TURING MACHINE

Lecturer Akib Zaman, Dept. of CSE, UIU

Turing Machine

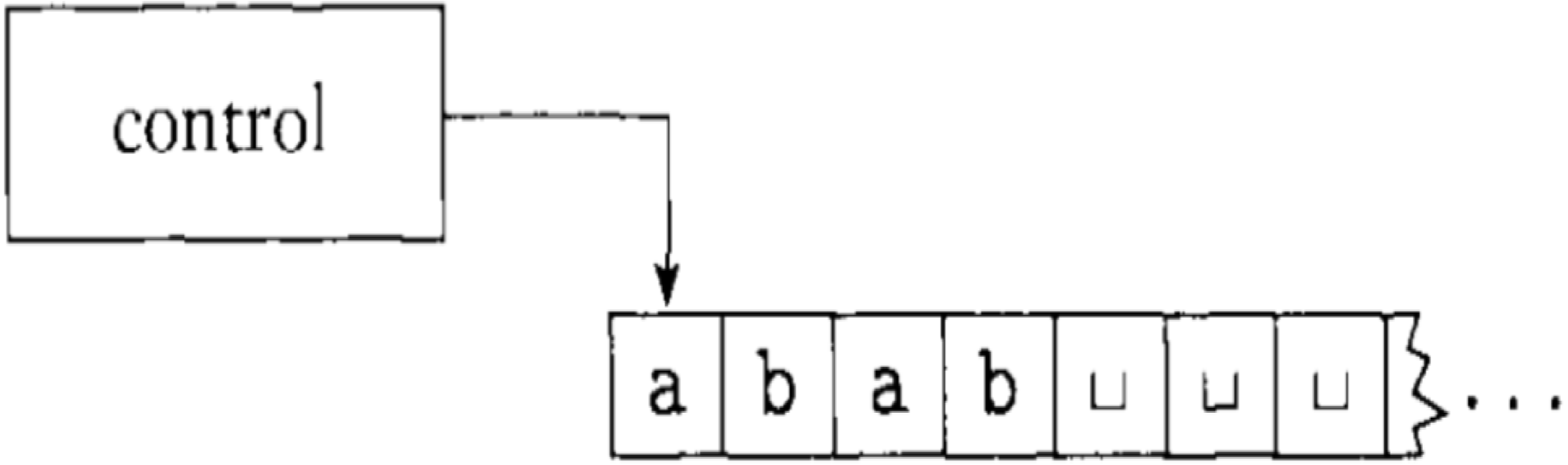
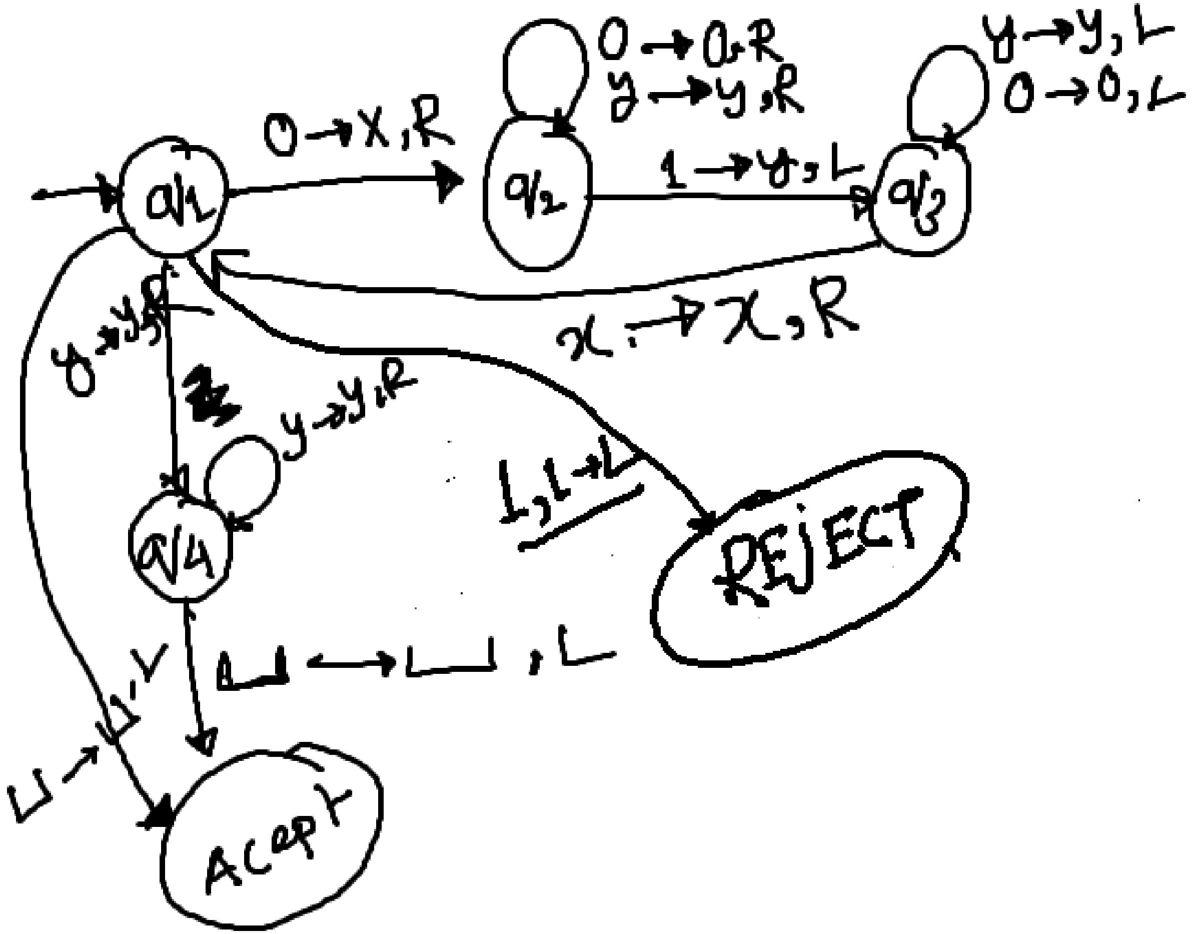


FIGURE 3.1
Schematic of a Turing machine

Differences between Pushdown Automaton and Turing Machine

- A Turing machine can both write on the tape and read from it.
- The read-write head can move both to the left and too the right.
- The tape is infinite.
- The special states for rejecting and accepting takes effect immediately.



Tape Traversal

$q_1\,0\,0\,1\,1$

$x\,q_2\,0\,1\,1$

$x\,0\,q_2\,1\,1$

$x\,q_3\,0\,y\,1$

$q_3\,x\,0\,y\,1$

$x\,q_1\,0\,y\,1$

$x\,x\,q_2\,y\,1$

$x\,x\,y\,q_2\,1$

$x\,x\,q_3\,y\,y$

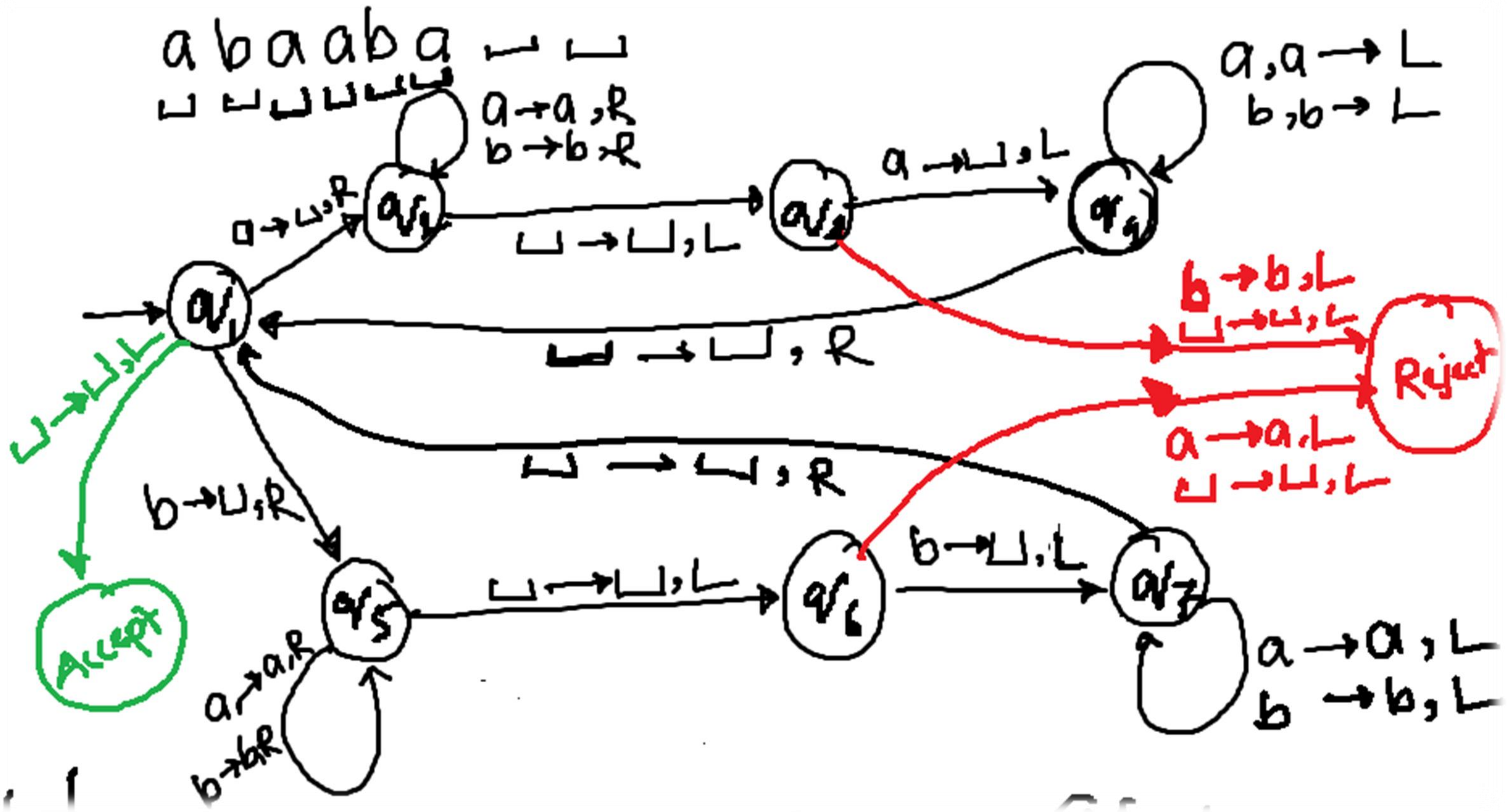
$x\,q_3\,x\,y\,y$

$x\,x\,q_1\,y\,y$

$x\,x\,y\,q_4\,y$

$x\,x\,y\,y\,q_4\,\sqcup$

$x\,x\,y\,y\,q_{accept}$


ababa ␣ ␣
a → a, R
b → b, R
a → ␣, R
␣ → ␣, L
a → ␣, L
a, a → L
b, b → L
b → b, L
␣ → ␣, L
␣ → ␣, R
Reject
a → a, L
␣ → ␣, L
␣ → ␣, L
␣ → ␣, R
b → ␣, R
Accept
␣ → ␣, L
b → ␣, L
a → a, L
b → b, L
a → a, R
b → b, R

$q_1abbaabba␣$

$␣q_2bbaabba␣$

$␣bq_2baabba␣$

$␣bbq_2aabba␣$

$␣bbaq_2abba␣$

$␣bbaaq_2bba␣$

$␣bbaabq_2ba␣$

$␣bbaabbq_2a␣$

$␣bbaabbaq_2␣$

$␣bbaabbq_3a␣$

$␣bbaabq_4b␣␣$

$␣bbaaq_4bb␣␣$

⋮

$␣q_4bbaabb␣␣$

$q_4␣bbaabb␣␣$

$␣q_1bbaabb␣␣$

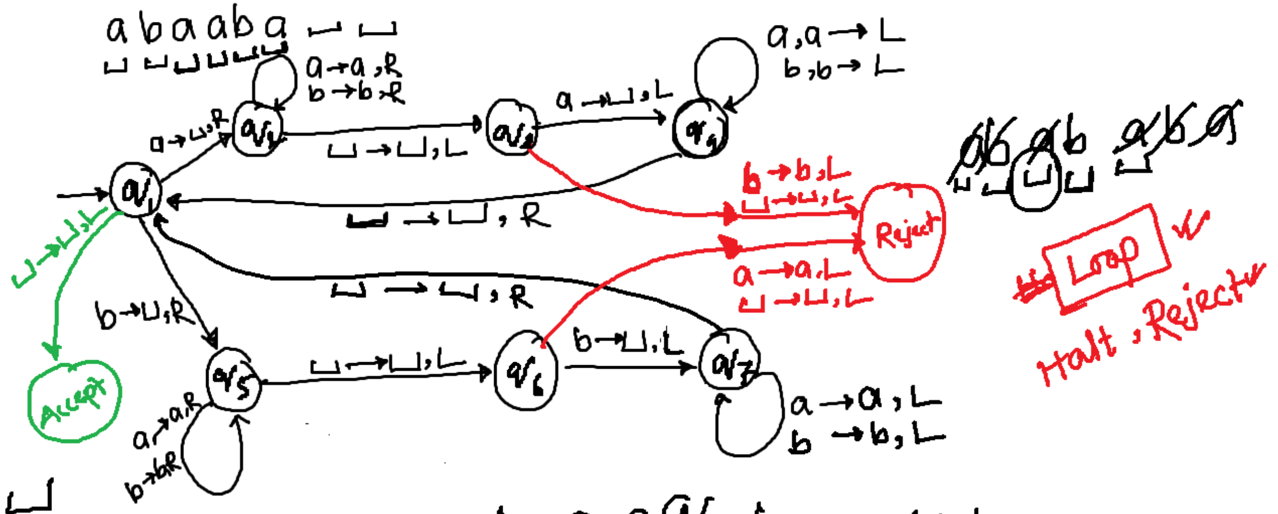
$␣␣q_5baabb␣␣$

⋮

$␣␣baabq_5b␣␣$

$␣␣baabbq_5␣␣$

$␣␣baabq_6b␣␣$



$␣␣baaq_7b␣␣␣$

⋮

$␣␣q_7baab␣␣␣$

$␣q_7␣baab␣␣␣$

$␣␣q_1baab␣␣␣$

Turing Machine (Continuation…)

- Turing Machine $\mathbf{M_1}$

$$B = \{w\#w \mid w \in \{0,1\}^*\}$$

Turing Machine (Continuation…)

- **M_1** algorithm is as follows:

$M_1 =$ "On input string w:

1. Zig-zag across the tape to corresponding positions on either side of the # symbol to check whether these positions contain the same symbol. If they do not, or if no # is found, *reject*. Cross off symbols as they are checked to keep track of which symbols correspond.
2. When all symbols to the left of the # have been crossed off, check for any remaining symbols to the right of the #. If any symbols remain, *reject*; otherwise, *accept*."

Turing Machine (Continuation…)

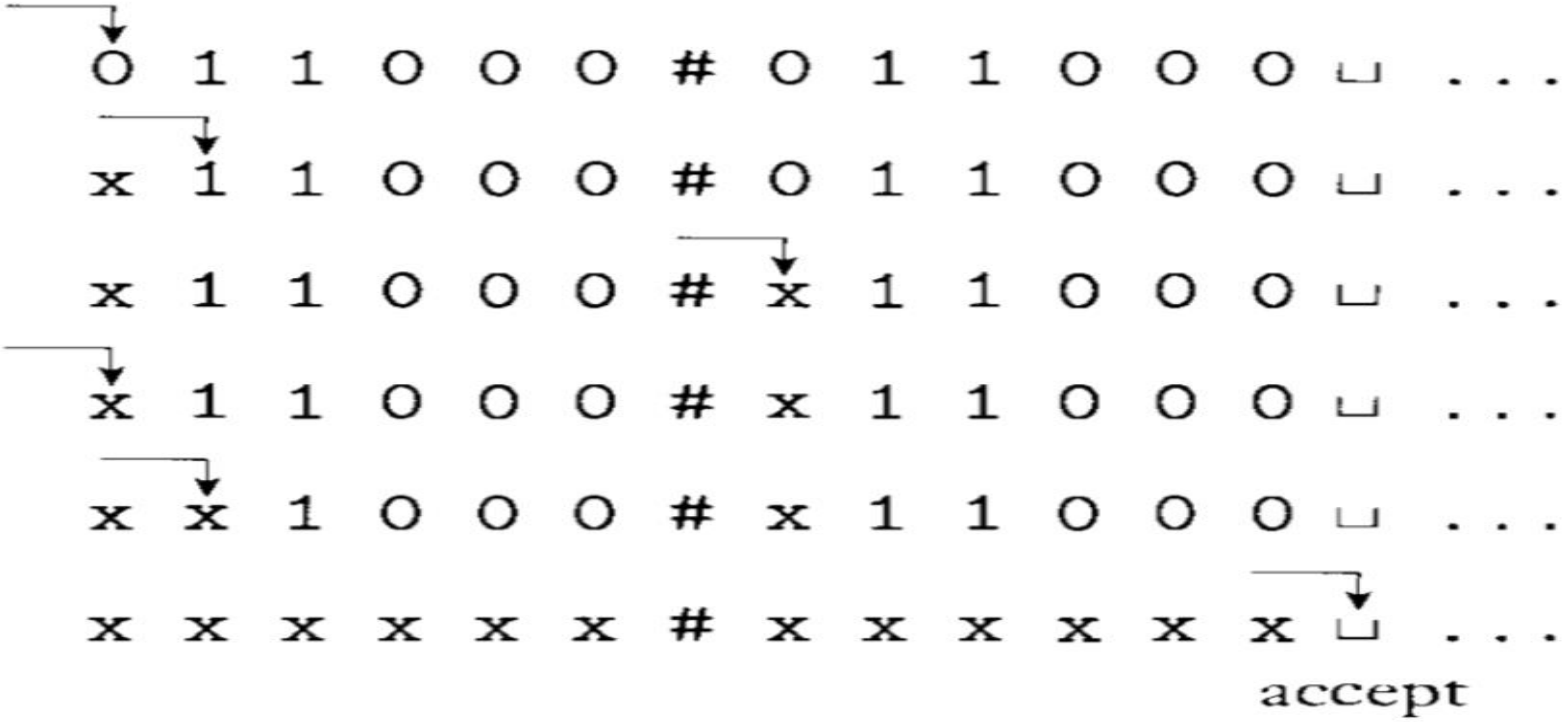


FIGURE 3.2
Snapshots of Turing machine M_1 computing on input 011000#011000

Formal Definition of a Turing Machine

DEFINITION 3.3

A ***Turing machine*** is a 7-tuple, $(Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$, where Q, Σ, Γ are all finite sets and

1. Q is the set of states,
2. Σ is the input alphabet not containing the ***blank symbol*** $\sqcup$,
3. Γ is the tape alphabet, where $\sqcup \in \Gamma$ and $\Sigma \subseteq \Gamma$,
4. $\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{\mathrm{L}, \mathrm{R}\}$ is the transition function,
5. $q_0 \in Q$ is the start state,
6. $q_{\text{accept}} \in Q$ is the accept state, and
7. $q_{\text{reject}} \in Q$ is the reject state, where $q_{\text{reject}} \neq q_{\text{accept}}$.

Configuration of a Turing Machine

- As a Turing Machine computes, changes occur in
  - Current state
  - Current tape contents
  - Current head location
- A setting of these three items called a **configuration** of a Turing Machine
- For a state q and two strings u and v over the tape alphabet, we write uqv for the configuration where
  - the current state is q
  - the current tape contents is uv and
  - the current head location is the first symbol of v
- The tape contains only blanks following the last symbol of v

Configuration of a Turing Machine (Continuation…)

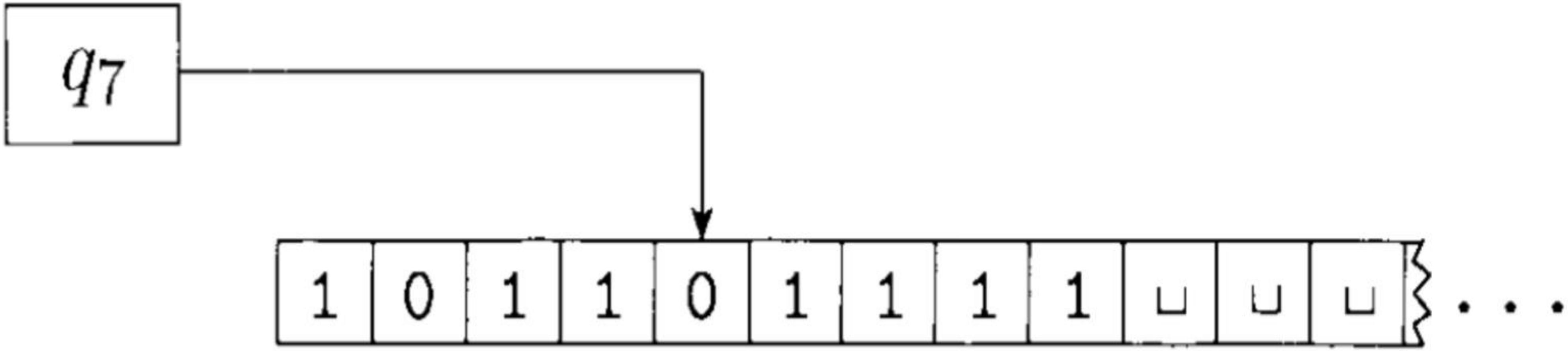


FIGURE 3.4
A Turing machine with configuration $1011q_701111$

Configuration of a Turing Machine (Continuation…)

- Configuration **C_1 yields configuration C_2** if TM can legally go from C_1 to C_2 in a single step

Suppose that we have a, b, and c in Γ, as well as u and v in Γ^* and states q_i and q_j. In that case $ua\,q_i\,bv$ and $u\,q_j\,acv$ are two configurations. Say that

$$ua\,q_i\,bv \quad \text{yields} \quad u\,q_j\,acv$$

if in the transition function $\delta(q_i, b) = (q_j, c, \mathrm{L})$. That handles the case where the Turing machine moves leftward. For a rightward move, say that

$$ua\,q_i\,bv \quad \text{yields} \quad uac\,q_j\,v$$

if $\delta(q_i, b) = (q_j, c, \mathrm{R})$.

Configuration of a Turing Machine (Continuation…)

- Special cases occur when the head is at one of the ends of the configuration
- For the left-hand end, the configuration $\mathbf{q_i bv}$ yields $\mathbf{q_j cv}$ if the transition is left moving and it yields $\mathbf{cq_j v}$ for the right moving transition
- For the right-hand end , the configuration $\mathbf{uaq_i}$ is equivalent $\mathbf{uaq_i} \sqcup$

Configuration of a Turing Machine (Continuation…)

- Start Configuration
- Accepting Configuration
- Rejecting Configuration
- Halting Configuration
- A TM M accepts input w if a sequence of configurations $\mathbf{C_1, C_2, \ldots, C_k}$ exists, where

1. C_1 is the start configuration of M on input w,
2. each C_i yields C_{i+1}, and
3. C_k is an accepting configuration.

Turing Recognizable and Turing Decidable Language

DEFINITION 3.5

Call a language ***Turing-recognizable*** if some Turing machine recognizes it.[1]

DEFINITION 3.6

Call a language ***Turing-decidable*** or simply ***decidable*** if some Turing machine decides it.[2]

Examples of Turing Machine

EXAMPLE 3.7 ....

Here we describe a Turing machine (TM) M_2 that decides $A = \{0^{2^n} \mid n \geq 0\}$, the language consisting of all strings of 0s whose length is a power of 2.

QUESTION:

- Difference Between TM and PDA.
- TM Recognizable and TM-Decidable Machine.

Examples of Turing Machine (Continuation…)

$M_2 =$ "On input string w:

1. Sweep left to right across the tape, crossing off every other 0.
2. If in stage 1 the tape contained a single 0, *accept*.
3. If in stage 1 the tape contained more than a single 0 and the number of 0s was odd, *reject*.
4. Return the head to the left-hand end of the tape.
5. Go to stage 1."

Examples of Turing Machine (Continuation…)

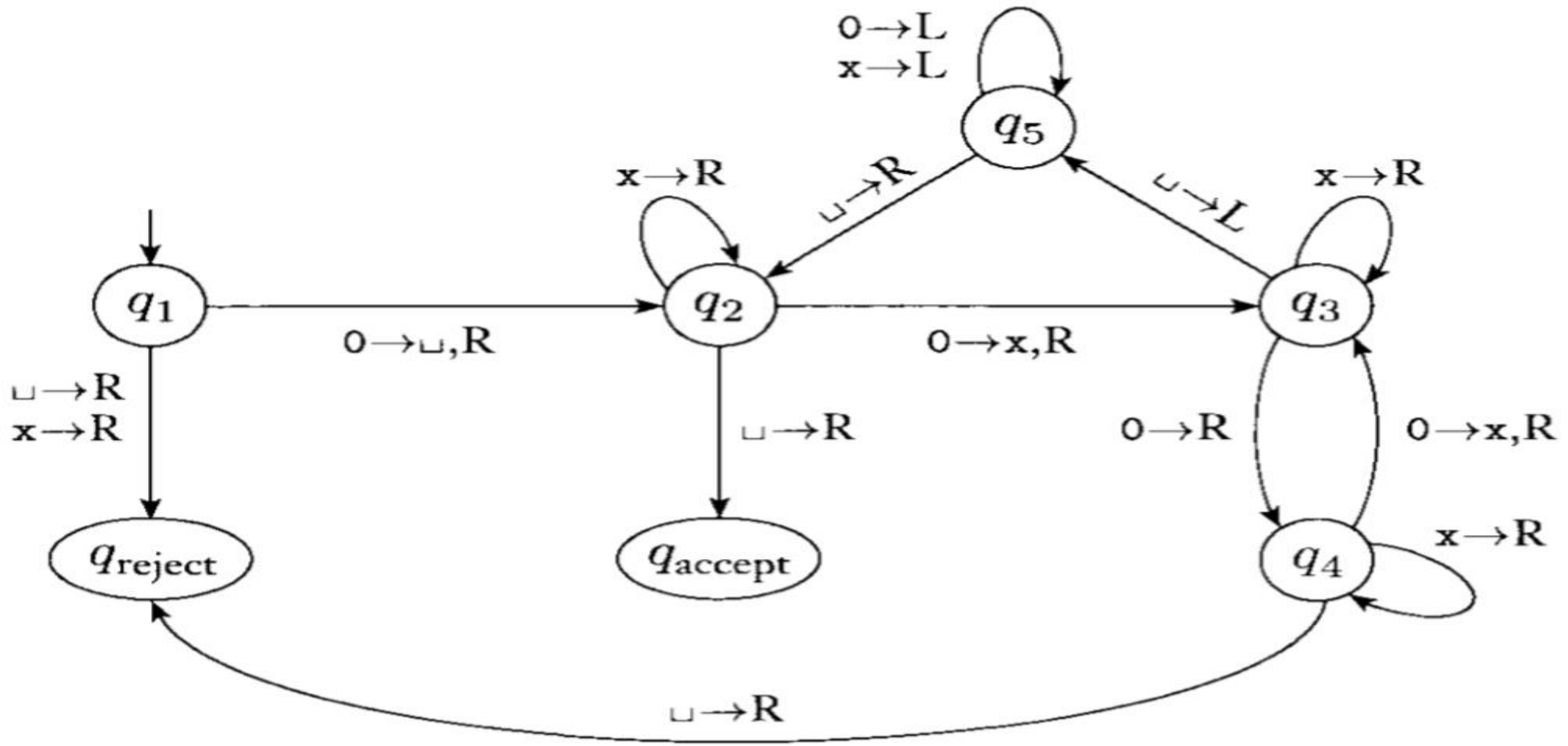


FIGURE 3.8
State diagram for Turing machine M_2

Examples of Turing Machine (Continuation…)

Now we give the formal description of $M_2 = (Q, \Sigma, \Gamma, \delta, q_1, q_{\text{accept}}, q_{\text{reject}})$:

- $Q = \{q_1, q_2, q_3, q_4, q_5, q_{\text{accept}}, q_{\text{reject}}\}$,
- $\Sigma = \{0\}$, and
- $\Gamma = \{0, \text{x}, \sqcup\}$.
- We describe δ with a state diagram (see Figure 3.8).
- The start, accept, and reject states are q_1, q_{accept}, and q_{reject}.

Examples of Turing Machine (Continuation…)

- **q_1 :**
  - Blank the leftmost 0
  - If starts with a blank, **reject**
  - If starts with X, **reject**
- **q_2 :**
  - Move right so long as X’s are encountered
  - If blank encountered while moving to the right, **accept** (The only accepting condition)
  - 0 replaced with X

Examples of Turing Machine (Continuation…)

- **q_3 :**
  - Skip the 0, this is the next 0 after the last replaced 0 in **q_2**
  - Skip all the X's
  - When a blank is found, we have reached the right end of the string, move left
- **q_4:**
  - Skip all the X's
  - 0 replaced with X, this 0 is after the last skipped 0 in **q_3**
  - A blank here will mean an odd number of 0's, **reject** the string
- **q_5 :**
  - Skip to left all the X's and 0's
  - When a blank is found, we have reached the left end of the string, move right

Examples of Turing Machine (Continuation…)

$q_1 0000$

$\sqcup q_2 000$

$\sqcup \mathrm{x} q_3 00$

$\sqcup \mathrm{x} 0 q_4 0$

$\sqcup \mathrm{x} 0 \mathrm{x} q_3 \sqcup$

$\sqcup \mathrm{x} 0 q_5 \mathrm{x} \sqcup$

$\sqcup \mathrm{x} q_5 0 \mathrm{x} \sqcup$

$\sqcup q_5 \mathrm{x} 0 \mathrm{x} \sqcup$

$q_5 \sqcup \mathrm{x} 0 \mathrm{x} \sqcup$

$\sqcup q_2 \mathrm{x} 0 \mathrm{x} \sqcup$

$\sqcup \mathrm{x} q_2 0 \mathrm{x} \sqcup$

$\sqcup \mathrm{xx} q_3 \mathrm{x} \sqcup$

$\sqcup \mathrm{xxx} q_3 \sqcup$

$\sqcup \mathrm{xx} q_5 \mathrm{x} \sqcup$

$\sqcup \mathrm{x} q_5 \mathrm{xx} \sqcup$

$\sqcup q_5 \mathrm{xxx} \sqcup$

$q_5 \sqcup \mathrm{xxx} \sqcup$

$\sqcup q_2 \mathrm{xxx} \sqcup$

$\sqcup \mathrm{x} q_2 \mathrm{xx} \sqcup$

$\sqcup \mathrm{xx} q_2 \mathrm{x} \sqcup$

$\sqcup \mathrm{xxx} q_2 \sqcup$

$\sqcup \mathrm{xxx} \sqcup q_{\mathrm{accept}}$

Examples of Turing Machine (Continuation…)

- **Example 3.9**: Turing Machine $\mathbf{M_1}$

$$B = \{w\#w \mid w \in \{0,1\}^*\}$$

Examples of Turing Machine (Continuation…)

- Formal definition of TM **M_1**

$M_1 = (Q, \Sigma, \Gamma, \delta, q_1, q_{accept}, q_{reject})$

$Q = \{q_1, \ldots, q_{14}, q_{accept}, q_{reject}\}$,

$\Sigma = \{0,1,\#\}$, and $\Gamma = \{0,1,\#,x,\sqcup\}$.

We describe δ with a state diagram (see the following figure).

The start, accept, and reject states are q_1, q_{accept}, and q_{reject}.

Examples of Turing Machine (Continuation…)

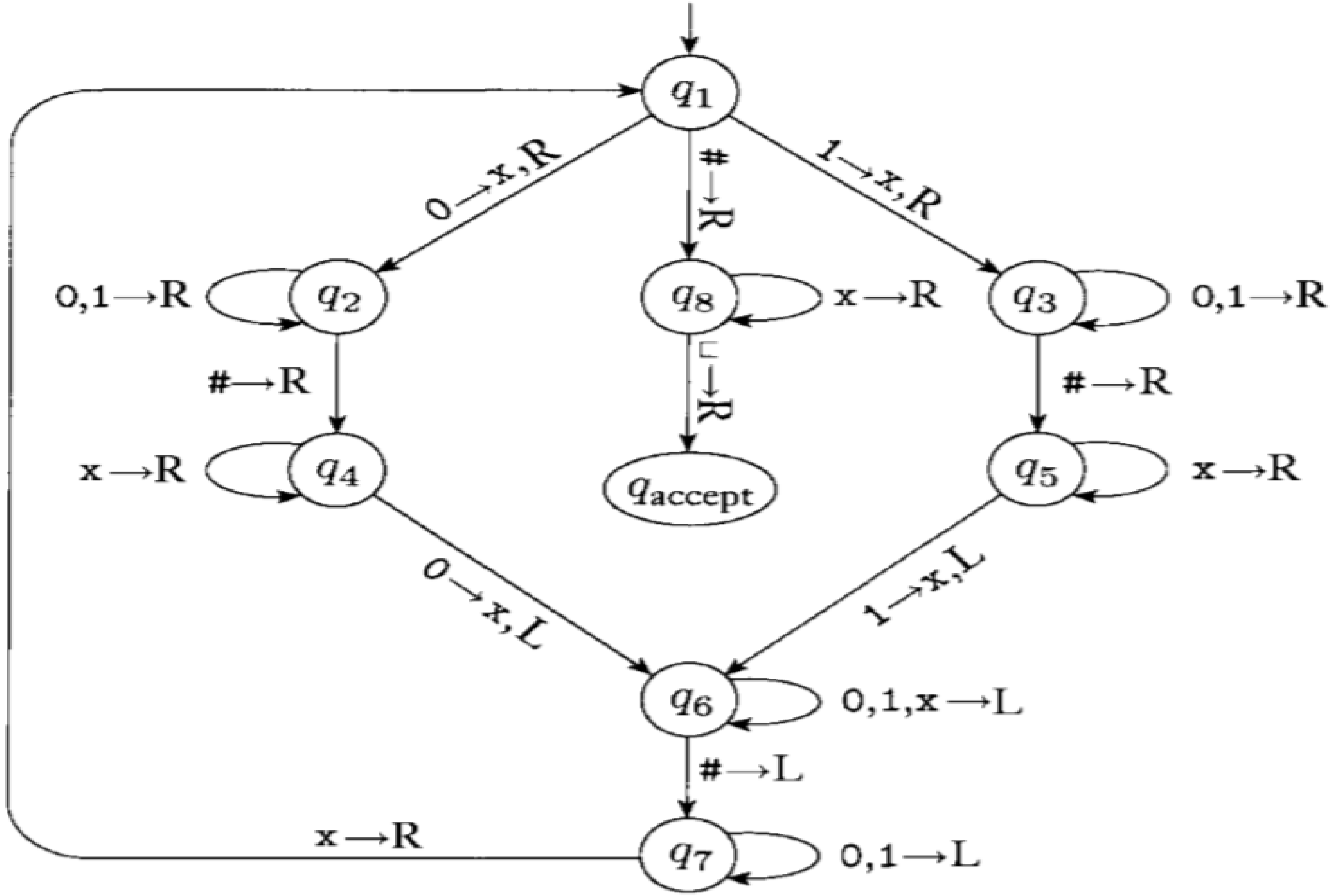


FIGURE 3.10
State diagram for Turing machine M_1

Examples of Turing Machine (Continuation…)

- **Example 3.11**: Turing Machine $\mathbf{M_3}$

$$C = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i \times j = k \text{ and } i, j, k \geq 1\}$$

Examples of Turing Machine (Continuation…)

$M_3 =$ "On input string w:

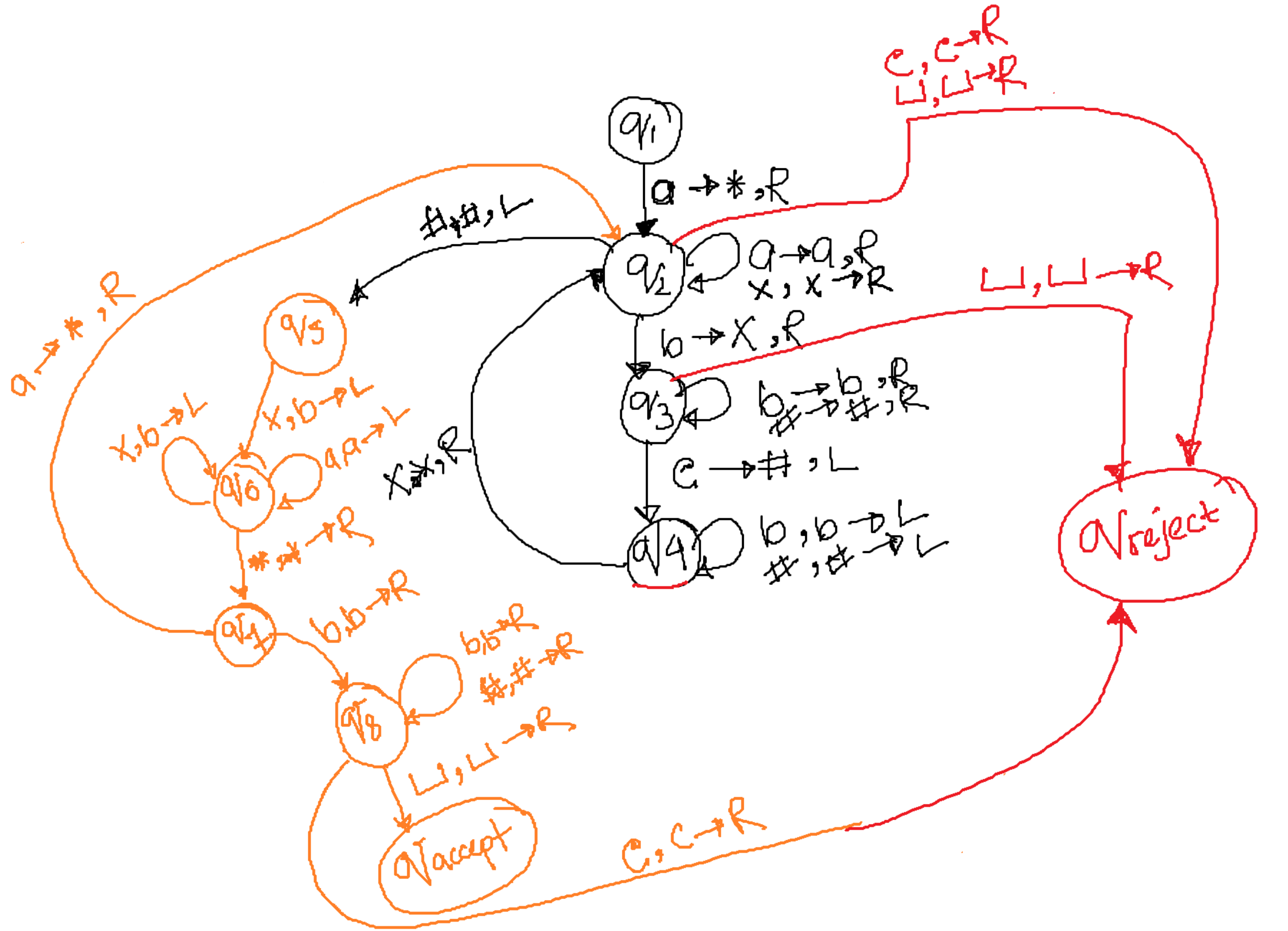
1. Scan the input from left to right to determine whether it is a member of $\mathtt{a}^+\mathtt{b}^+\mathtt{c}^+$ and *reject* if it isn't.
2. Return the head to the left-hand end of the tape.
3. Cross off an a and scan to the right until a b occurs. Shuttle between the b's and the c's, crossing off one of each until all b's are gone. If all c's have been crossed off and some b's remain, *reject*.
4. Restore the crossed off b's and repeat stage 3 if there is another a to cross off. If all a's have been crossed off, determine whether all c's also have been crossed off. If yes, *accept*; otherwise, *reject*."

Examples of Turing Machine (Continuation…)

- Modification of $\mathbf{M_3}$ algorithm:
  - Mark left hand end of the input by ∐ symbol
  - Use three different symbols for marking a, b and c on the tape
    - Use of ***** symbol for marking **a**
    - Use of **X** symbol for marking **b**
    - Use of **#** symbol for marking **c**

Examples of Turing Machine (Continuation…)

Fig: State Diagram for TM M3 (modified)



q1
a → *, R
q2
a → a, R
x, x → R
b → X, R
q3
b → b, R
→ #, R
c → #, L
q4
b, b → L
#, # → L
#, #, L
X, X, R
q5
X, b → L
q6
a, a → L
*, * → R
q7
b, b → R
q8
#, # → R
⊔, ⊔ → R
⊔, ⊔
q accept
q reject
c, c → R
⊔, ⊔ → R

Examples of Turing Machine (Continuation…)

EXAMPLE 3.12

Here, a TM M_4 is solving what is called the *element distinctness problem*. It is given a list of strings over $\{0,1\}$ separated by #s and its job is to accept if all the strings are different. The language is

$$E = \{\#x_1\#x_2\# \cdots \#x_l \mid \text{each } x_i \in \{0,1\}^* \text{ and } x_i \neq x_j \text{ for each } i \neq j\}.$$

Machine M_4 works by comparing x_1 with x_2 through x_l, then by comparing x_2 with x_3 through x_l, and so on. An informal description of the TM M_4 deciding this language follows.

Examples of Turing Machine (Continuation…)

M_4 = "On input w:

1. Place a mark on top of the leftmost tape symbol. If that symbol was a blank, *accept*. If that symbol was a #, continue with the next stage. Otherwise, *reject*.
2. Scan right to the next # and place a second mark on top of it. If no # is encountered before a blank symbol, only x_1 was present, so *accept*.
3. By zig-zagging, compare the two strings to the right of the marked #s. If they are equal, *reject*.
4. Move the rightmost of the two marks to the next # symbol to the right. If no # symbol is encountered before a blank symbol, move the leftmost mark to the next # to its right and the rightmost mark to the # after that. This time, if no # is available for the rightmost mark, all the strings have been compared, so *accept*.
5. Go to Stage 3."