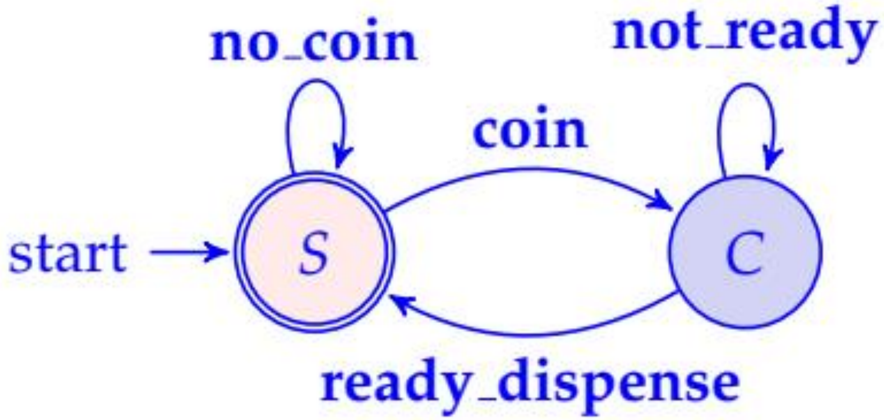
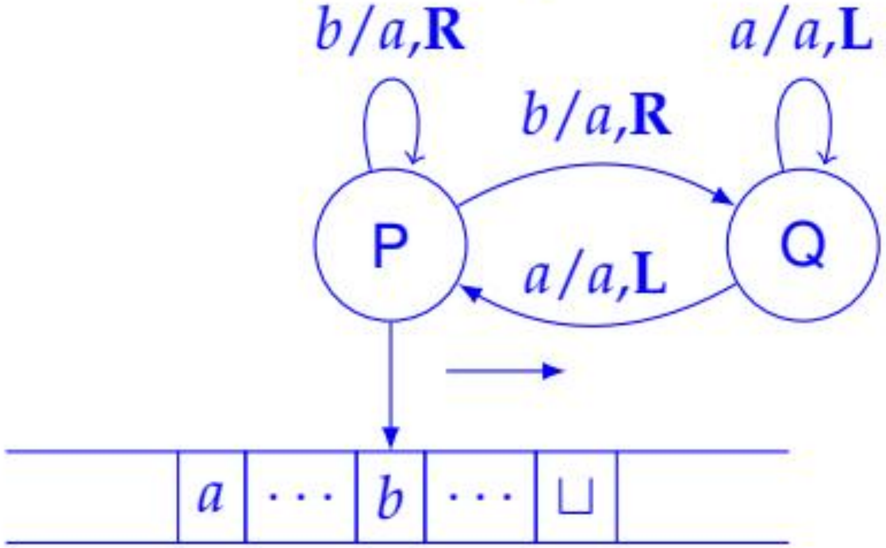
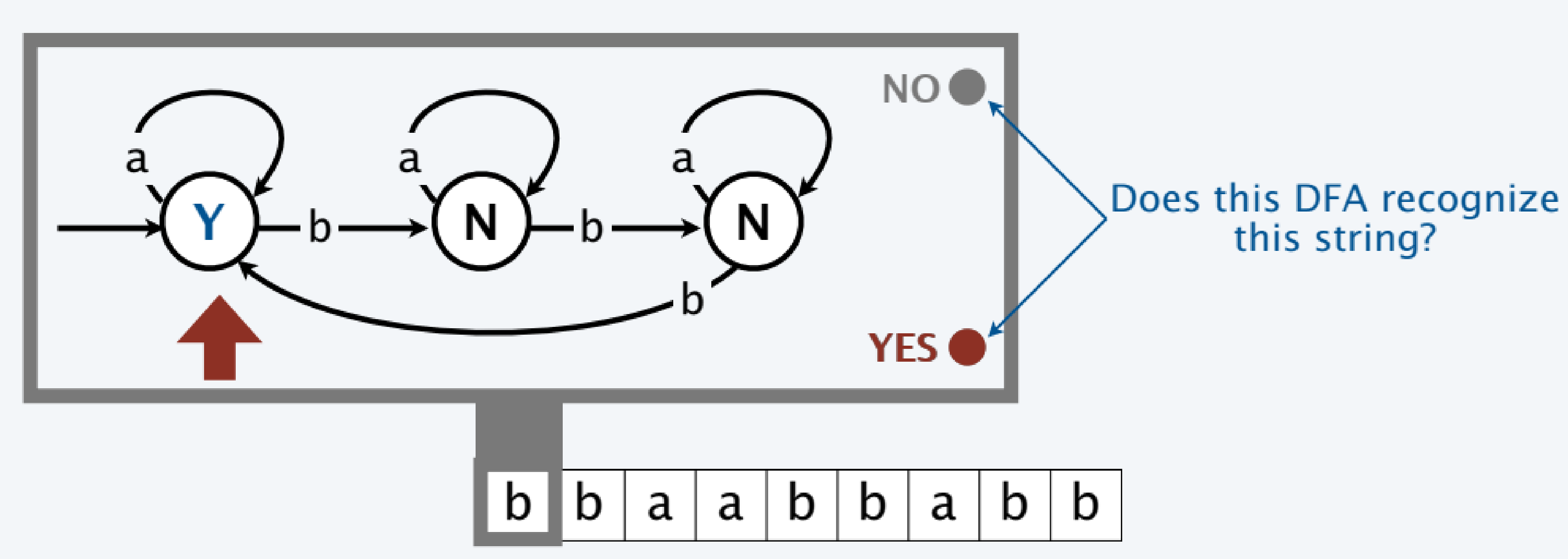
Turing Makinalar

Finite instruction machine with finite memory (Finite State Automata)

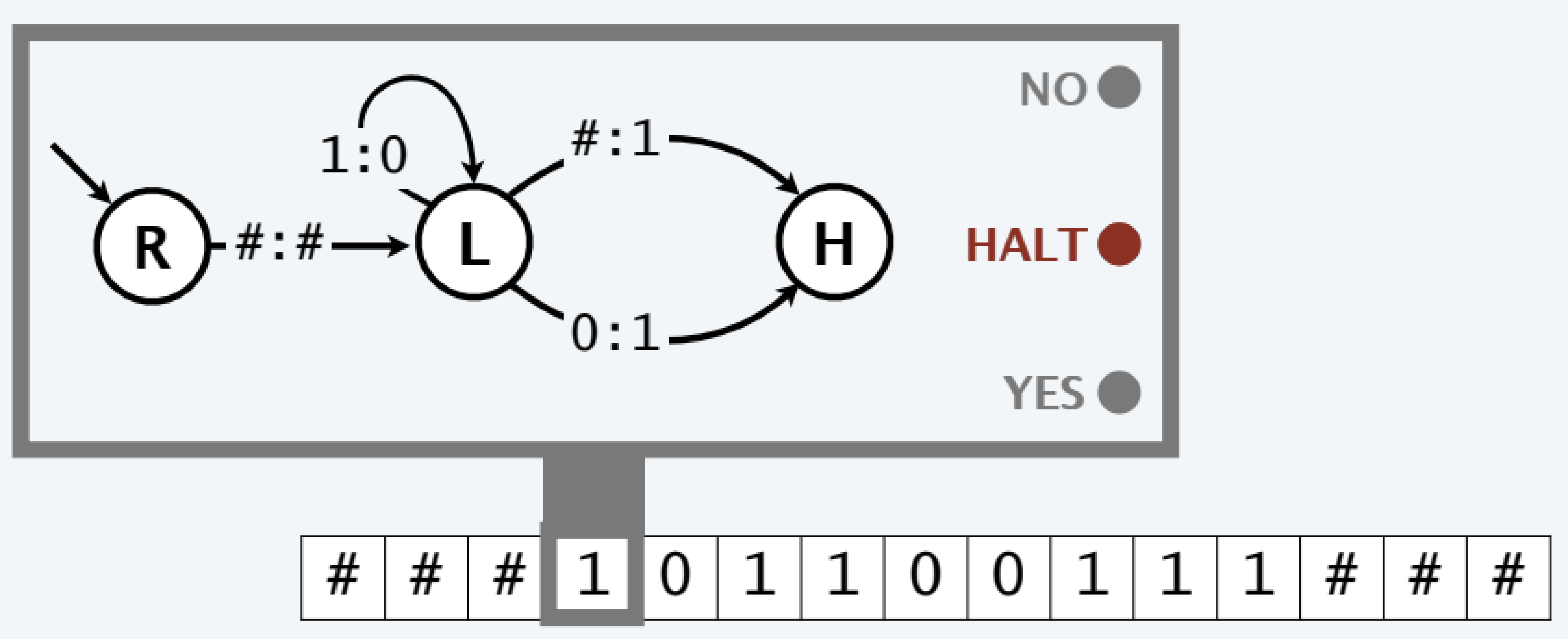


Finite instruction machine with unbounded memory (Turing machine)





NO
YES
a
a
a
Y
N
N
b
b
b
Does this DFA recognize this string?
b b a a b b a b b



NO
HALT
YES
1:0
#:1
R
#:#
L
H
0:1
1 0 1 1 0 0 1 1 1 # #

DFAs vs TMs

Similarities

- Simple model of computation.
- Input on tape is a finite string with symbols from a finite alphabet.
- Finite number of states.
- State transitions determined by current state and input symbol.

Differences

DFAs

- Can read input symbols from the tape.
- Can only move tape head to the right.
- Tape is finite (a string).
- One step per input symbol.
- Can *recognize* (turn on "YES" or "NO").

TMs

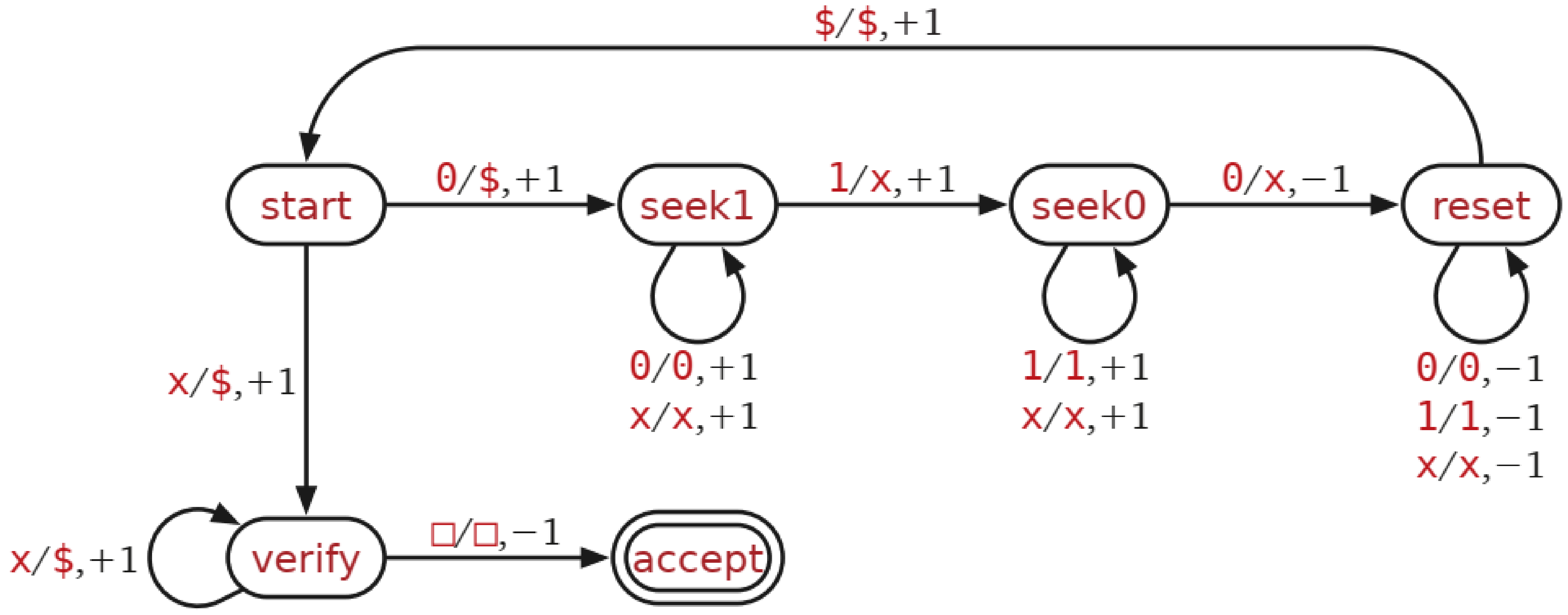
- Can read from or write onto the tape.
- Can move tape head either direction.
- Tape does not end (either direction).
- No limit on number of steps.
- Can also *compute* (with output on tape).

$$L = \{0^n 1^n 0^n \mid n \geq 0\}$$

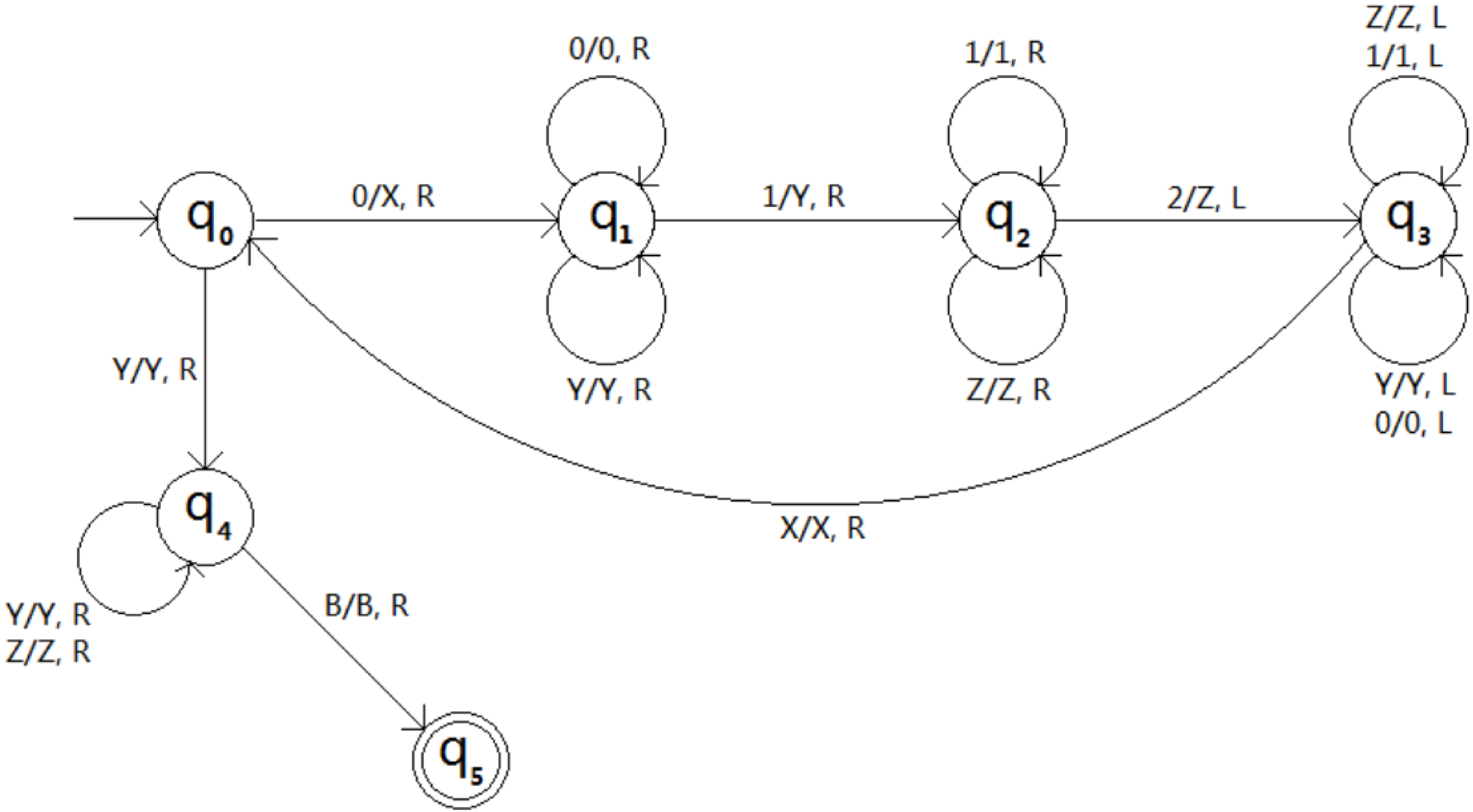
$$\Gamma = \{0, 1, \$, x, \square\}$$

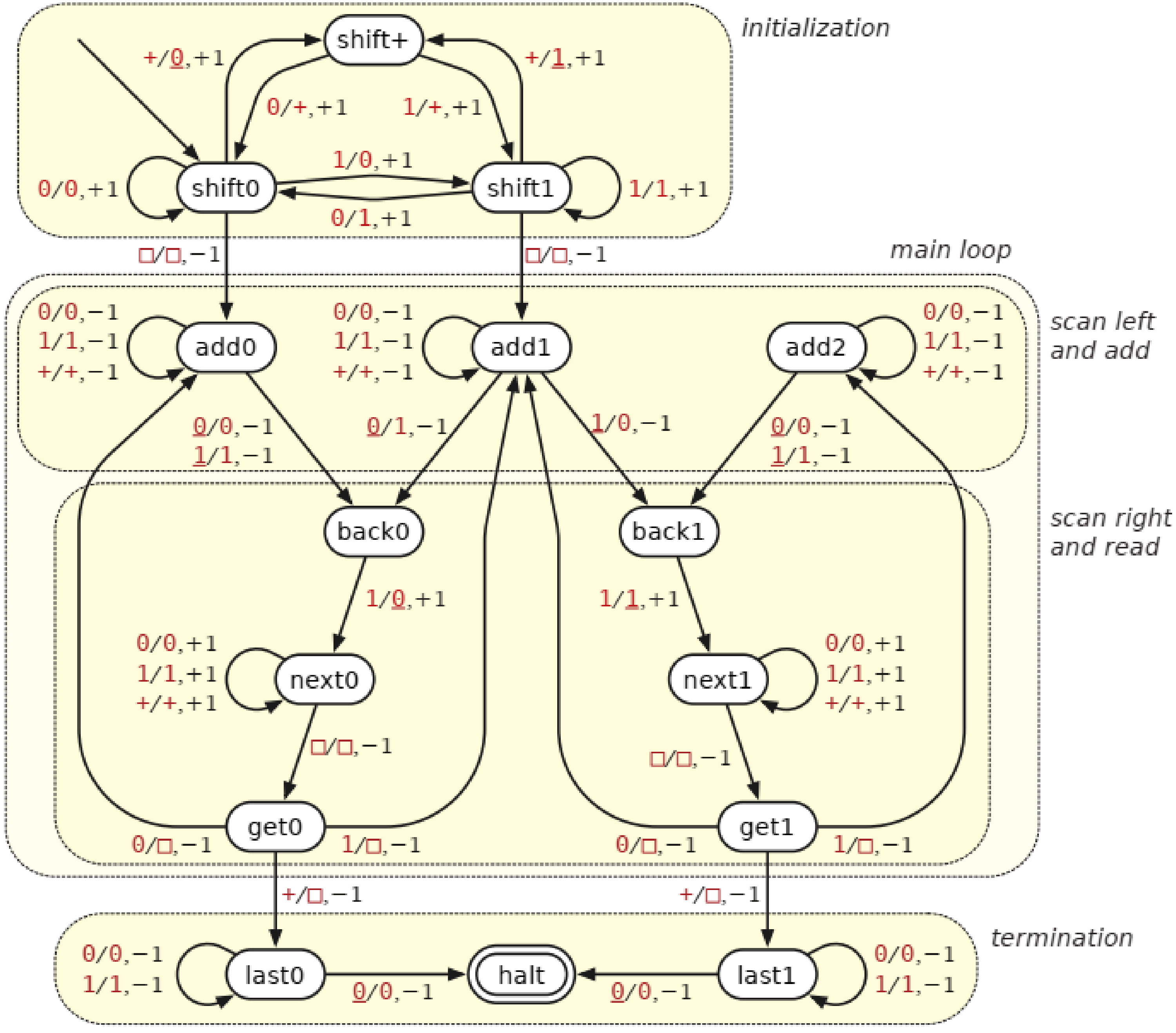
$$\Sigma = \{0, 1\}$$

$$Q = \{\text{start}, \text{seek1}, \text{seek0}, \text{reset}, \text{verify}, \text{accept}, \text{reject}\}$$



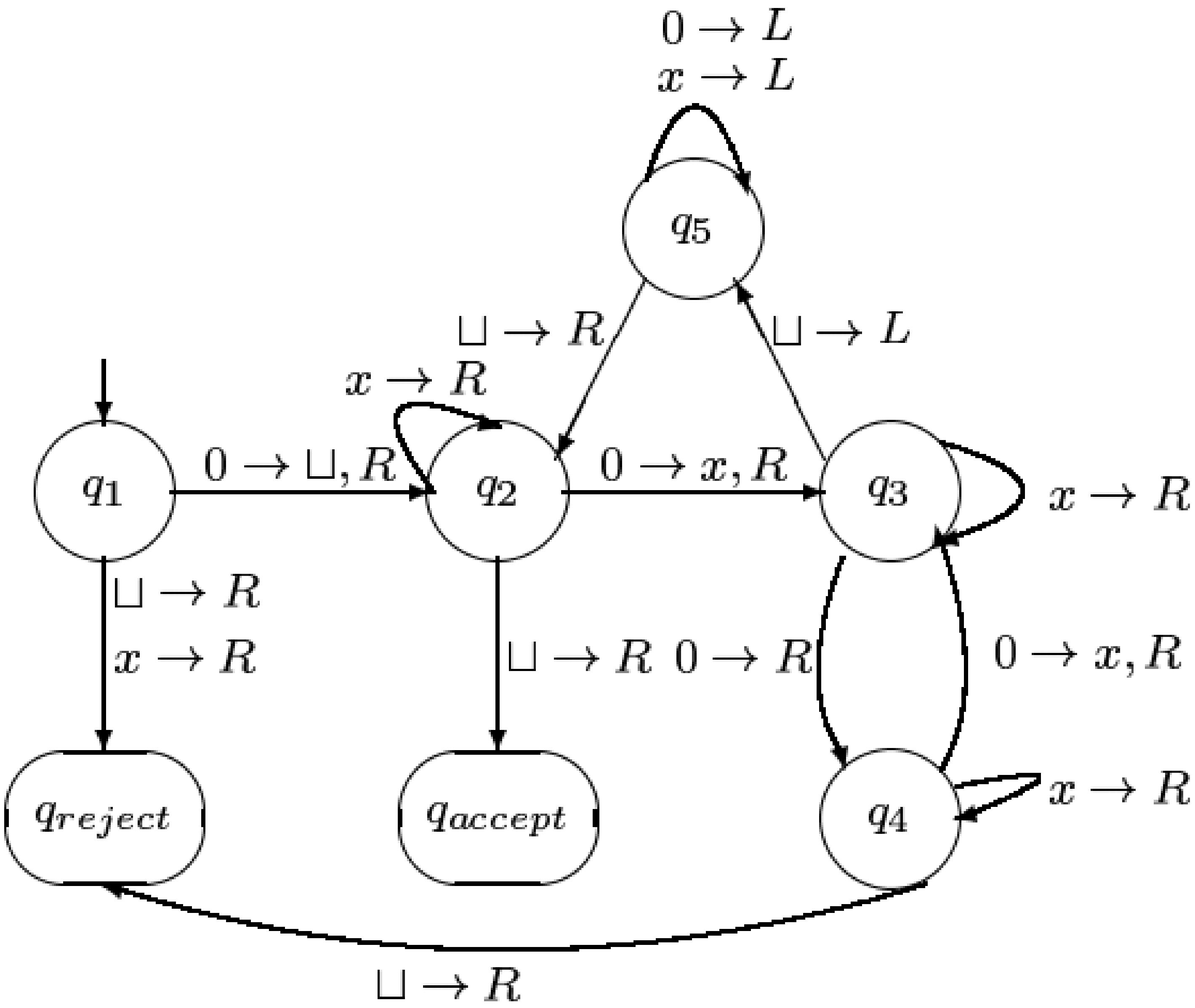
$$\{0^n 1^n 2^n : n \geq 1\}$$





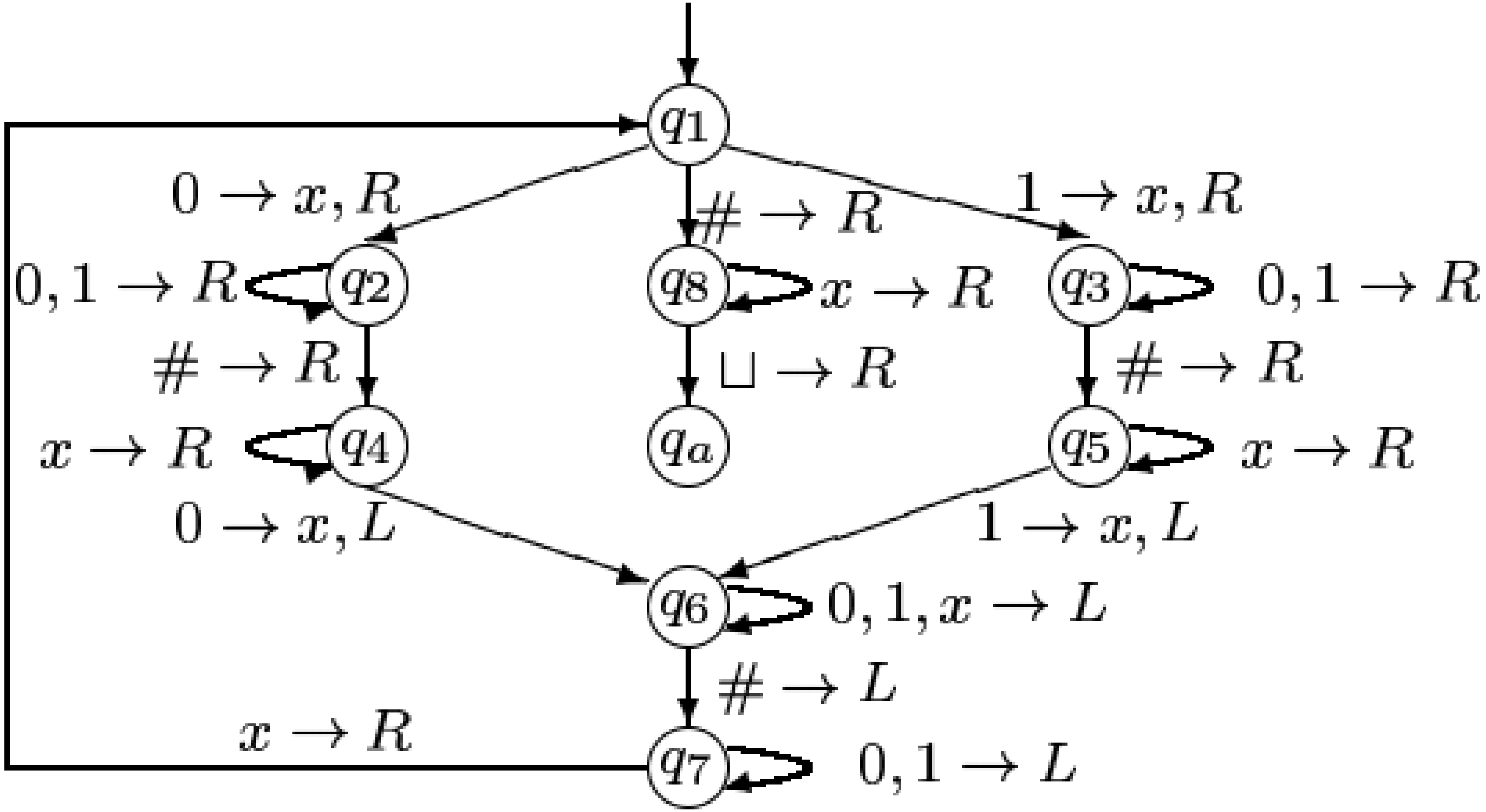
A Turing machine that adds two binary numbers of the same length.

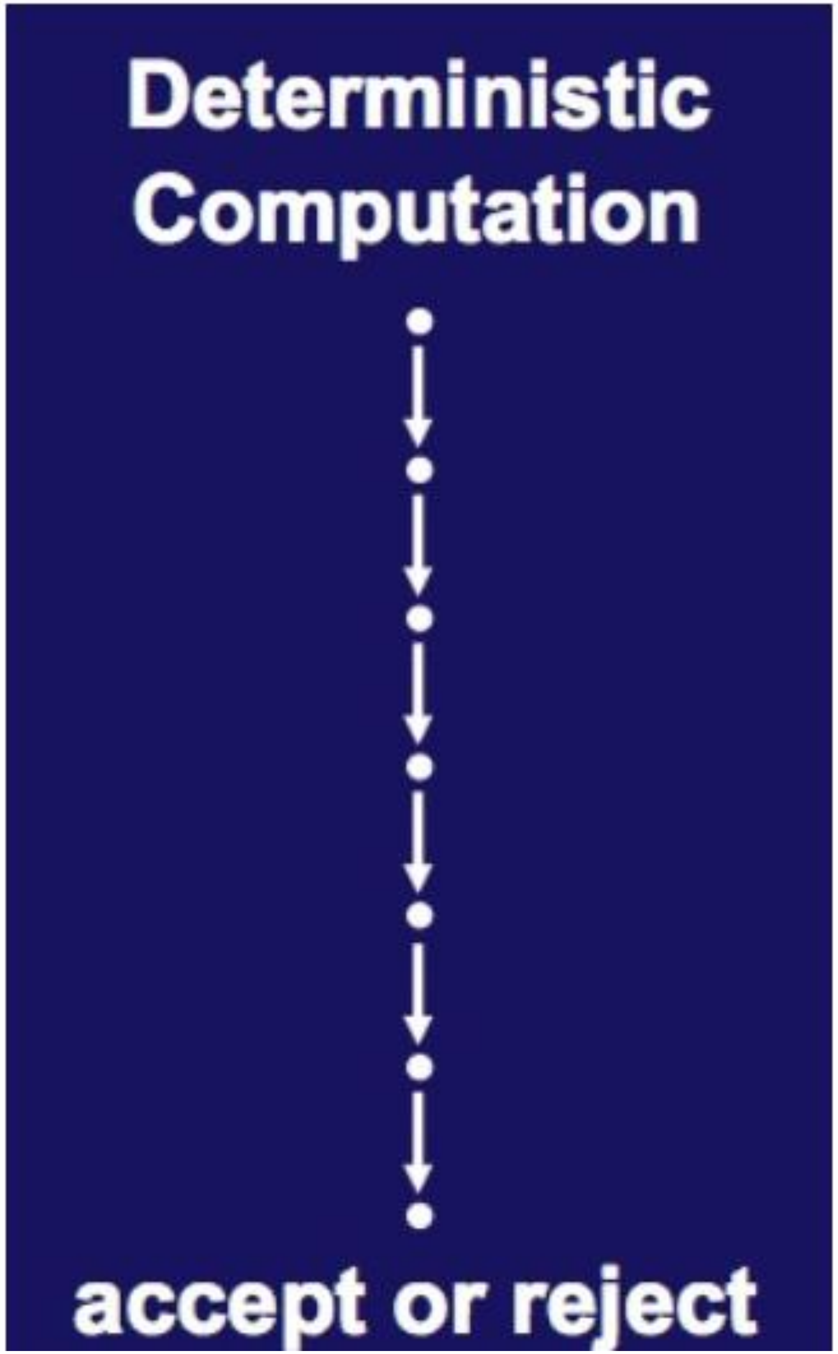
$$A = \{0^{2^n} | n \geq 0\}$$

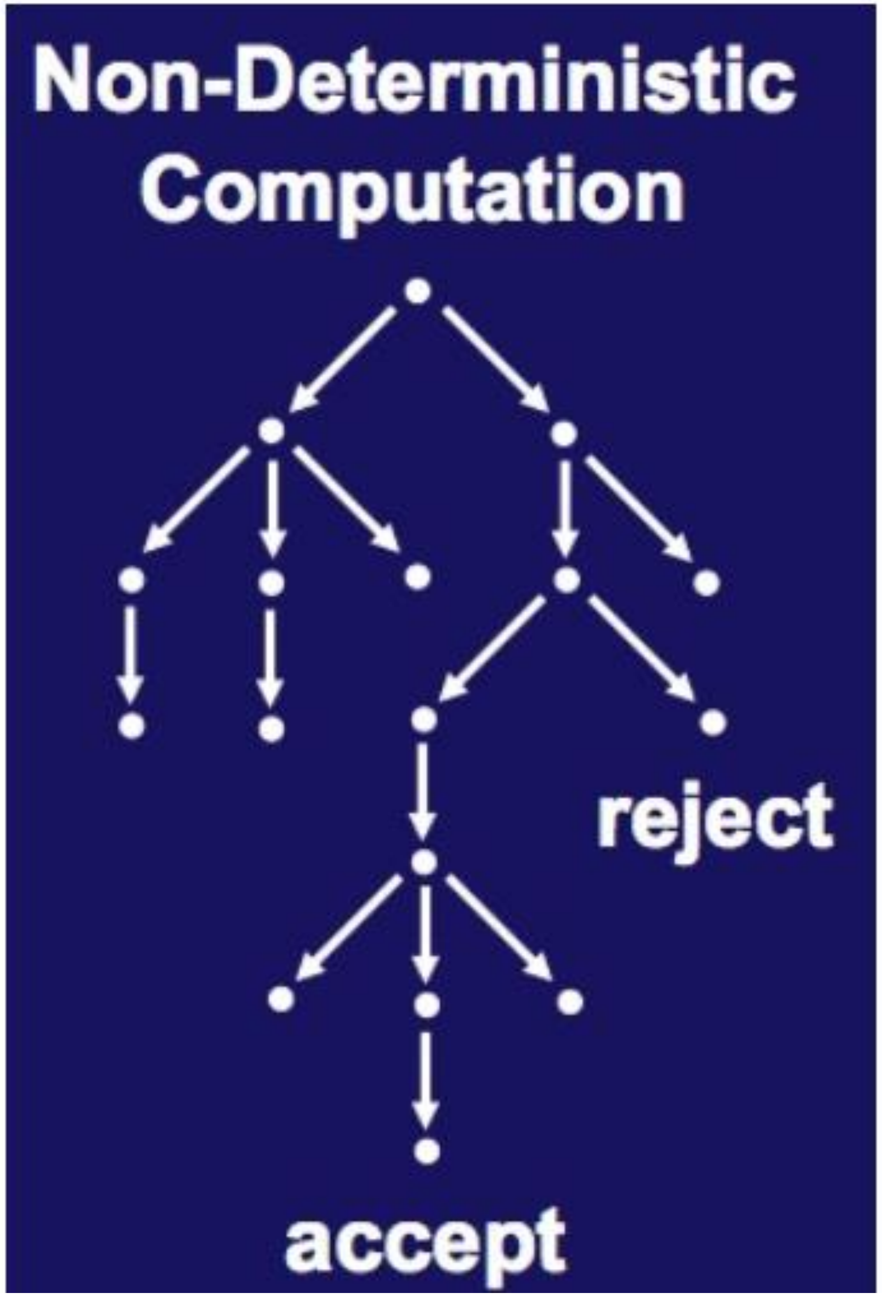

$0 \to L$
$x \to L$
q_5
$\sqcup \to R$
$\sqcup \to L$
$x \to R$
q_1
$0 \to \sqcup, R$
q_2
$0 \to x, R$
q_3
$x \to R$
$\sqcup \to R$
$x \to R$
$\sqcup \to R$
$0 \to R$
$0 \to x, R$
q_{reject}
q_{accept}
q_4
$x \to R$
$\sqcup \to R$

$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, q_a, q_r)$ is the TM that decides the language $B = \{w\#w | w \in \{0,1\}^*\}$

- $Q = \{q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_a, q_r\}$
- $\Sigma = \{0, 1, \#\}, \Gamma = \{0, 1, \#, x, \sqcup\}$




Deterministic Computation
accept or reject


Non-Deterministic Computation
reject
accept