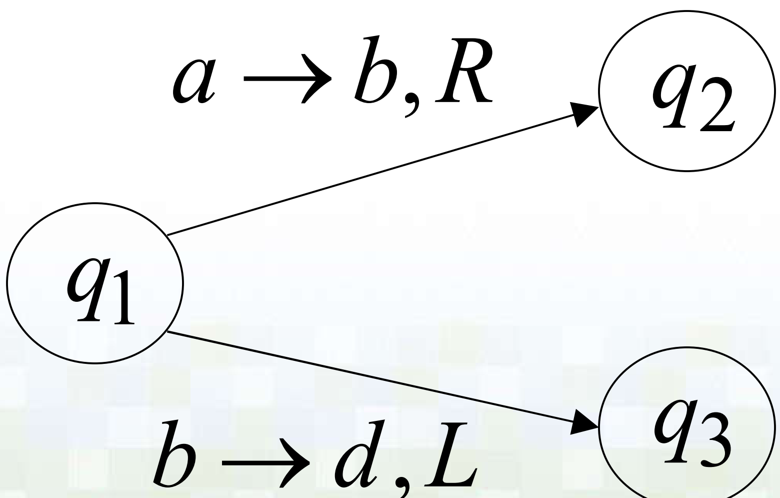


No epsilon transitions allowed

Partial Transition Function

Example:





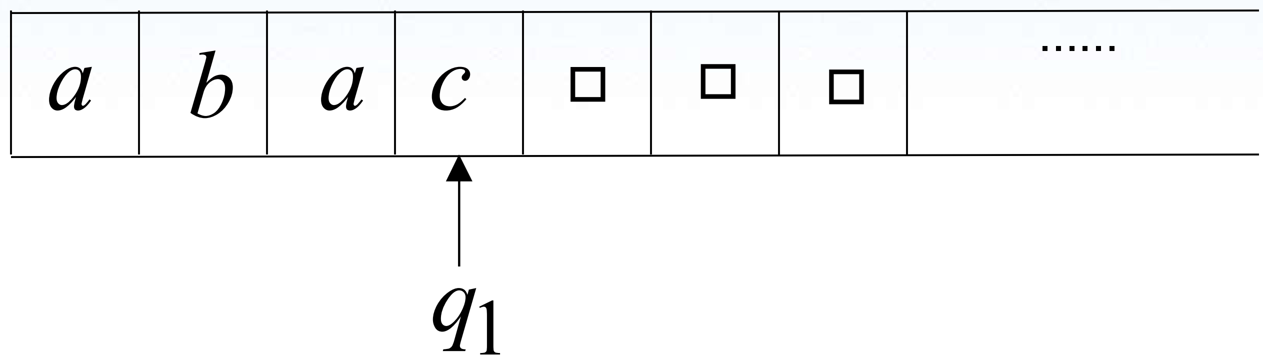
Allowed:

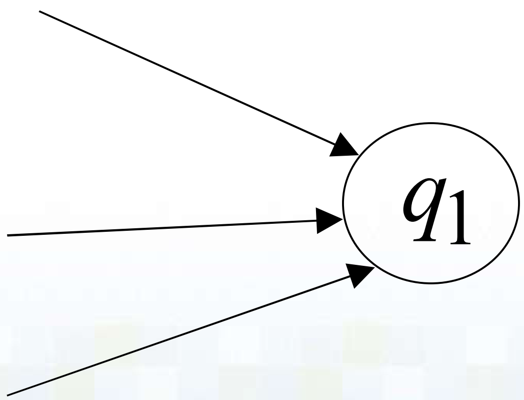
No transition
for input symbol c

Halting

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

Halting Example 1:

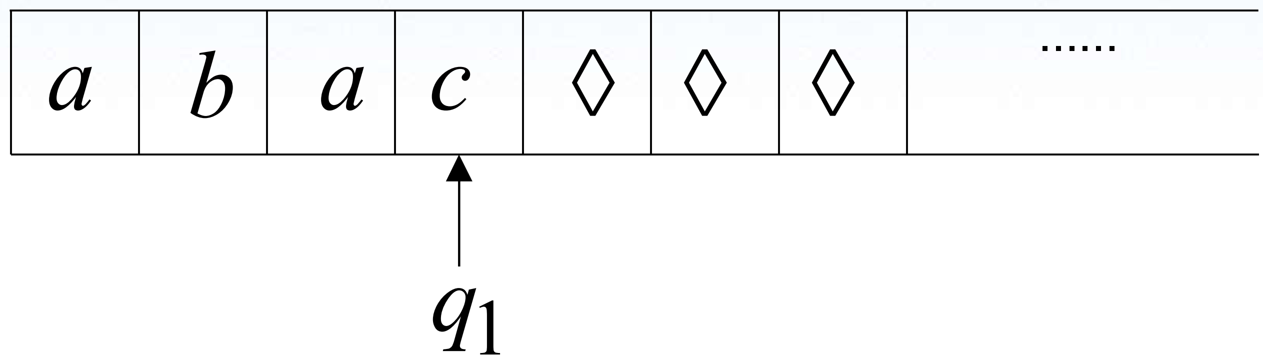


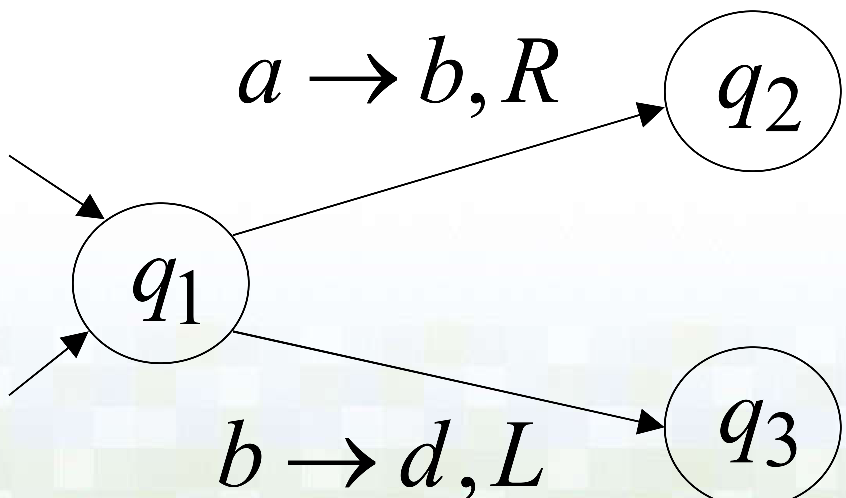


No transition from q_1

HALT!!!

Halting Example 2:





No possible transition from q_1 and symbol c

HALT!!!

Accepting States

Allowed

Not Allowed

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

Acceptance

Accept Input

string

If machine halts
in an accept state

Reject Input

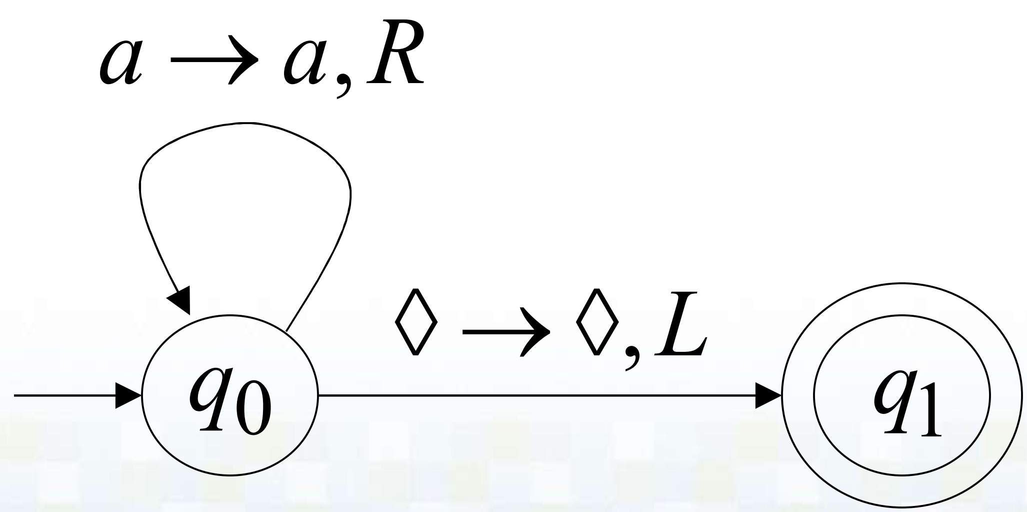
string

If machine halts
in a non-accept state
or
If machine enters
an *infinite loop*

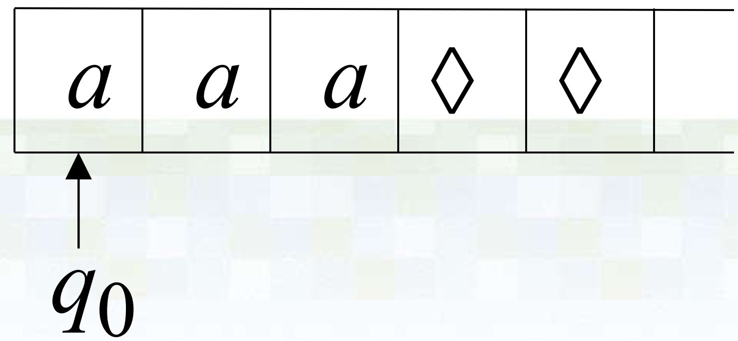
Turing Machine Example

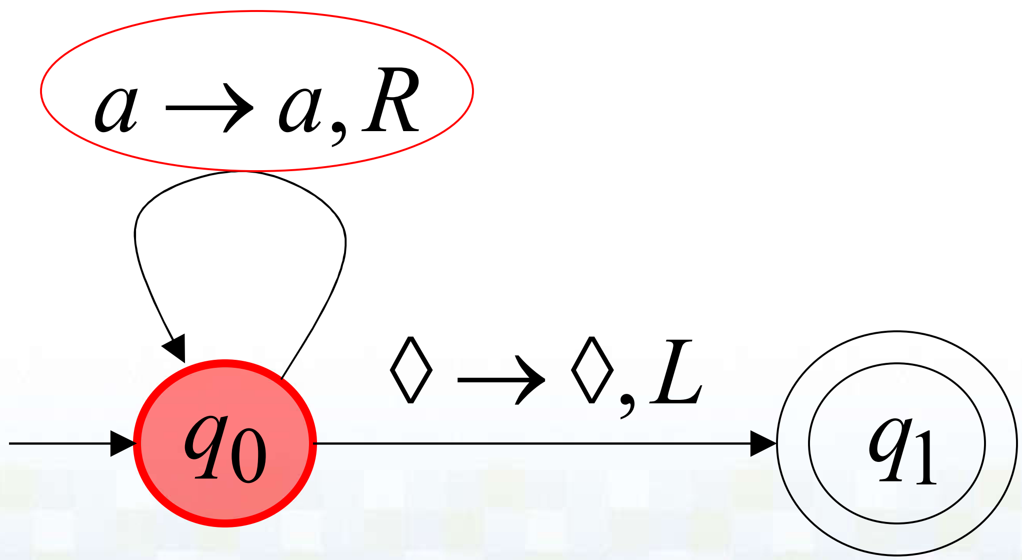
Input alphabet $\Sigma = \{a, b\}$

Accepts the language: a^*

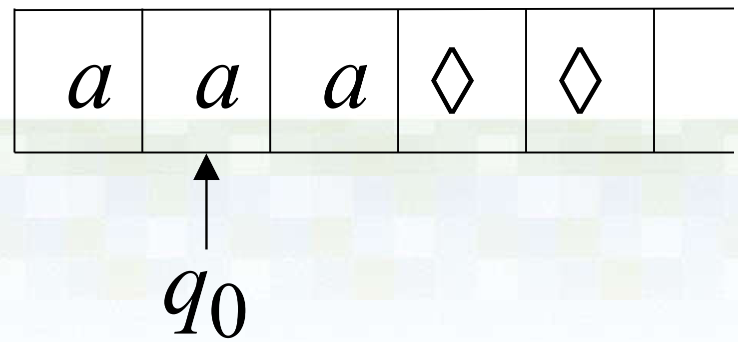


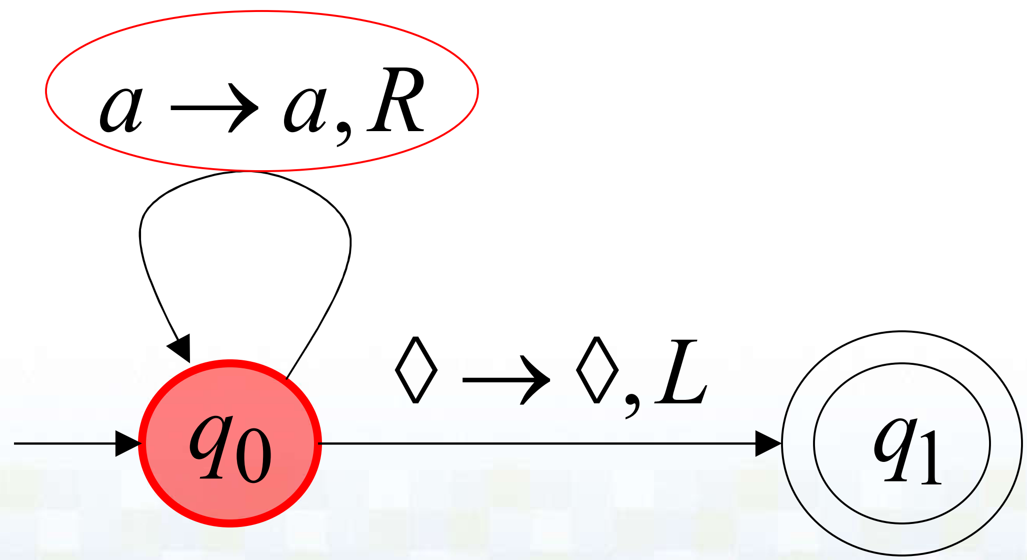
Time 0


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

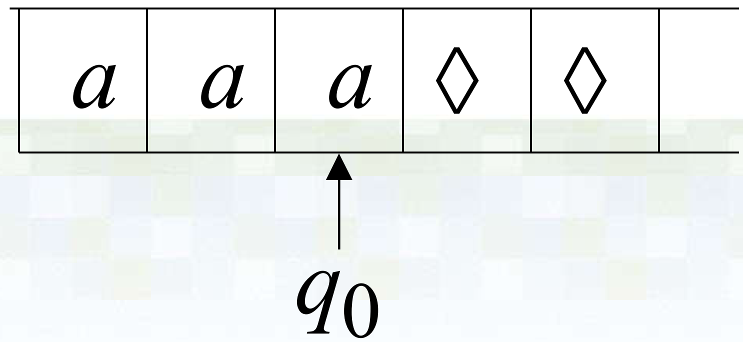

$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

Time 1

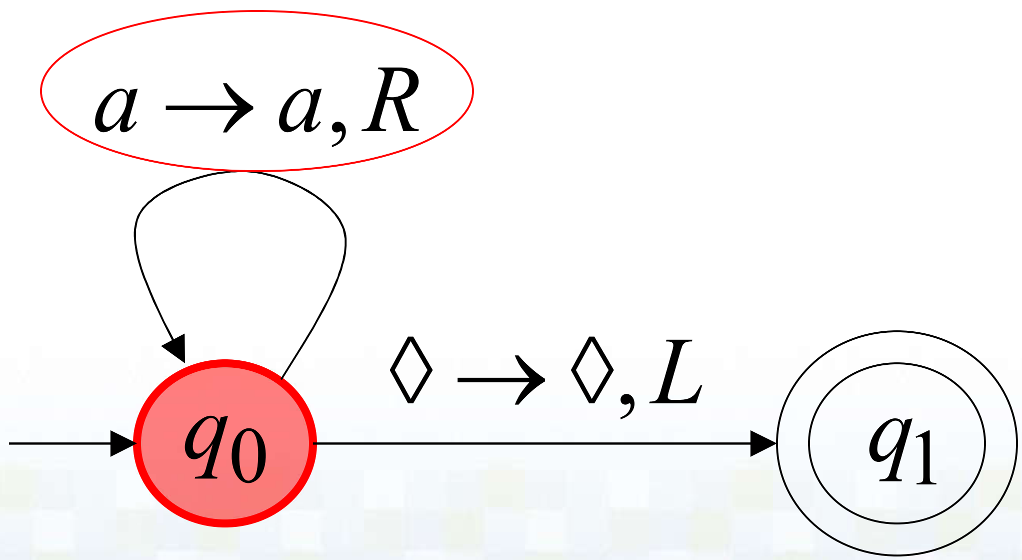




Time 2

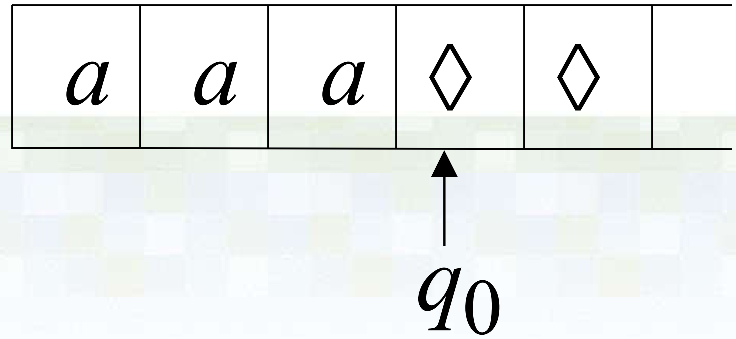


a a a $\Diamond$ $\Diamond$
q_0

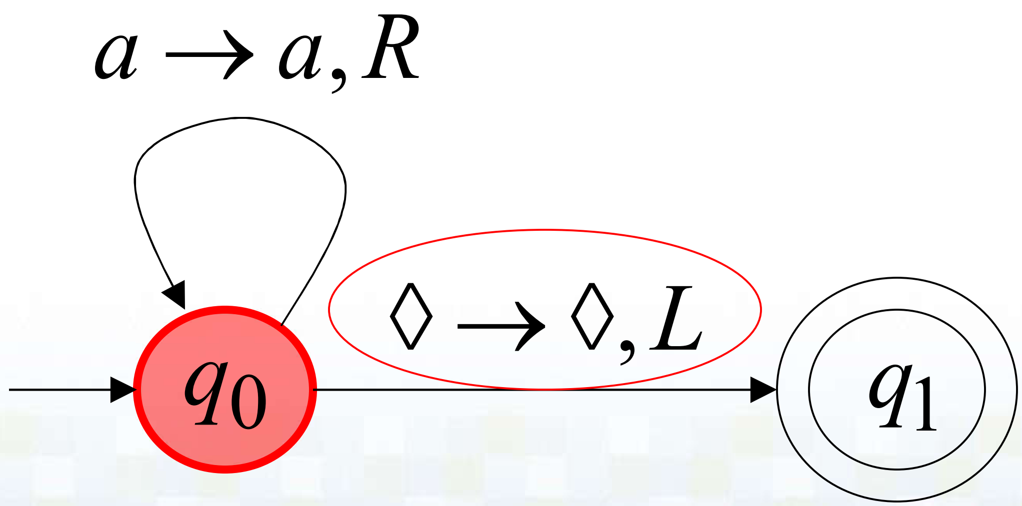


$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

Time 3

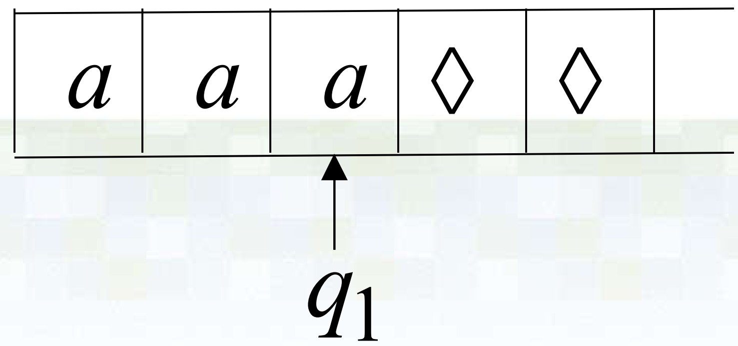


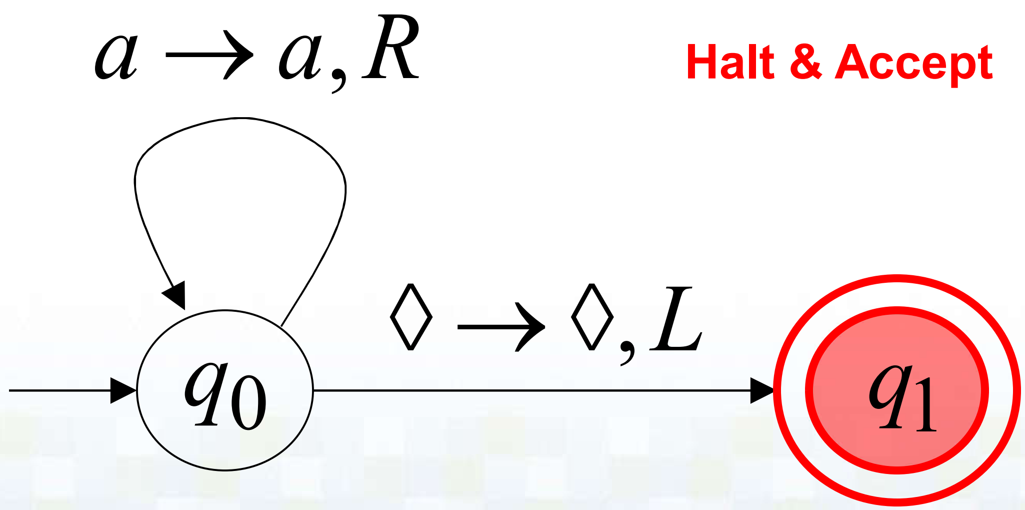
a a a $\Diamond$ $\Diamond$
q_0



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

Time 4

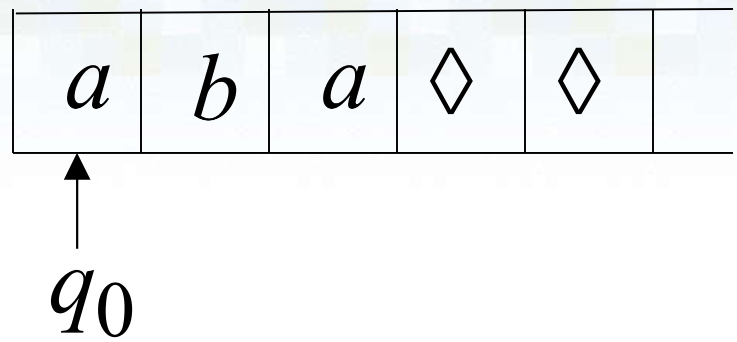


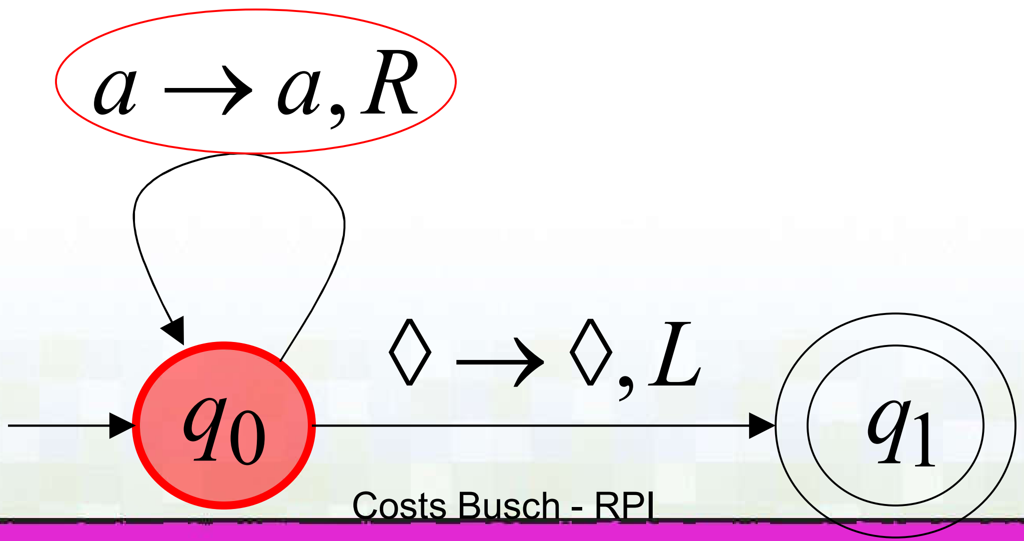


Halt & Accept

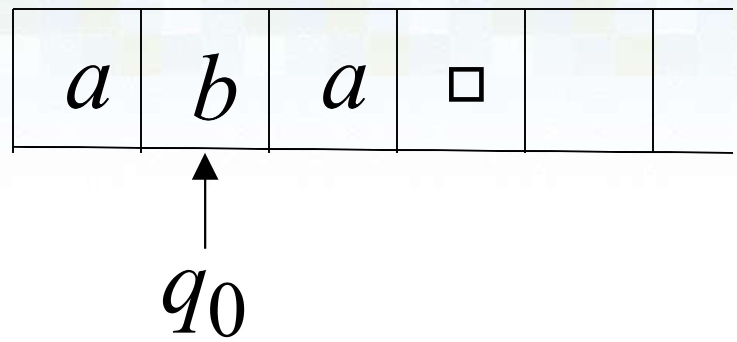
Rejection Example

Time 0



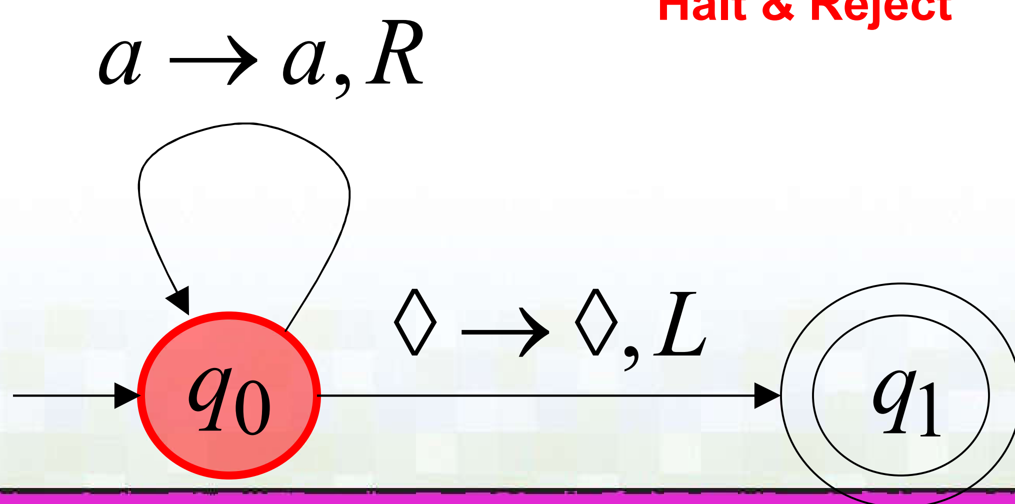


Time 1



No possible Transition

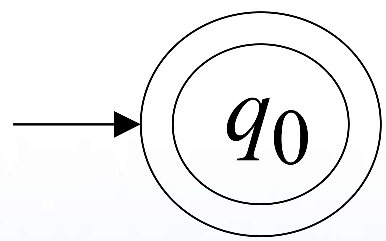
Halt & Reject



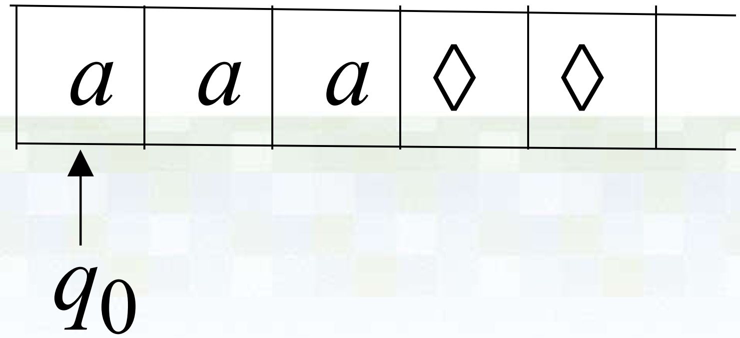
A simpler machine for same language

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

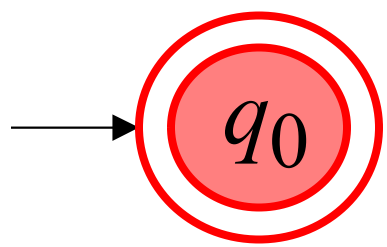
Accepts the language: a^*



Time 0



Halt & Accept

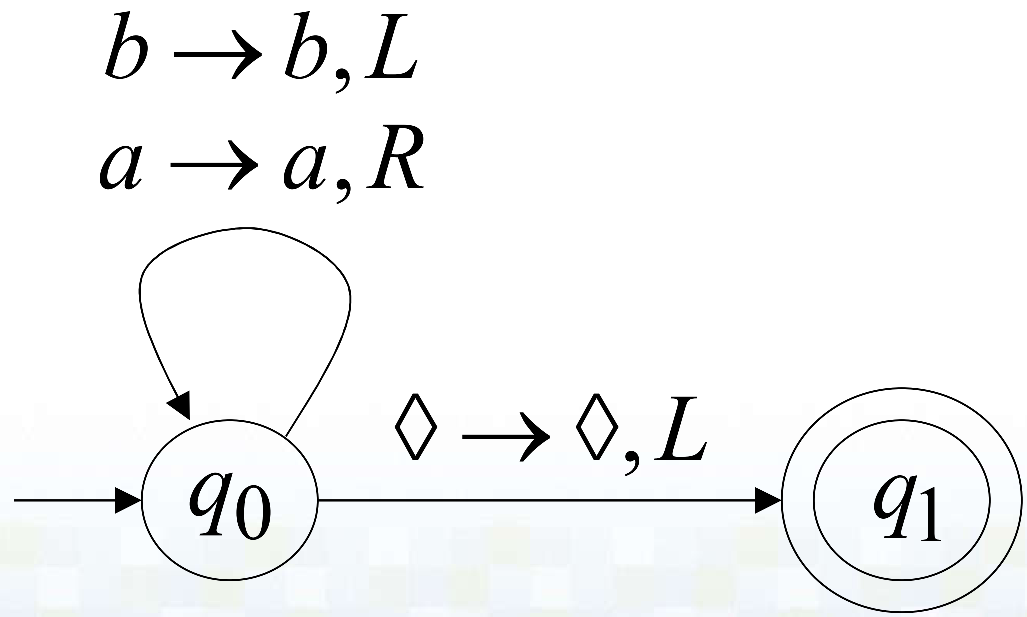


Not necessary to scan input

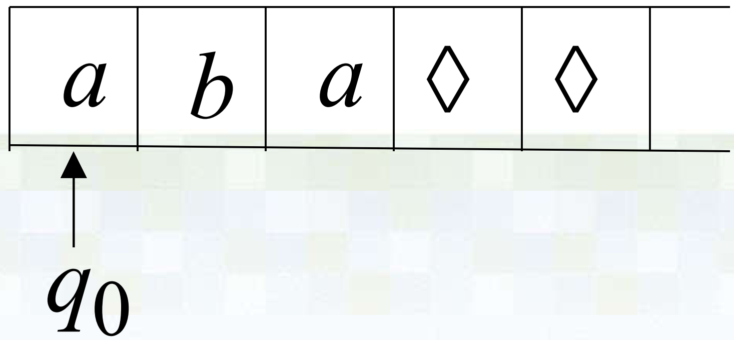
Infinite Loop Example

A Turing machine
for language

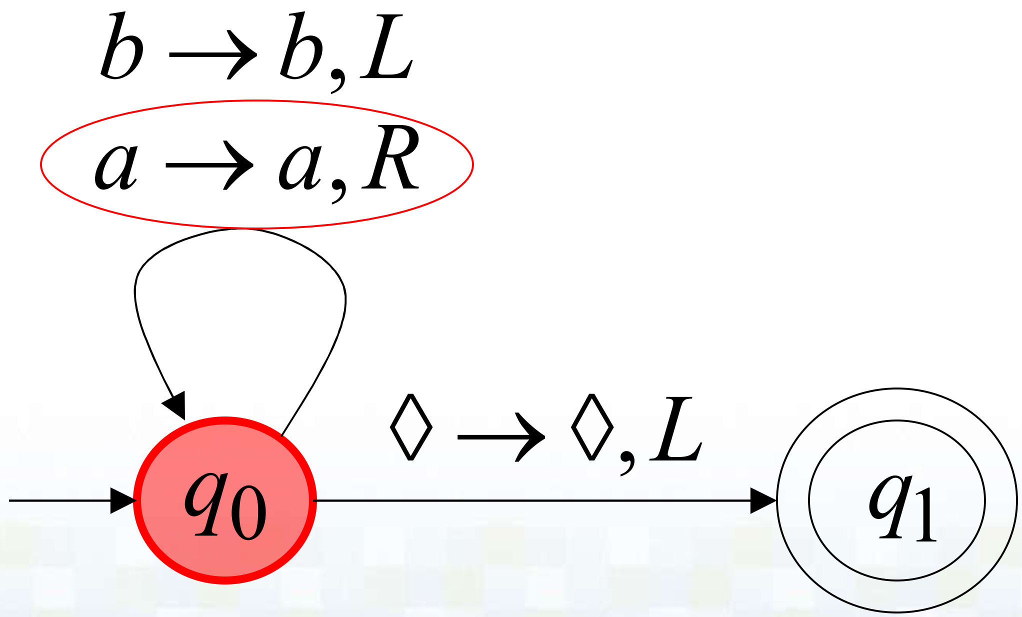
$$a^* + b(a+b)^*$$



Time 0

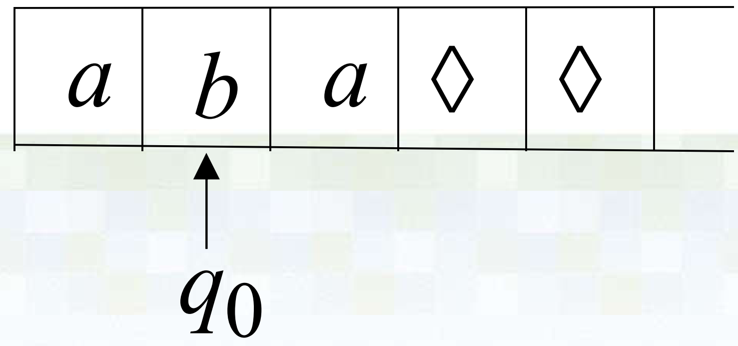


a
b
a
◊
◊
q_0

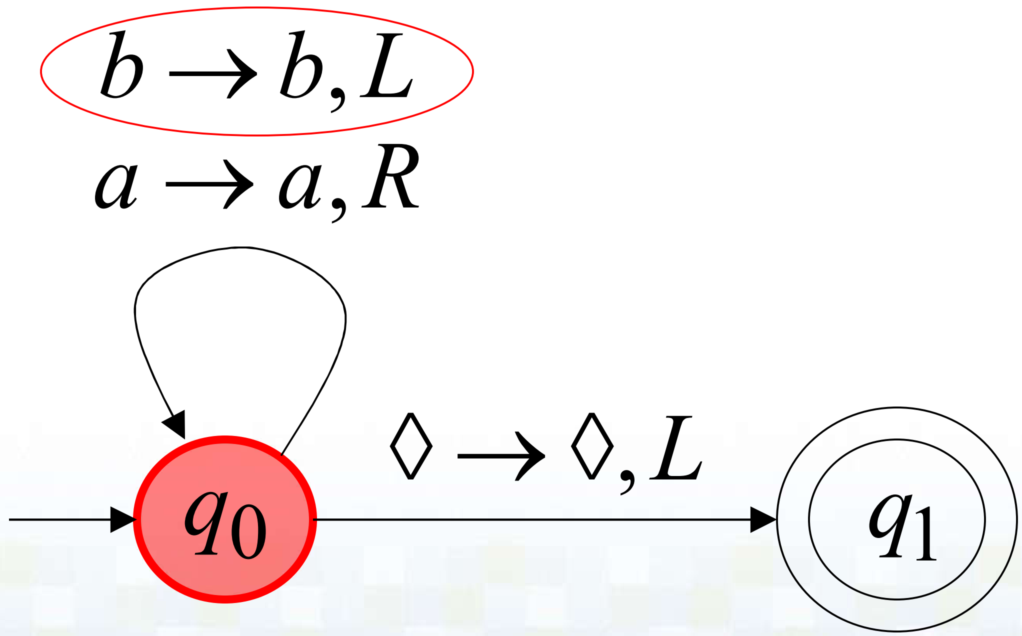


$b \rightarrow b, L$
$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

Time 1

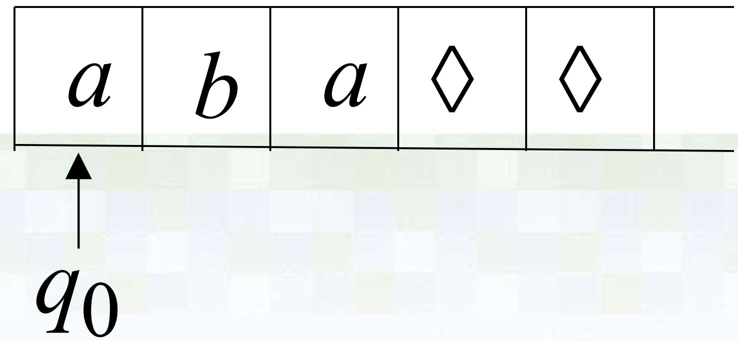


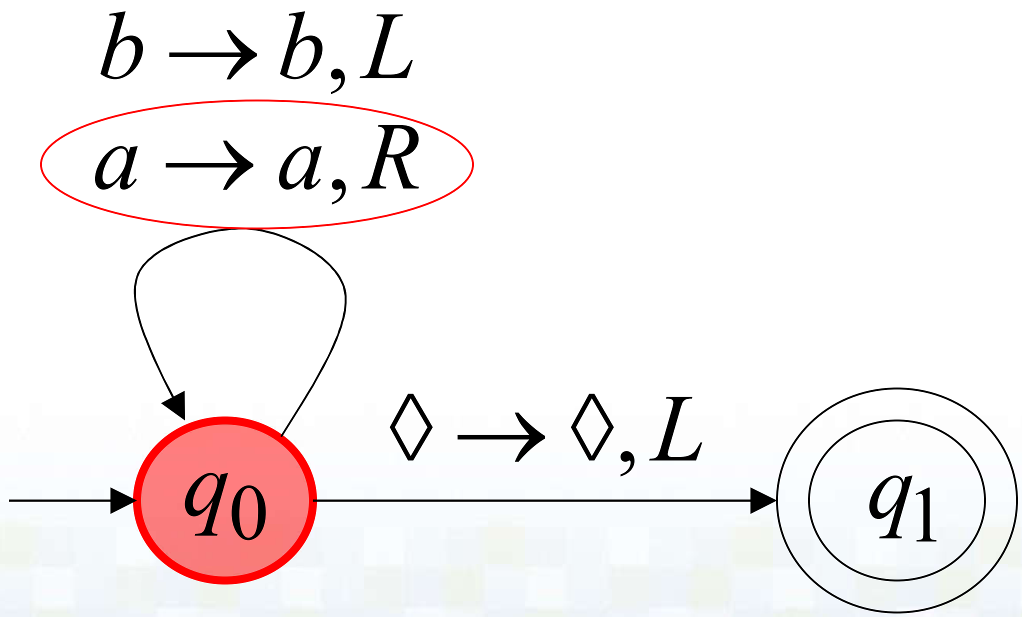
a b a $\Diamond$ $\Diamond$
q_0



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

Time 2


a
b
a
◊
◊
q_0

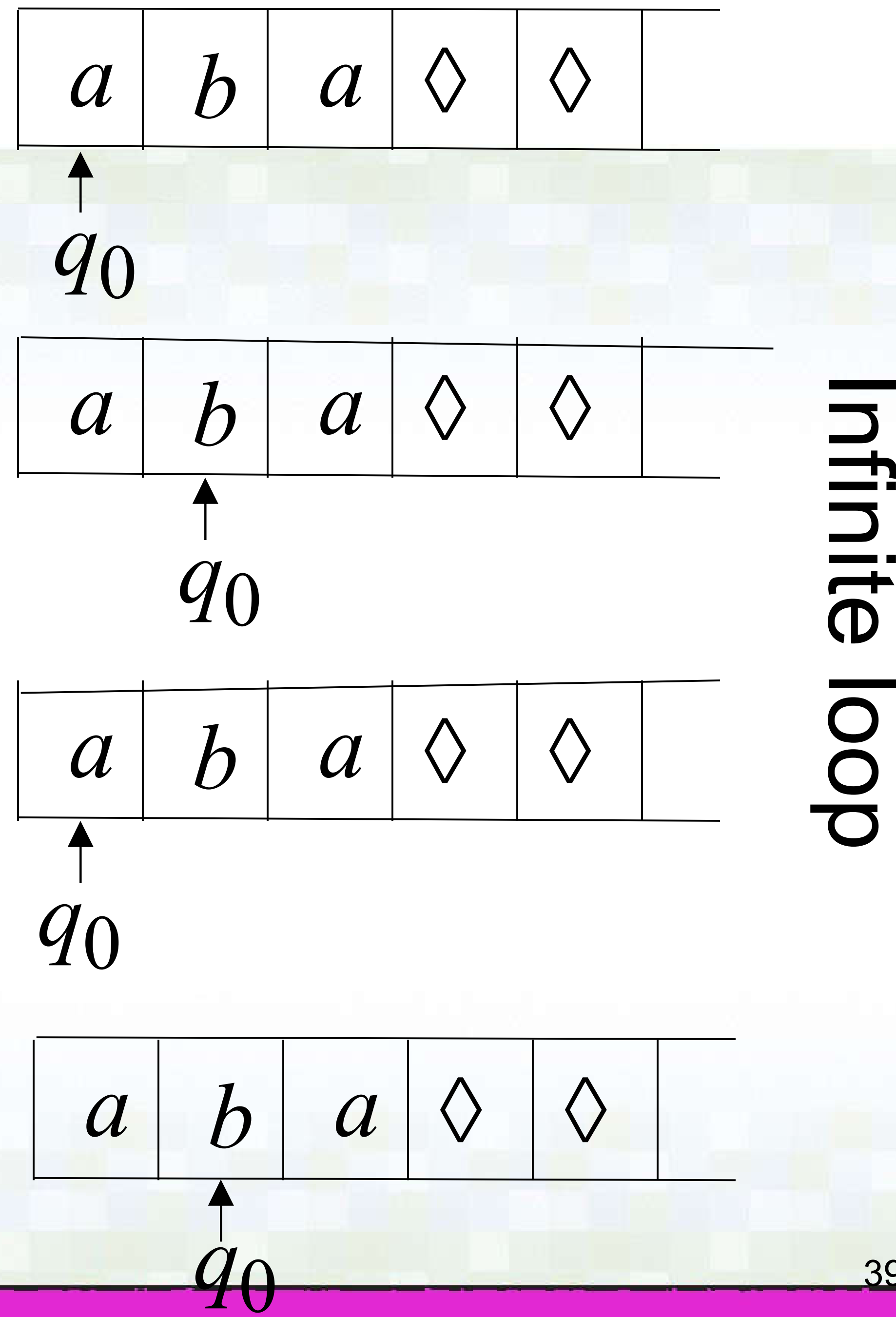

$b \rightarrow b, L$
$a \rightarrow a, R$
$\Diamond \rightarrow \Diamond, L$
q_0
q_1

Time 2

Time 3

Time 4

Time 5



Formal Definition of a Turing machine

A Turing Machine is represented by a 7-tuple

$T = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$:

Q is a finite set of states

Σ is the input alphabet, where $\square \notin \Sigma$

Γ is the tape alphabet, a superset of Σ; $\square \in \Gamma$

$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is the transition function

$q_0 \in Q$ is the start state

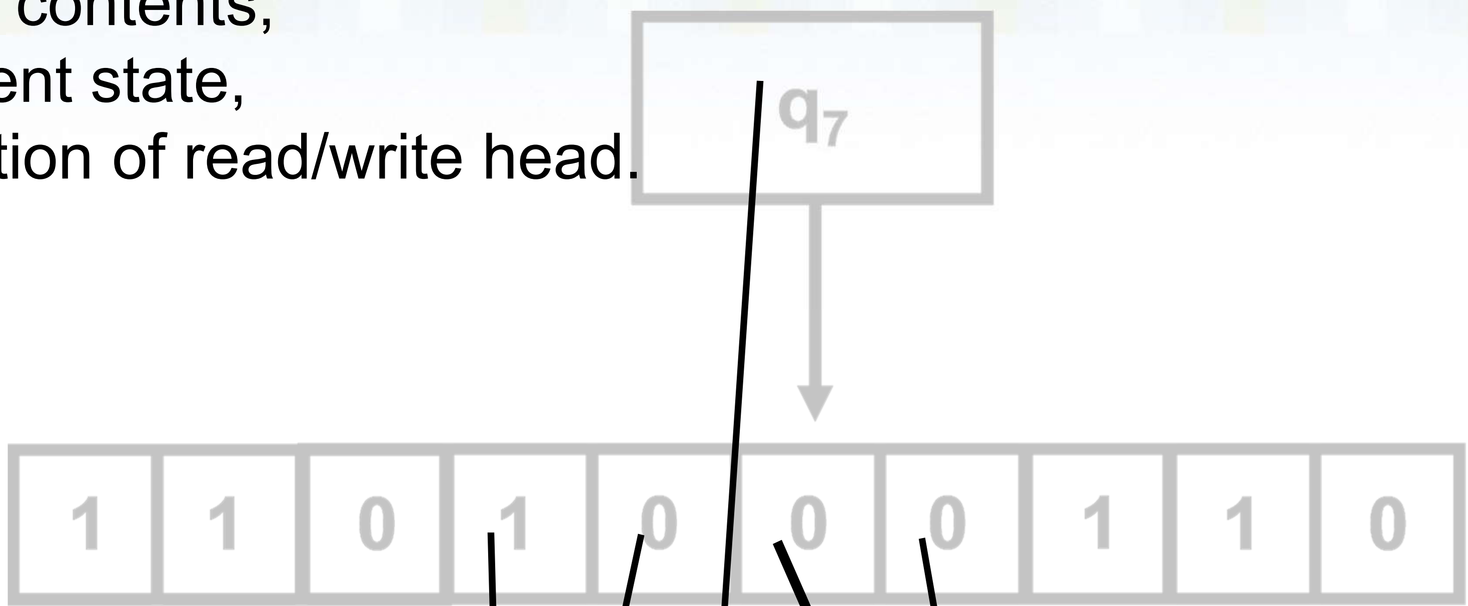
$q_{accept} \in Q$ is the accept state

$q_{reject} \in Q$ is the reject state, and $q_{reject} \neq q_{accept}$

Configuration of a Turing machine

A configuration in a MT includes

(1) tape contents,
(2) current state,
(3) location of read/write head.

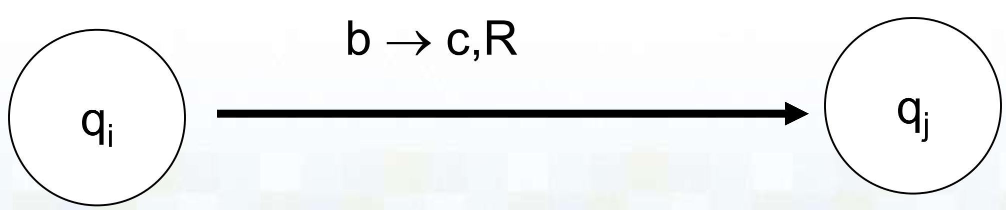


$$\mathbf{11010q_700110}$$

State diagrams of TMs

Like with PDA, we can represent Turing machines by (elaborate) diagrams.

If transition rule says: $\delta(q_i,b) = (q_j,c,R)$, then:



Languages of Turing machines

A TM *recognizes* a language iff *it accepts* all and only those strings in the language.

A language L is called *Turing-recognizable* or *recursively enumerable* iff some TM recognizes L.

A TM *decides* a language L iff it accepts all strings in L <u>and</u> rejects all strings not in L.

A language L is called *decidable* or *recursive* iff some TM decides L.

A TM that decides $\{ 0^{2^n} \mid n \geq 0 \}$

We want to accept iff:

- the input string consists entirely of zeros, and
- the number of zeros is a power of 2.

High-Level Idea.

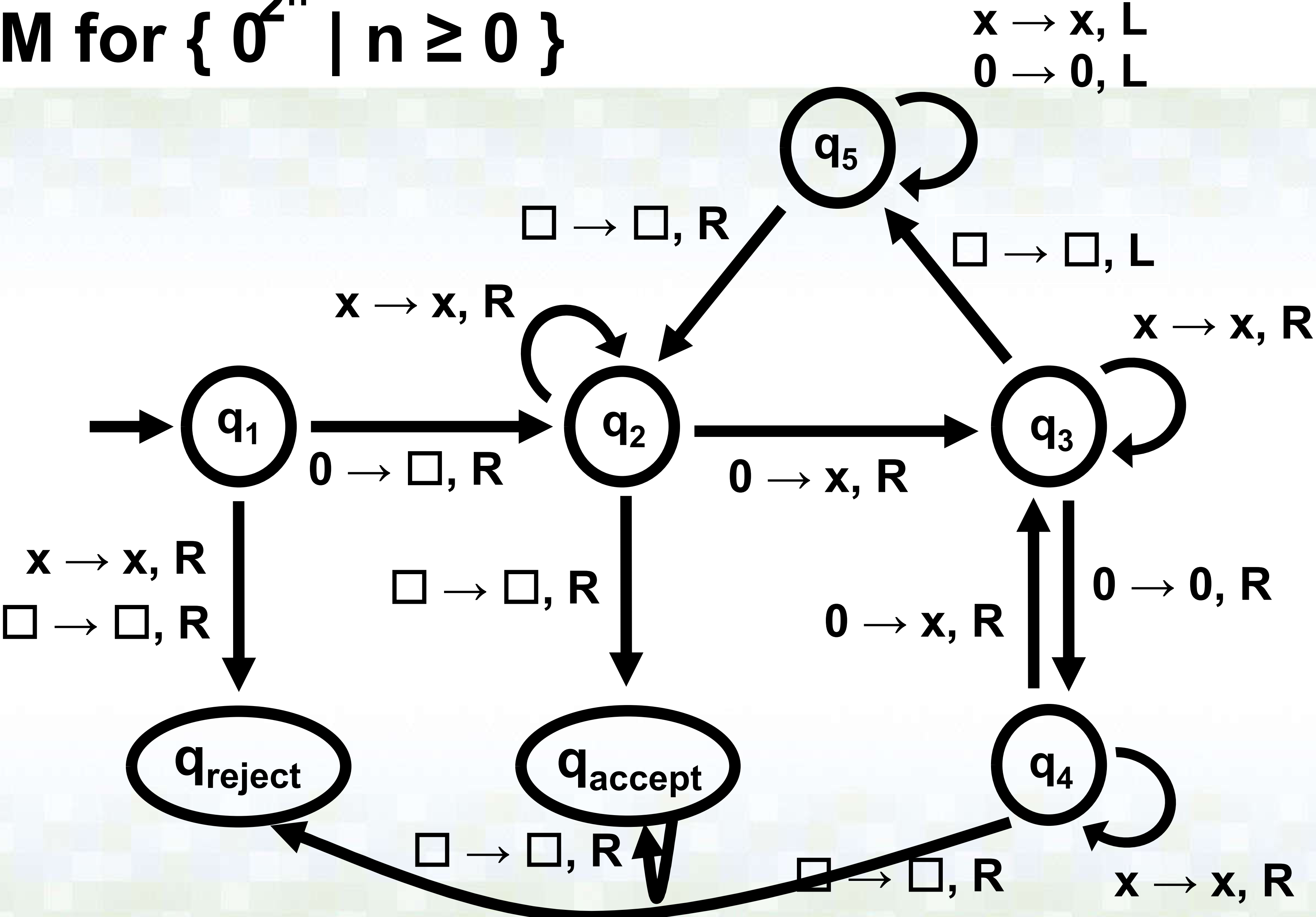
- Repeatedly divide the number of zeros in half until it becomes an odd number.
- If we are left with a single zero, then accept.
- Otherwise, reject.

Example: a TM that decides $\{ 0^{2^n} \mid n \geq 0 \}$

Description:

1. Sweep from left to right, cross out every other **0**. (Divides number in half.)
2. If in step 1, the tape had only one **0**, *accept*.
3. Else if the tape had an odd number of **0**’s, *reject*.
4. Move the head back to the first input symbol.
5. Go to step 1.

TM for { 0^{2^n} | n ≥ 0 }



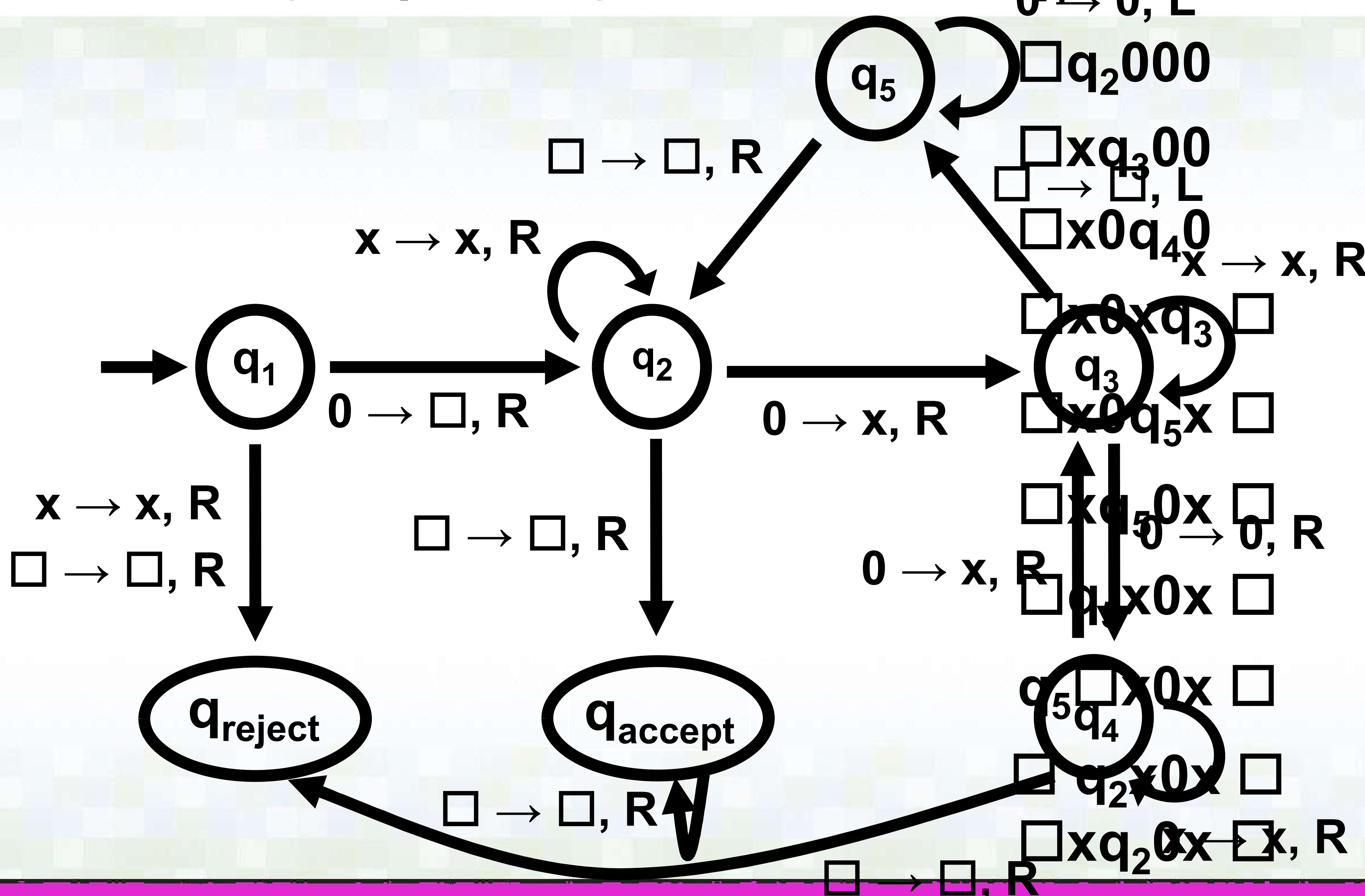
x → x, L
0 → 0, L
q5
□ → □, R
□ → □, L
x → x, R
x → x, R
q1
q2
q3
0 → □, R
0 → x, R
x → x, R
□ → □, R
□ → □, R
0 → 0, R
0 → x, R
qreject
qaccept
q4
□ → □, R
□ → □, R
x → x, R

Formal Description

$T = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$, $Q = \{q_1, q_2, q_3, q_4, q_5\}$,
$\Sigma = \{0\}$, $\Gamma = \{0, X, \square\}$, q_1 is the start state

1) $\delta(q_1, 0) = (q_2, \square, R)$	**Mark the beginning of string**
2) $\delta(q_1, \square) = (q_{reject}, \square, R)$	**Reject $\square$**
3) $\delta(q_1, X) = (q_{reject}, X, R)$	**Reject X**
4) $\delta(q_2, 0) = (q_3, X, R)$	**Mark 0 with X**
5) $\delta(q_2, X) = (q_2, X, R)$	**Move right through X's**
6) $\delta(q_2, \square) = (q_{accept}, \square, R)$	**Accept strings with length is a power of 2**
7) $\delta(q_3, 0) = (q_4, 0, R)$	**Skip one symbol 0**
8) $\delta(q_3, X) = (q_3, X, R)$	**Move right through X's**
9) $\delta(q_3, \square) = (q_5, \square, L)$	**Read blank symbol on the right. Move to state q5**
10) $\delta(q_4, 0) = (q_3, X, R)$	**Mark 0 with X (Skip one 0, mark one 0)**
11) $\delta(q_4, X) = (q_4, X, R)$	**Move right through X's**
12) $\delta(q_4, \square) = (q_{reject}, \square, R)$	**Reject strings with length is not a power of 2**
13) $\delta(q_5, 0) = (q_2, 0, L)$	**Move the head back to the beginning of string**
14) $\delta(q_5, X) = (q_5, X, L)$	
15) $\delta(q_5, \square) = (q_2, \square, R)$	

TM for $\{0^{2^n} \mid n \geq 0\}$



TM for $\{0^{2^n} \mid n \geq 0\}$