Introduction

In the previous chapters, we discussed two of the major approaches to the modelling of computation viz. the automata approach and the grammatical approach. Under automata approach, we discussed finite automata and pushdown automata while under the grammatical approach, we discussed context-free languages.

Further, we made the following observations:

a. A finite automaton computational model is computationally equivalent to a regular language model.

b. A pushdown automaton model is computationally equivalent to a context-free language model.

c. A pushdown automaton model is more powerful, when compared to a finite automaton model in the sense that every language accepted by a finite automaton is also recognised by the pushdown automaton. However, there are languages, viz. the language $\{a^n b^n : n \in N\}$, which are recognised by pushdown automata but not by finite automata.

d. There are languages, including the language $\{a^n b^n c^n : n \in N\}$, which are not accepted even by pushdown automata.

This prompts us to introduce a more powerful model of automata approach, which recognises more languages than a pushdown automaton model, called Turing machine. It was first proposed by *Alan Turing* in 1936 and was designed to meet the following objectives:

a. They should be automata, i.e., their construction and function should be in the same general spirit as the other computational models.

b. They should be as simple as possible, to describe, to define formally and to reason about.

c. They should be as general as possible, in terms of the computations they can carry out.

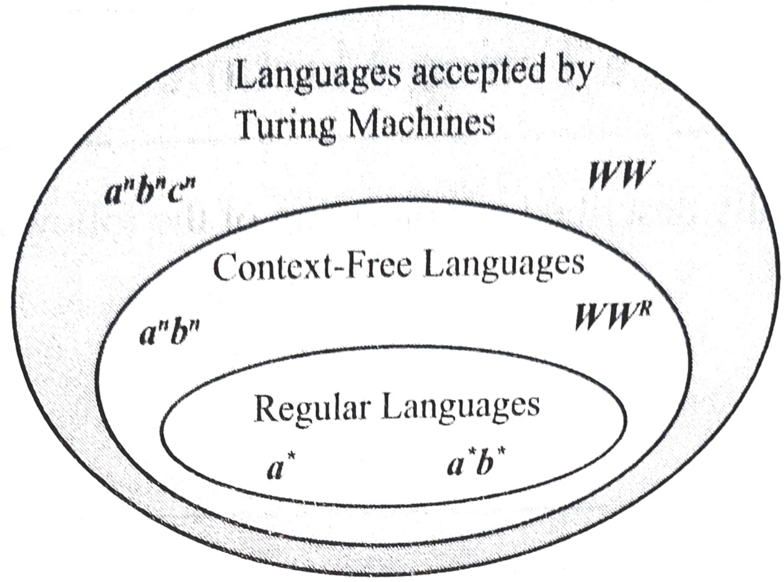


Figure 13. *The Language Hierarchy*

Definition:

There exists the following equivalent definitions for the concept of T.M:

a. Turing machines, are simple abstract computational devices intended to help investigate the extent and limitations of what can be computed.
b. A turing machine is a kind of state machine. At any time, the machine is in any one of the finite number of states. Instructions for a turing machine include the specification of conditions, under which the machine will make transitions from one state to other.

Alan Mathison Turing (1912–1954) was a British mathematician and cryptographer. He went to King's College, Cambridge in 1931 to study Mathematics. Turning graduated from Cambridge in Mathematics in 1934 and was a fellow at Kings for two years, during which period he wrote his now famous paper published in 1937–*On Computable Numbers, with an Application to the* Entscheidungs problem.

Turing is considered to be one of the fathers of modern computer science. He provided an influential formalisation of the concept of algorithm and computation–the Turing machine. He formulated the now widely accepted 'Turing' version of the Church-Turing thesis, that is any practical computing model has either the equivalent or a subset of the capabilities of a Turing machine. During World War II, he was the director of the Naval Enigma Hut at Bletchley Park for sometime and remained as the chief cryptanalyst of the Naval Enigma effort, throughout the war. After the war, he designed one of the earliest electronic programmable digital computers at the National Physical Laboratory and, shortly thereafter, actually built another early machine at the University of Manchester. He also, amongst many other things, made significant and characteristically provocative contributions to the discussion "Can machines think?".

13.1 Components of a Turing Machine

A Turing machine is usually described as consisting of the following three components.

- tape
- head
- control unit.

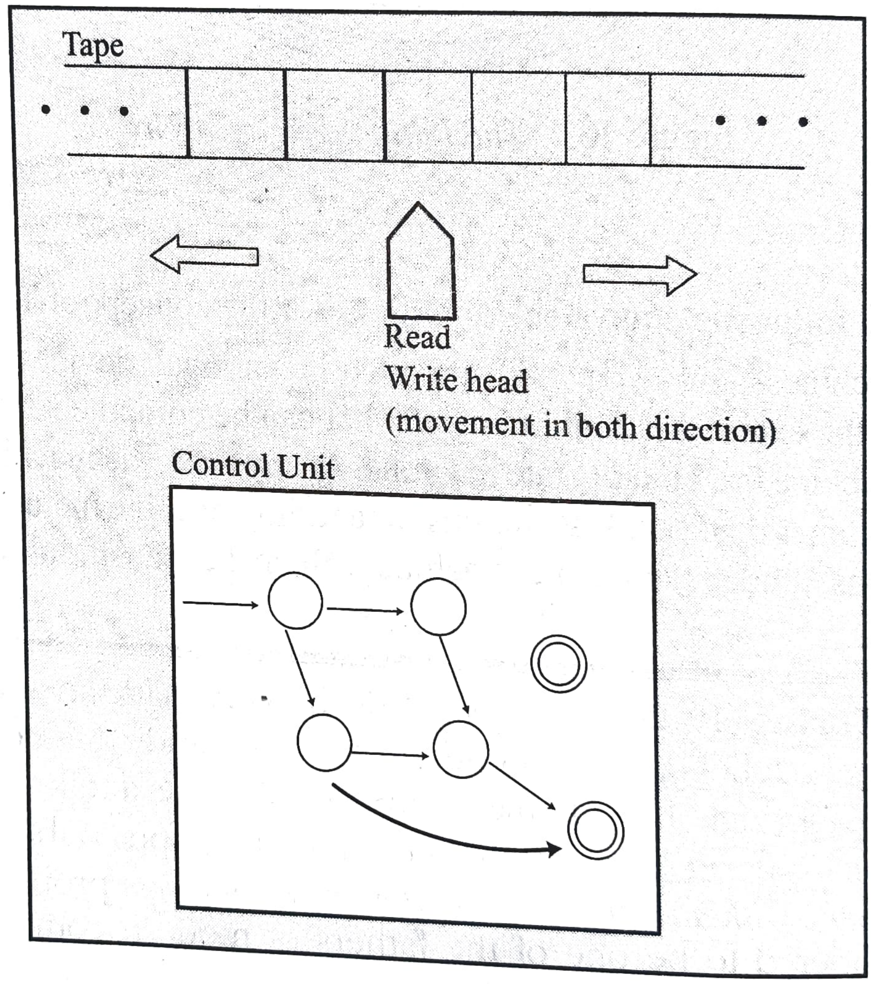


Figure 13.1. *Components of a Turing Machine*

a. **TAPE**

A tape is divided into a sequence of numbered cells or squares, one next to other. Each cell contains a symbol from some finite alphabet. The alphabet contains a blank symbol (B) and one or more other symbols. The set of symbols of the tape is denoted by Γ. The tape is assumed to be arbitrarily extensible to the left as well as to the right. This implies that the Turing machine is always supplied with as *much tape as it needs for its computation*. Cells that have not been written before are assumed to be filled with the blank symbol.

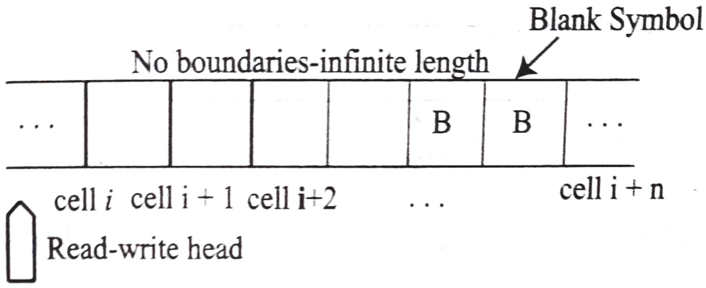


Figure 13.2. *The Tape with Cell i Indicating the Start Cell for the Current Computation*

b. **HEAD**

- A tape head, is always stationed at one of the tape cells and provides communication for the interaction between the tape and the control unit.
- In a single step, a tape head reads the contents of a cell on the tape (reads a symbol), replaces it with some other character (writes a symbol) and repositions itself to the next cell to the right or to the left of the one it has just read or does not move (moves left or right or does not move).
  This course of action is called the *move of a Turing machine.*
- At the beginning of the processing, the tape head always begins by reading the input in cell i. The head can never move left from the cell i and if it is given an order to do so, the machine crashes.

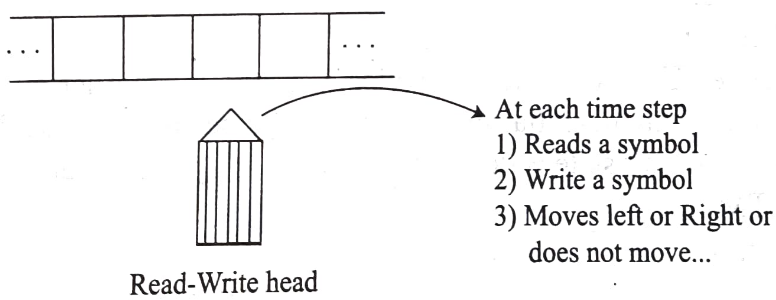


Figure 13.3. *Head*

EXAMPLE 13.1.1:

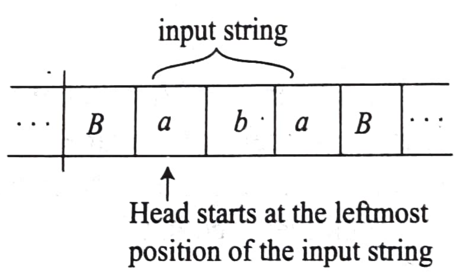


Figure 13.4. *Tape at Time-0*

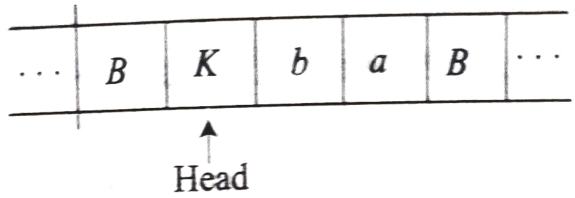


Figure 13.5. *Tape at time-1 After the Action: 1. Reads a 2. Writes K 3. Moves Right*

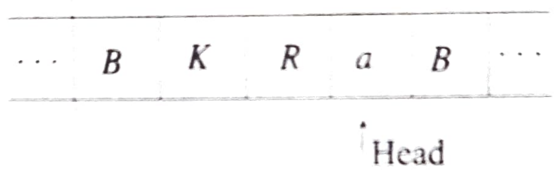


Figure 13.6. *Tape at Time-2 After the Action: 1. Reads b 2. Writes R 3. Moves Right*

c. **Control Unit**

The reading from the tape or writing into the tape is determined by the control unit. It contains a finite set of states Q. The states are categorised into three viz.,

(i) *The Initial State:* It is the state of control, just at the time when TM starts its operations. The initial state is denoted by q_0 and $q_0 \in Q$.
(ii) *The Halt State:* This is the state in which TM stops all further operations. The halt state is distinct from the initial state i.e., for a TM the halt and the initial states cannot be the same. The halt state is denoted by h and $h \sqsubset Q$. There can be one or more halt states in a TM.
(iii) *Other states.*

13.1.1 *Tape and Head of a FA/PDA Vs. Tape and Head of a TM*

The following are the differences in the roles of the tape and the tape head of a FA/PDA and the tape and head of a TM:

a. The cells of the tape of a FA or a PDA are only read/scanned but are never changed/written into, whereas the cells of the tape of a TM may be written also.
b. The tape head of a FA or a PDA always moves from left to right. However, the tape head of a TM can move in both the directions.

From the above two differences, it is clear that for a FA or a PDA, the information in the tape cells which is already scanned does not play any role in deciding the future moves of the automaton. On the other hand, in the case of a TM, the information contents of all the cells (including the ones earlier scanned) play a role in deciding the future moves.

13.1.2 *Halt State of a TM Vs. Set of Final States of a FA/PDA*

- A TM on entering the halt state stops making moves and whatever string is there on the tape will be taken as the output, irrespective of whether the position of head is at the end or in the middle of the string on the tape.
- If a FA/PDA enters a final state while scanning a symbol of the input tape, it can still go ahead with the repeated activities of moving to the right, scanning the symbol under the head and entering a new state etc. Further, the portion of a string from left to the symbol under the tape head is accepted, if the state is a final state, and is rejected if it is not.

13.2 Description of a Turing Machine

There are different ways to describe the task of a Turing machine:

13.2.1 *The Transition Diagram*

The turing machine can be represented using the transition diagram. For a directed graph, an *arc* going from a vertex (which corresponds to the state P) to the vertex that corresponds to the state q, and the also the *edge label*, can be represented in different forms as follows:

Form-1:

Read symbol $(a) \longrightarrow$ Write symbol (b), move Left(L) or move Right(R) or No move (N)

$$p \xrightarrow{a \longrightarrow b,L} q$$

Figure 13.7. *Edge Label Format*

Form-2:

Read symbol (a)/Write symbol (b), move Left(L) or move Right (R) or do not move (N)

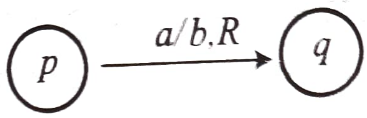


Figure 13.8. *Edge Label Format*

EXAMPLE 13.2.1: Consider the initial configuration of the tape, as shown in figure 13.9.

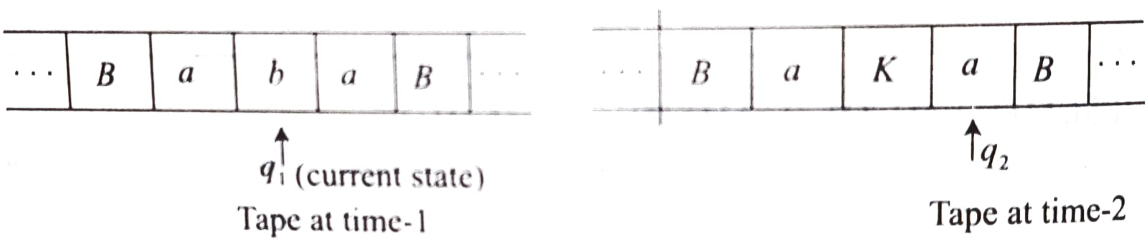


Figure 13.9.

The transition diagrams for this situation, by using the different edge label forms, are:

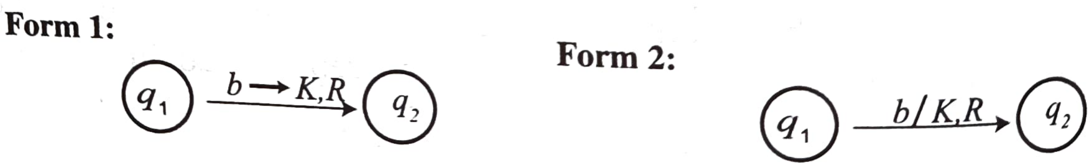


Figure 13.10. *Transition Diagram Using the different Edge Label Forms*

13.2.2 *5-Tuple Specification*

The action performed by a TM, from one state to another state, can be specified by using the 5-tuple:

$\langle$ State-1, Read Symbol, Write Symbol, $L/R/N$, State-2 $\rangle$

EXAMPLE: 5-tuple specification, for the action performed by the TM in figure 13.9, is:

$$< q_1, b, K, R, q_2 > .$$

13.2.3 *Transition Table*

The description of how a TM operates for a given set of symbols on the tape can be represented in a tabular format, called transition table or **action table**. In other words, the transition table describes the following for the given state and the symbol it currently reads:

a. write a symbol
b. move the head (left one step(L) or right one step(R) or no move (N))
c. assume the same or a new state, as prescribed.

The transition tables can be represented in different forms as shown:

Form-1:

Current State	Read Symbol	Write Symbol	Move Tape	Final State	5-tuples Specification

Table 13.1 *Action table form-1*

Form-2:

Current State	q_1			...	q_n		
Tape Symbol	Write Symbol	Move Tape	Next State	...	Write Symbol	Move Tape	Next State

Table 13.2 *Action table form-2*

Form-3:

States \ Tape Symbol	a_1	a_2	a_3	$\cdots$	a_n
	<action>				

where <action> = $\langle$next state, write symbol, move$\rangle$.

Table 13.3 *Action table form-3*

EXAMPLE 13.2.2: Consider a TM, whose task is described in the transition diagram shown in figure 13.11:

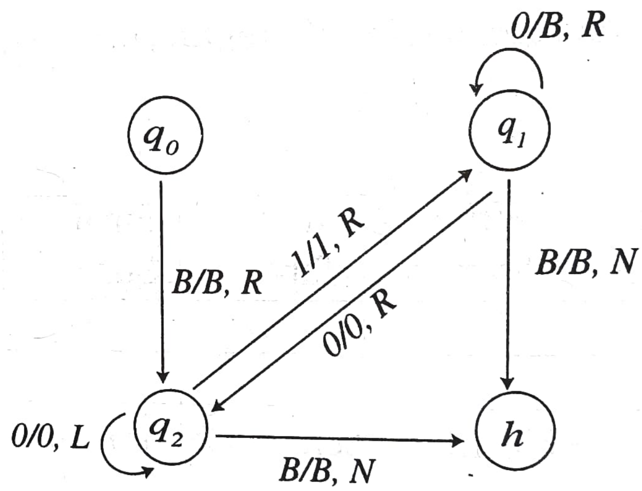


Figure 13.11. *Transition Diagram*

The transition table, in different forms for the above TM is given below:

a.

Current State	Read Symbol	Write Symbol	Move Tape	Final State	5-tuple Specification
q_0	B	B	R	q_2	(q_0, B, B, R, q_2)
q_1	0	0	R	q_2	$(q_1, 0, 0, R, q_2)$
q_1	1	B	R	q_1	$(q_1, 1, B, R, q_1)$
q_1	B	B	N	h	(q_1, B, B, N, h)
q_2	0	0	L	q_2	$(q_2, 0, 0, L, q_2)$
q_2	1	1	R	q_1	$(q_2, 1, 1, R, q_1)$

Table 13.4 *Transition table in form-1*

States \ Tape Symbols	0	1	B
q_0	–	–	$< q_2, B, R >$
q_1	$< q_2, 0, R >$	$< q_1, B, R >$	$< h, B, N >$
q_2	$< q_2, 0, L >$	$< q_1, 1, R >$	$< h, B, N >$

Table 13.5 *Transition table in form-3*

b.

Current States	q_0			q_1			q_2		
Tape Symbol	Write Symbol	Move	Next State	Write Symbol	Move	Next State	Write Symbol	Move	Next State
0	–	–	–	0	R	q_2	0	L	q_2
1	–	–	–	B	R	q_1	1	R	q_1
B	B	R	q_2	B	N	h	B	N	h

Table 13.6 *Transition table in form-2*

13.3 Observations on TM

a. **No $\in$-Transitions** are allowed in a TM.

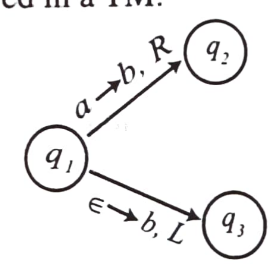


Figure 13.12. *Transition Diagram, $\in \to b, L$ is not Allowed*

b. **A Turing machine halts** if there are no possible transitions to follow.
c. In case the TM halts, we say that the word on the input tape is accepted by the TM.
d. In a TM, the halt states have no outgoing transitions.
e. **Infinite loop** in a TM: (Hanging in some states)

Because of the infinite loop:
(i) the final state cannot be reached.
(ii) the machine never halts.

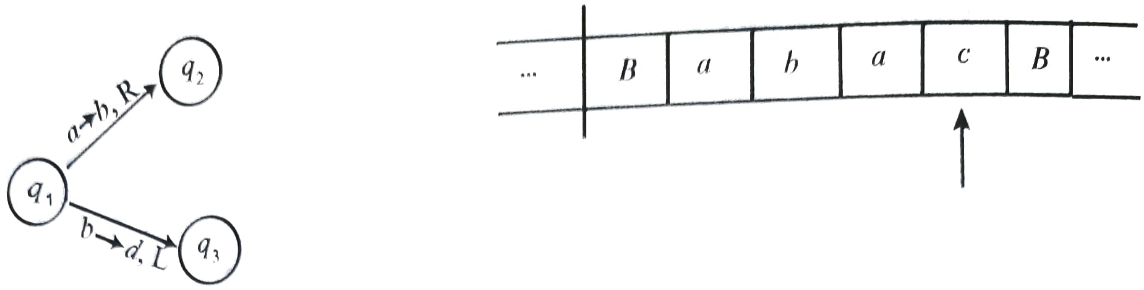


Figure 13.13. *A TM which Halts, Since no Possible Transition*

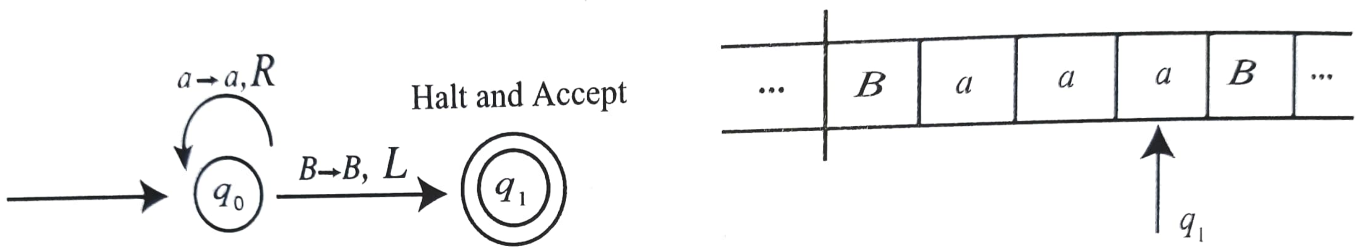


Figure 13.14. *A TM that accepts the word $w = aaa$*

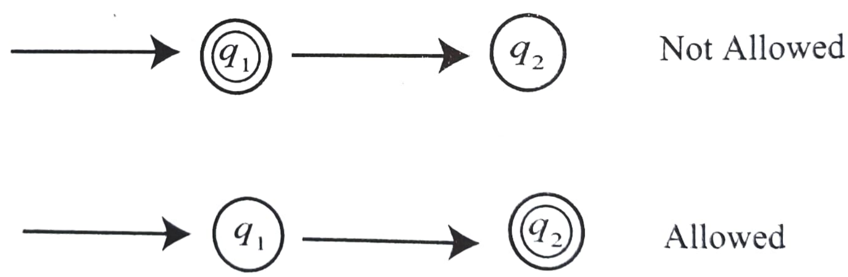


Figure 13.15. *Transition Diagram Showing No Outgoing Transition for Halt State*

(iii) the input is not accepted.

EXAMPLE 13.3.1: Consider a TM, whose action is described in figure 13.16:

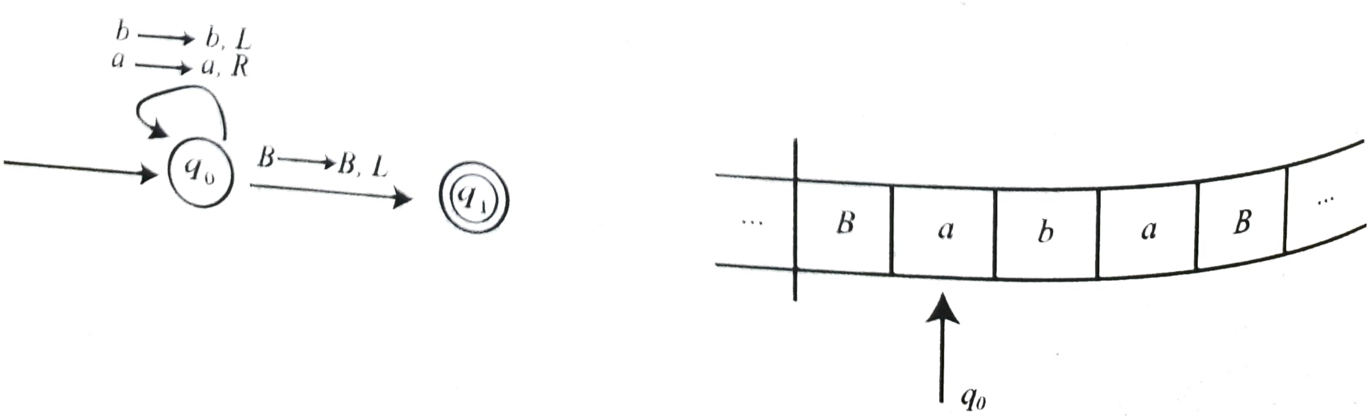


Figure 13.16. *A TM at time 0*

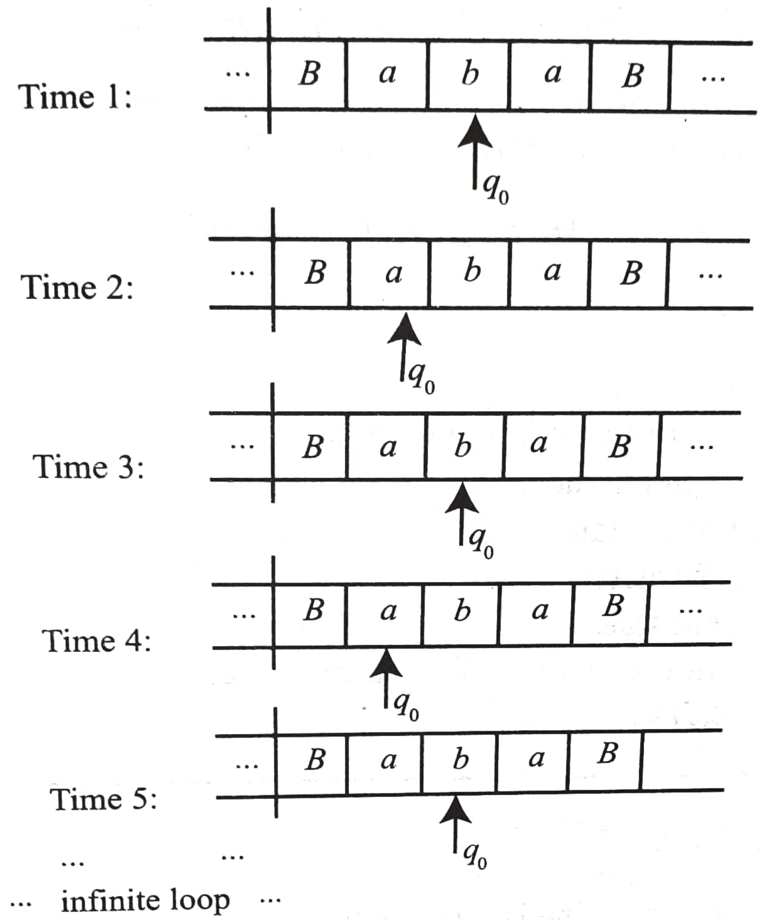


Figure 13.17. *Infinite Loop*

13.4 Elements of TM

A TM has the following seven characteristics:

a. A finite set Q of states, $q_0, q_1, \cdots, q_n$.
b. A finite input alphabet of letters, $\sum = \{a, b, \ldots\}$.
c. A finite alphabet Γ, of tape characters. The tape alphabet does not contain blank B, although a TM can write B onto its tape which is called *erasing*.
d. A transition function δ, which tells how the machine goes from one step to the next i.e., δ describes the following to be performed for a given character scanned at the current state:
   - what character to be written on the tape and,
   - tape head movements (Left, Right, No move) or (L, R, N).
e. An initial state q_0.
f. A special symbol B indicating blank character. $\sum$ does not include B.
g. A set of halt states 'h'.

13.4.1 *Ordered Seven-Tuple Specification of a TM*

Formally a turing machine M is represented as a 7-tuple

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$$

where

a. Q is the finite set of states
b. Σ is the finite set of non-blank symbols
c. Γ is the set of tape characters
d. $q_0 \in Q$ is the initial state
e. B is the blank character
f. $h \sqsubseteq Q$ is the final state
g. δ is the transition function of a Turing machine and is defined as $Q \times \Gamma$ to $Q \times \Gamma \times \{L, R, N\}$.

13.4.2 *Transitions of a TM*

The transition of a turing machine is represented as

$$\boxed{\delta(q_i, a_k) = (q_j, a_1, x)}$$

for $q_i \in Q$, $(a_k, a_1) \in \Gamma$ and x is any one of the values 'L', 'R' and 'N'.

The meaning of $\delta(q_i, a_k) = (q_j, a_1, k)$ is that, if q_i is the current state of the TM and a_k is the cell currently under the head, then TM writes a_1 in the cell currently under the head, enters the state q_j and the head moves to the adjacent cell to the right, if the value of x is R. Otherwise, the head moves to the adjacent cell to the left, if the value of x is L and continues scanning the same cell, if the value of x is N.

EXAMPLE 13.4.1: For the TM in figure 13.11, the transition functions are:

a. $\delta(q_0, B) = (q_2, B, R)$.
This means that for q_0 as the current state and B as the cell currently under the head, the TM writes B in the cell currently under the head, enters the state q_2 and the head moves to the adjacent cell at right. Similarly, the other transitions of TM are:
b. $\delta(q_1, 0) = (q_2, 0, R)$
c. $\delta(q_1, 1) = (q_1, B, R)$
d. $\delta(q_1, B) = (h, B, N)$
e. $\delta(q_2, 0) = (q_2, 0, L)$

f. $\delta(q_2, 1) = (q_1, 1, R)$
g. $\delta(q_2, B) = (h, B, N)$.

13.5 Instantaneous Description of a TM

The complete state of a TM, at any point during a computation, may be described by.

a. the name of the state that in which the machine is
b. the symbols on the tape and
c. the cell that is currently being scanned.

A description of these three data is called *instantaneous description*(ID) or *configuration* of a TM. A simple way to represent such a description is shown below:

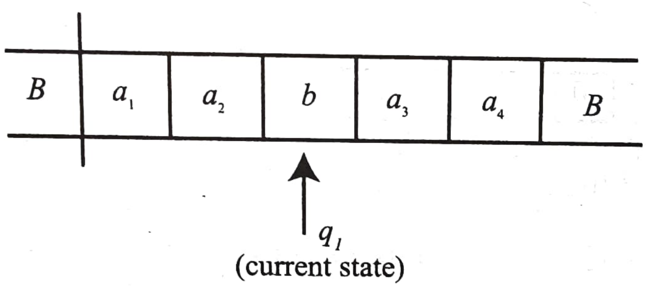


Figure 13.18. *Instantaneous Description:* $a_1\, a_2\, q_1\, ba_3\, a_4$

Formal Definition: An ID of a TM is a string xqy, where q is the current state, xy is the string made form the tape symbols Γ. The head points to the first character of the substring y. The initial ID is denoted by qxy, where q is the start state and the head points to the first symbol from left–x. The final ID is denoted by $xyqB$, where $q \in h$ is the final state and the head points to the blank character denoted by B.

EXAMPLE 13.5.1: Consider a TM, whose action is described in the transition table shown in table 13.7:

States \ Input	0	1	B
q_0	–	–	(q_2, B, R)
q_1	$(q_2, 0, R)$	(q_1, B, R)	(h, B, N)
q_2	$(q_2, 0, L)$	$(q_1, 1, R)$	(h, B, N)

Table 13.7 *Transition table*

The action of the TM with the string $w = 1010$ is shown in figure 13.19:

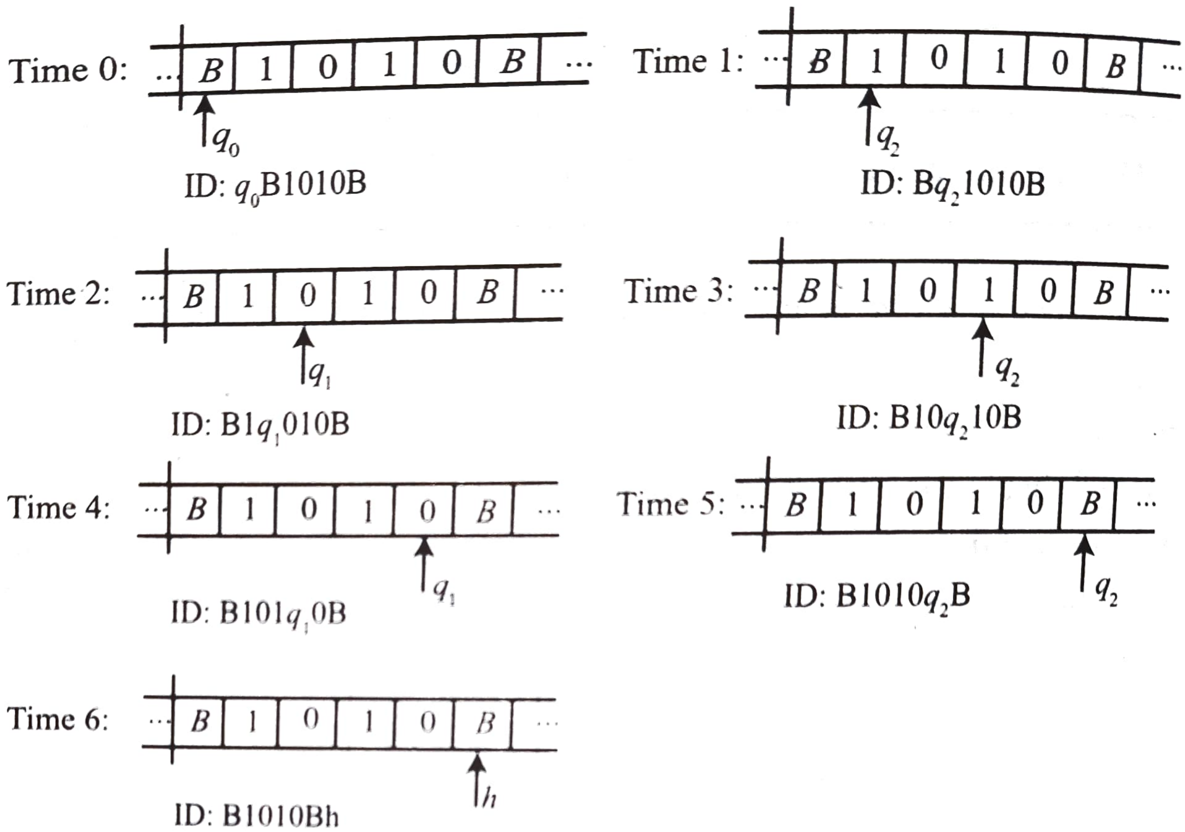


Figure 13.19. *ID of the TM for the String* $w = 1010$

13.6 Moves of a TM

As discussed in the previous sections, there are three possible different types of moves, viz.,

- Move to the left
- Move to the right and
- No move.

In this section, we give the formal definition to the moves of a TM.

Formally, let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ be a TM. Let the ID of M be

$$(q, a_1, a_2, \ldots, a_{i-1}, a_i, a_{i+1} \ldots a_n).$$

Consider the following transitions:

a. $\delta(q, a_i) = \delta(P, b, L)$, for moving to the left.

Case i: If $i > 1$, then the move of TM in going from the ID $(q, a_1a_2, \ldots, a_{i-1}, \mathbf{a_i}, a_{i+1} \ldots a_n)$ to ID $(P, a_1 \ldots a_{i-2}, \mathbf{a_{i-1}}, a_ia_{i+1}..a_n)$ is denoted by:

$$(q, a_1a_2 \ldots a_{i-1}\mathbf{a_i},\ a_{i+1} \ldots a_n) \vdash (P, a_1 \ldots a_{i-2}, \mathbf{a_{i-1}}, b, a_{i+1} \ldots a_n).$$

Case ii: If $i = 1$, the TM crashes, as it is already scanning the leftmost symbol at cell i and attempts to move to the left, which is not possible. Hence, move is not defined.

Case iii: If $i = n$ and B is the blank symbol, then

$$(q, a_1a_2 \ldots a_{n-1}, \mathbf{a_n}, e) \vdash (q, a_1a_2 \ldots a_{n-2}, a_{n-1}, B, e).$$

b. $\delta(q, a_i) = \delta(P, b, R)$, for moving to the right.

Case i: If $i < n$, then the move of TM is .

$$(q, a_1 \ldots a_{i-1}, \mathbf{a_i},\ a_{i+1} \ldots a_n) \vdash (P, a_1 \ldots a_{i-1}, b\ \mathbf{a_{i+1}}, a_{i+2} \ldots a_n).$$

Case ii: If $i = n$ then

$$(q, a_1 \ldots a_{n-1}, \mathbf{a_n}, e) \vdash (P, a_1 \ldots, B, e).$$

$\delta(q, a_i) = (P, b, N)$ when head does not move.

Then, the move is denoted as

$$(q, a_1 \ldots a_{i-1}, \mathbf{a_i}, a_{i+1} \ldots a_n) \vdash (P, a_1 \ldots a_{i-1}, \mathbf{b}, a_{i+1} \ldots a_n).$$

Note-1: e marks the end of the string.

EXAMPLE 13.6.1: Consider a TM, whose action is described in the transition table shown in table 13.8.

States \ Input	0	1	B
q_0	–	–	(q_2, B, R)
q_1	$(q_2, 0, R)$	(q_1, B, R)	(h, B, N)
q_2	$(q_2, 1, L)$	$(q_1, 1, R)$	(q_2, B, R)

Table 13.8 *Transition table*

The action of TM for the string $w = 0101$ is as follows:

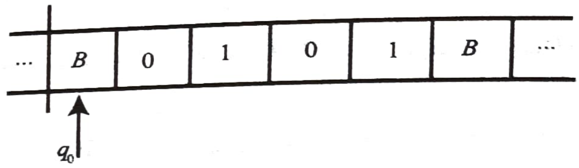


Figure 13.20. *Tape with Symbols 0101*

Consider the following transitions:

a. $\delta(q_0, B) = (q_2, B, R)$, this is represented by the move as

$$q_0B0101B \vdash Bq_20101B$$

b. $\delta(q_2, 0) = (q_2, 1, L)$

$$Bq_20101B \vdash q_2B1101B$$

c. $\delta(q_2, B) = (q_2, B, R)$

$$q_2B1101B \vdash Bq_21101B$$

d. $\delta(q_2, 1) = (q_1, 1, R)$

$$Bq_21101B \vdash B1q_1101B$$

e. $\delta(q_1, 1) = (q_1, B, R)$

$$B1q_1101B \vdash B1Bq_101B$$

f. $\delta(q_1, 0) = (q_2, 0, R)$

$$B1Bq_101B \vdash B1B0q_21B$$

g. $\delta(q_2, 1) = (q_1, 1, R)$

$$B1B0q_21B \vdash B1B01q_1B$$

h. $\delta(q_1, B) = (h, B, N)$

$$B1B01q_1B \vdash B1B01Bh$$

Thus the **move** is:

$$q_0B0101B \vdash Bq_20101B \vdash q_2B1101B \vdash Bq_21101B \vdash B1q_1101B \vdash B1Bq_101B \vdash B1B0q_21B \vdash B1B01q_1B \vdash B1B01Bh$$

The **equivalent notation** is:

$$q_0B0101B \overset{*}{\vdash} B1B01Bh.$$

13.7 String Classes in TM

Every Turing machine TM, over the alphabet Σ, divides the set of input string w into three classes:

a. **Accept (TM)** is the set of all strings $w \in \Sigma^*$, such that, if the tape initially contains w and the TM is then run, then TM ends in a HALT state.
b. **Loop (TM)** is the set of all strings $w \in \Sigma^*$, such that, if the tape initially contains w and the TM is then run, then the TM loops forever (infinite loop).
c. **Reject (TM)** is the set of all strings $w \in \Sigma^*$ such that any of the following three cases arise:

Case (i): There may be a state and a symbol under the tape head, for which δ does not have a value.

Case (ii): If the head is reading the leftmost cell (cell i), containing the symbol x, the state of TM is say q then $\delta(q, x)$ suggests a move to the left of the current cell. However, as there is no cell to the left as of the leftmost cell, no move is possible.

Case (iii): If TM enters an infinite loop or if a TM rejects a given string w because of above two cases, we say that the TM crashes (terminates unsuccessfully).

13.8 Language Accepted by a TM

The language accepted by a TM is the set of accepted strings $w \in \Sigma^*$.

Formally, Let $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ be a TM. The language accepted by M denoted by $L(M)$ is defined as
$L(M) = \{w | w \in \Sigma^*$ and if $w = a_1, \ldots a_n$, then

$$(q_0, e, a_1, a_2, \ldots a_n) \overset{*}{\vdash\!\!-\!\!-} (h, b_1, \ldots b_{j-1}, b_j \ldots b_n)$$

for some $b_1, b_2 \ldots b_n \in \Gamma^*\}$

or

$$L(M) = \{w : q_0 w \overset{*}{\vdash\!\!-\!\!-} x_1 h x_2\}.$$

a. **Turing Acceptable Language**
A language L over some alphabet is said to be Turing Acceptable language, if there exists a TM, M such that $L = L(M)$.

b. **Turing Decidable Language**
A language L over Σ i.e., $L \sqsubseteq \Sigma^*$ is said to be Turing decidable, if both the languages L and its complement $\Sigma^* - L$ are Turing acceptable.

c. **Recursively enumerable language**
A language L is recursively enumerable, if it is accepted by a TM.

We discuss in detail, the TM languages, in chapter 14-section 14.7.

13.9 Role of TM's

The TMs are designed to play atleast the following three roles:

a. Accepting devices for languages (similar to the role played by FAs and PDAs).
b. Computer of functions
In this role, a TM represents a particular function (say the SUM function which gives as output, the sum of two positive integers given as input). Here the initial input represents an argument of the function and the (final) string on the tape (when the TM enters the Halt State) is treated as the value obtained by the application of the function to the argument represented by the initial string.
c. An enumerator of strings of a language, that outputs the strings of a language (one at a time) in some systematic order i.e. as a list.

13.10 Design of TM's

The basic strategy for designing a TM is given below:

a. The objective of scanning a symbol by the tape head is to know about the future status. The machine must remember the symbols scanned previously, by going to the next unique state.
b. The number of states must be minimised. This can be achieved by changing the states:
   - only when there is a change in the written symbol or
   - when there is a change in the movement of the tape head.

13.10.1 TM as Accepting Devices for Languages

This concept is illustrated through the following examples.

EXAMPLE 13.10.1: Design a TM that erases all non-blank symbols on the tape, over the alphabets $\{a, b\}$.

Design Strategy: The TM in state q_0 must perform the following operations:

- On input symbol a, replace a by B, move the tape head towards right and stay at q_0.
- On input symbol b, replace b by B, move the tape head towards right and stay at q_0.
- On input symbol B, replace B by B, change the state to h and do not move tape head.

Thus the TM $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ is,

$$M = (\{q_0, h\}, \{a, b\}, \{a, b, B\}, \delta, q_0, B, h)$$

where δ is given by:

Transition diagram:

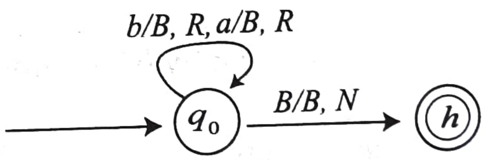


Figure 13.21. *Transition Diagram for M to Erase All Non-Blank Symbols*

Transition table:

Tape Symbol / States	a	b	B
q_0	$< q_0, B, R >$	$< q_0, B, R >$	$< h, B, N >$
h	–	–	Accept

Table 13.9 *Transition table for M*

TM action for the string $w = abab$

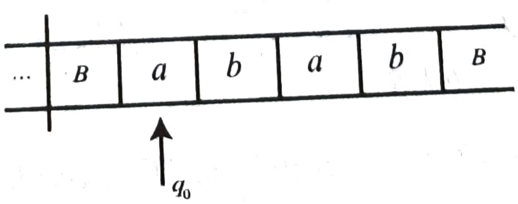


Figure 13.22. *The Tape*

$$\text{ID is: } q_0ababB \vdash Bq_0babB$$
$$\vdash BBq_0abB$$
$$\vdash BBBq_0bB$$
$$\vdash BBBBq_0B$$
$$\vdash BBBBBh$$

Since the final state h is reached, the string $abab$ is accepted.

EXAMPLE 13.10.2: Design a TM that accepts the language of all strings, over the alphabet $\Sigma = \{a, b\}$, whose second letter is b.

Design Strategy:

Step-1: In state q_0

- On input symbol a, change to state q_1, replace a by a and move the tape head towards right.
- On input symbol b, change to state q_1, replace b by b and move the tape head towards right.

Step-2: In state q_1 on input symbol b, replace b by b and move the tape head towards right.

Step-3: In state q_2

- On input symbol a, stay at q_2, replace a by a and move the tape head towards right.
- On input symbol b, stay at q_2, replace b by b and move the tape head towards right.
- On input symbol B, change to state h and do not move the tape head.

Thus, the TM, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ is

$$M = (\{q_0, q_1, q_2, h\}, \{a, b\}, \{a, b, B\}, \delta, q_0, B, h),$$

where δ is given by:

Transition diagram:

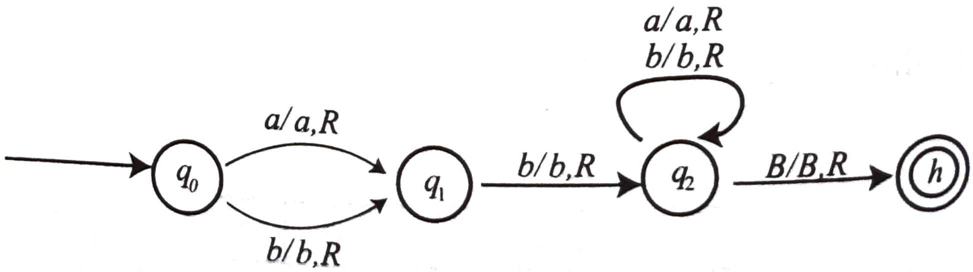


Figure 13.23. *Transition Diagram for TM*

Transition table:

Tape Symbol / States	a	b	B
q_0	$< q_1, a, R >$	$< q_1, b, R >$	–
q_1	–	$< q_2, b, R >$	–
q_2	$< q_2, a, R >$	$< q_2, b, R >$	$< h, B, N >$
h	–	–	Accept

Table 13.10 *Transition table for TM*

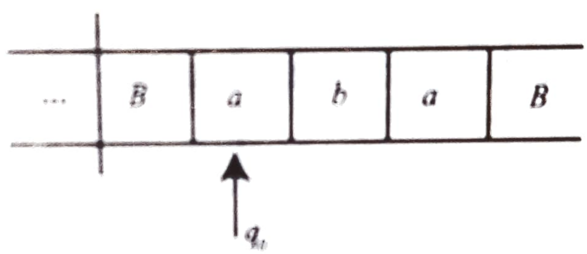


Figure 13.24. *The Tape*

a. **TM action for the string** $w = aba$

$$\text{ID is: } q_0abaB \vdash aq_1baB$$
$$\vdash abq_2aB$$
$$\vdash abaq_2B$$
$$\vdash abaBh$$

Since the final state h is reached, the string aba is accepted.

b. **TM action for the string** $w = aaa$

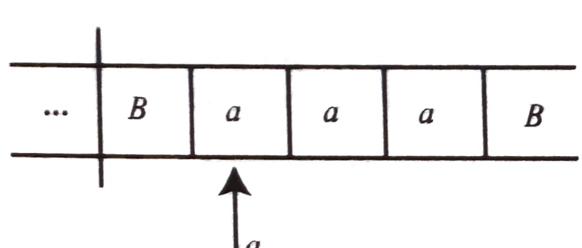


Figure 13.25. *The Tape*

$$\text{ID is: } \quad q_0aaaB \vdash aq_1aaB$$

Transition is not defined for (q_1, a). The machine halts and the string $w = aaa$ is rejected. In other words, we say that the TM crashes on giving string w as input.

EXAMPLE 13.10.3: Design a TM which accepts all strings of the form $a^n b^n$ for $n \geqslant 1$.

Design strategy:

Let q_0 be the start state and the tape head point to the first symbol of the string to be scanned.

Step-1: In state q_0,

- on input symbol a, change to state q_1, replace a by 'x' and move the tape head towards right.
- if you encounter y, change the state to q_3, replace y by y and move the tape head towards right.

Step-2: In state q_1, search for the leftmost b and replace it by y. Now, move the head to point to the leftmost b. When the head is moved towards b, the symbol encountered may be a or y. Irrespective of what symbol is encountered, replace a by a, y by y, remain in state q_1 and move the read towards right.

Transitions are :
$$\delta(q_1, a) = (q_1, a, R)$$
$$\delta(q_1, y) = (q_1, y, R)$$

Step-3: In state q_2, search for the rightmost x to get leftmost a. During this process, the symbols encountered may be y's and a's. Replace y by y, a by a, remain in state q_2 and move the head towards left.

Transitions are :
$$\delta(q_2, y) = (q_2, y, L)$$
$$\delta(q_2, a) = (q_2, a, L)$$

Once rightmost x is obtained. To get the leftmost a, replace x by x, change the state to q_0 and move the head towards right.

Transition :
$$\delta(q_2, x) = (q_0, x, R)$$

Step-4: In state q_3, search for y or blank B. On encountering y replace it by y, remain in q_3 and move the head towards right. On encountering B, change to state q_4, replace B by B and move towards left.

Thus, the TM, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ is

$$M = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b\}, \{a, b, x, y\}, \delta, q_0, B, \{q_4\}),$$

where δ is given by:

Transition diagram:

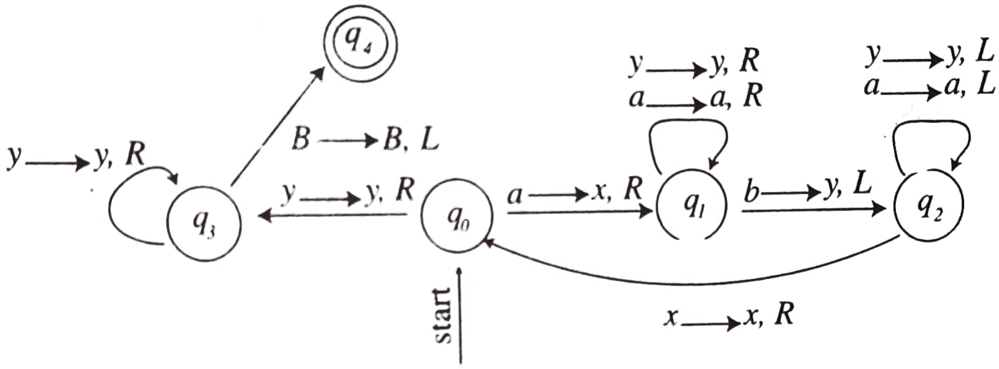


Figure 13.26. *Transition Diagram for TM*

Transition table:

States \ Tape Symbol	a	b	x	y	B
q_0	(q_1, x, R)	–	–	(q_3, y, R)	–
q_1	(q_1, a, R)	(q_2, y, L)	–	(q_1, y, R)	–
q_2	(q_2, a, L)	–	(q_0, x, R)	(q_2, y, L)	–
q_3	–	–	–	(q_3, y, R)	(q_4, B, L)

Table 13.11 *Transition table for TM*

TM action for the string $w = aabb$

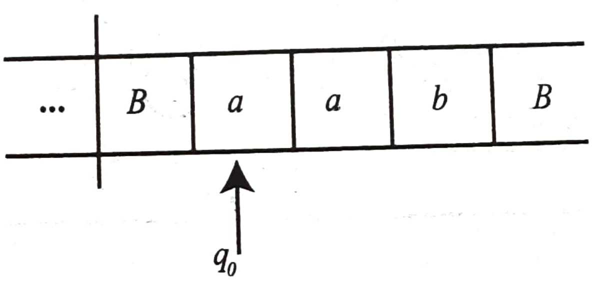
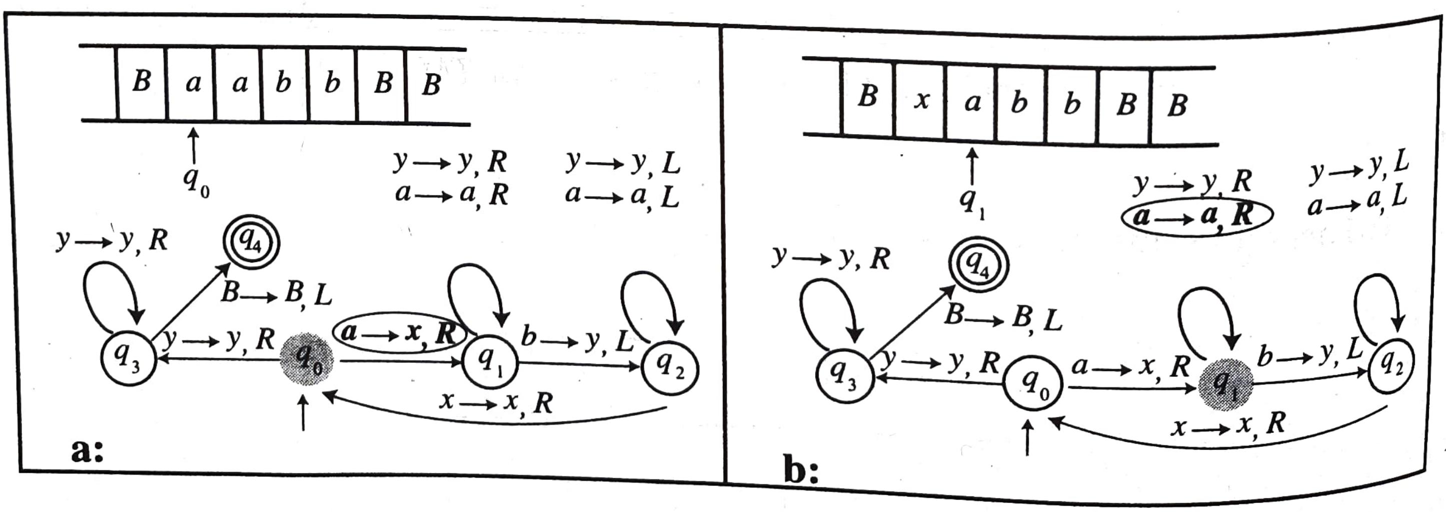


Figure 13.27. *The Tape*

ID is: $\quad q_0aabbB \vdash xq_1abbB$

$$\begin{aligned}
&\vdash xaq_1bbB\\
&\vdash xq_2aybB\\
&\vdash q_2xaybB\\
&\vdash xq_0aybB\\
&\vdash xxq_1ybB\\
&\vdash xxyq_1bB\\
&\vdash xxq_2yyB\\
&\vdash xq_2xyyB\\
&\vdash xxq_0yyB\\
&\vdash xxyq_3yB\\
&\vdash xxyyq_3B\\
&\vdash xxyyq_4
\end{aligned}$$

Since the final state q_4 is reached, the string *aabb* is accepted. The figure 13.28(a-n) below shows the action of the TM for the string *aabb*.



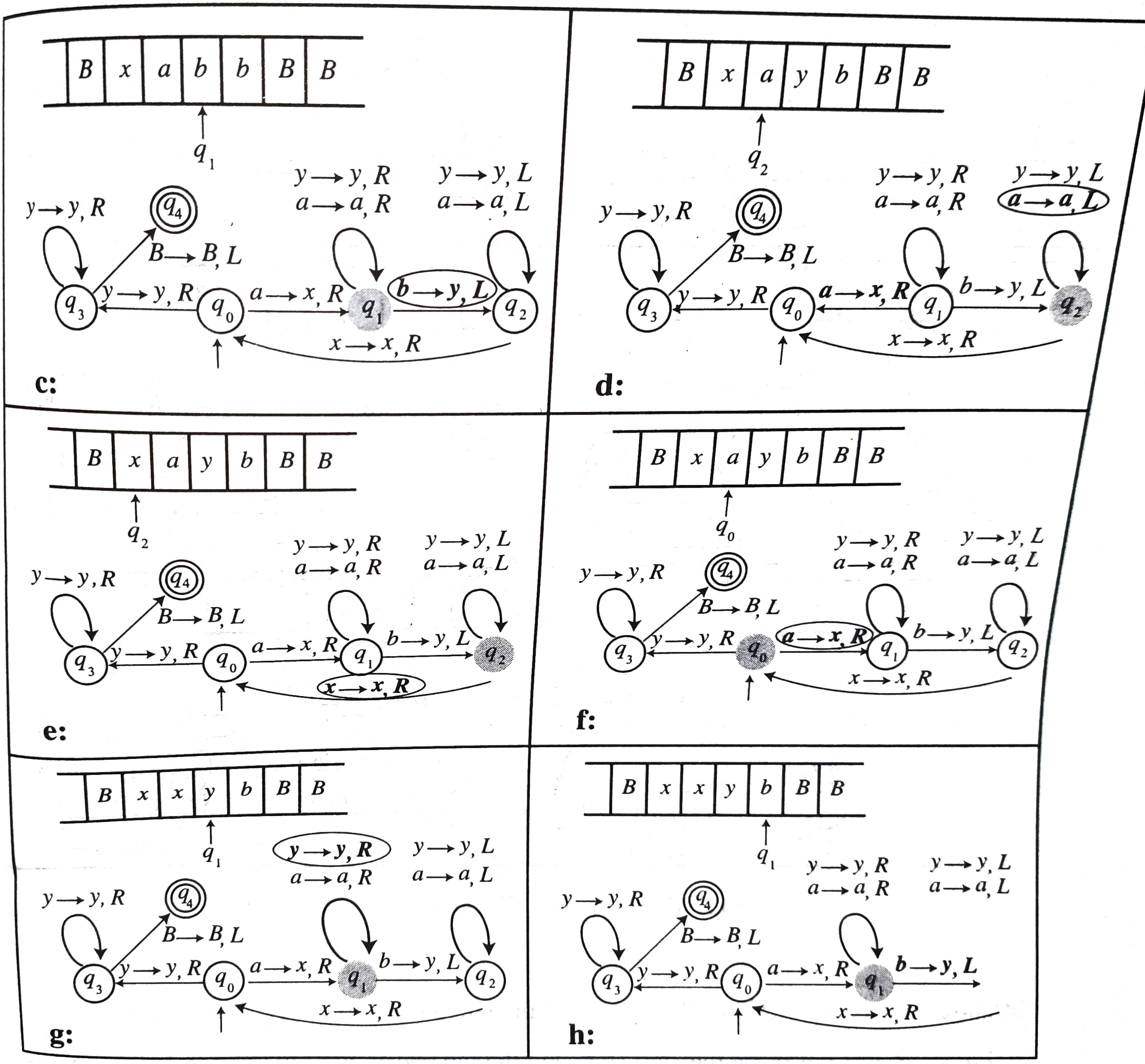


Figure 13.28. *a–n: The TM action for w = aabb.*

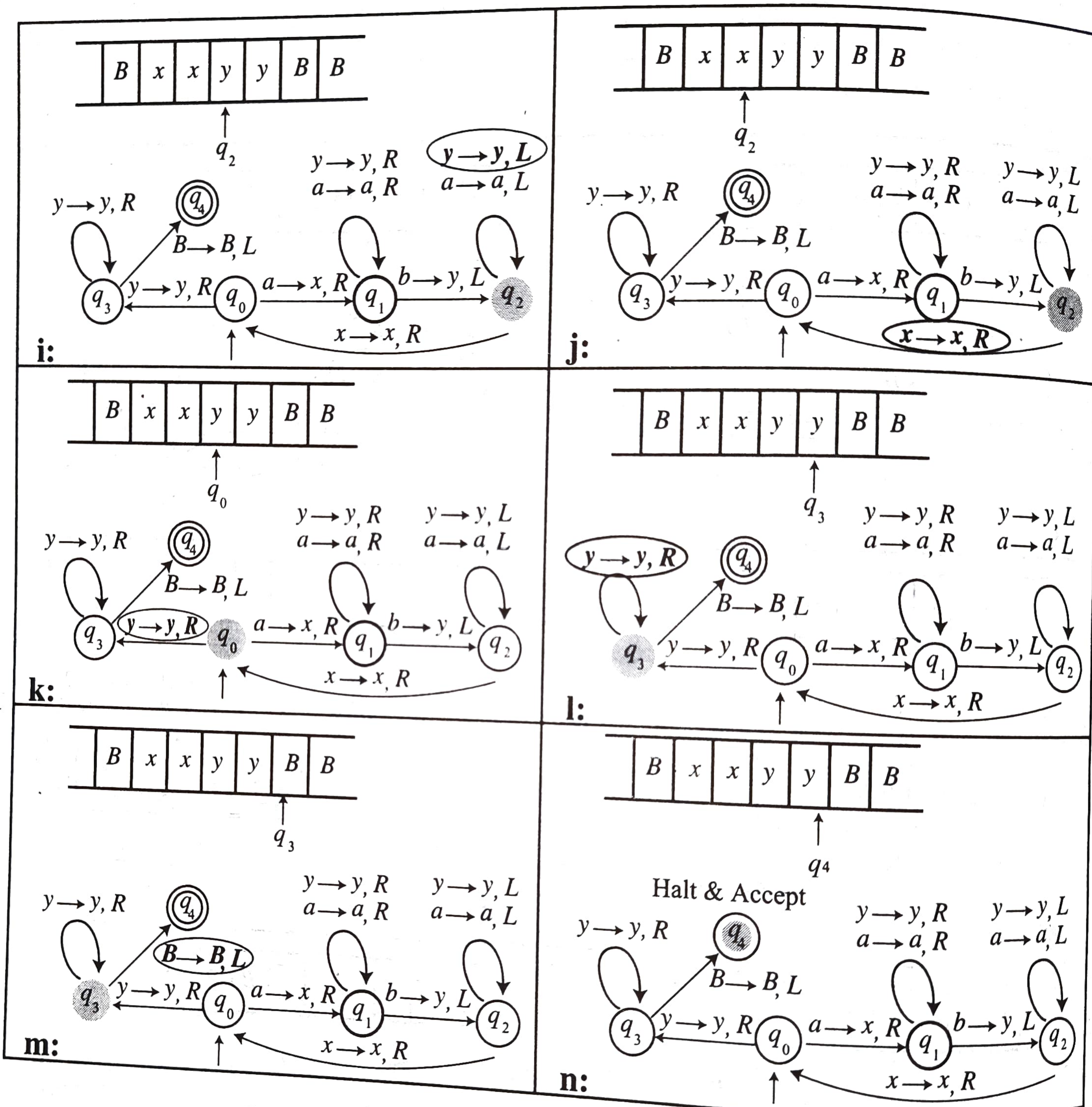


Figure 13.28. *a–n: The TM Action for $w = aabb$*

EXAMPLE 13.10.4: Design a TM to accept the language $L(M) = \{a^n b^n c^n | n \geq 1\}$.

Design strategy: Let q_0 be the start state and the tape head point to the first symbol of the string to be scanned.

Step-1: ■ In state q_0 on input symbol a, change to state q_1, replace a by x and move the tape head towards right.

- In state q_1 on input symbol b, change to state q_2, replace b by y and move the tape head towards right.
- In state q_2 on input symbol c, change to q_3, replace c by z and move the tape head towards left.

Step-2: In state q_1, search for the leftmost b. In the process of searching the symbols a or y may be encountered. So replace a by a, y by y and move the head towards right and stay in state q_1

Transitions are :
$$\delta(q_1, a) = (q_1, a, R)$$
$$\delta(q_1, y) = (q_1, y, R).$$

Step-3: In state q_2, search for the leftmost b. In this process of searching, the symbols b or z may be encountered. So, replace b by b, z by z, move the tape head towards right and remain in state q_2.

Transitions are :
$$\delta(q_2, b) = (q_2, b, R)$$
$$\delta(q_2, z) = (q_2, z, R).$$

Step-4: TM in state q_3 means that equal number of a's, b's and c's are replaced by equal number of x's, y's and z's. In q_3, search for the rightmost x to get the leftmost a. During this process, the symbols z, b, y, a and x may be encountered. So replace them by the same respective symbols, move the tape head towards left and stay in q_3.

Transitions are :
$$\delta(q_3, z) = (q_3, z, L), \quad \delta(q_3, b) = (q_3, b, L)$$
$$\delta(q_3, y) = (q_3, y, L), \quad \delta(q_3, a) = (q_3, a, L)$$

Now once x is encountered, replace x by x, change to state q_0 and move the head towards right to get the leftmost a.

Transition :
$$\delta(q_3, x) = (q_0, x, R)$$

Step-5: In state q_0 on input symbol y, change to state q_4, replace y by y and move the tape head towards right.

Step-6: In state q_4 on input symbol y, stay in q_4, replace y by y and move the head towards right.

Transition :
$$\delta(q_4, y) = (q_4, y, R).$$

If z is encountered in q_4, change to q_5, replace z by z and move the head towards right.

$$\text{Transition}: \qquad \delta(q_4, z) = (q_5, z, R).$$

(TM in q_4, with input z means that there are no b's and no c's).

Step-7: In state $q5$ on input symbol z, stay in q_5, replace z by z and move the head towards right, continue to be in q_5 for the input symbol z, so that there are only z's and no more c's. However, if B is encountered once, change to state q_6, replace B by B and move the head towards right.

$$\text{Transitions}: \qquad \delta(q_5, z) = (q_5, z, R)$$
$$\delta(q_5, B) = (q_6, B, R)$$

Thus $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ i.e.,

$$M = \left(\{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{a, b, c\}, \{a, b, c, x, y, z\}, \delta, q_0, B, q_6\right),$$

where δ is given by:

Transition Diagram:

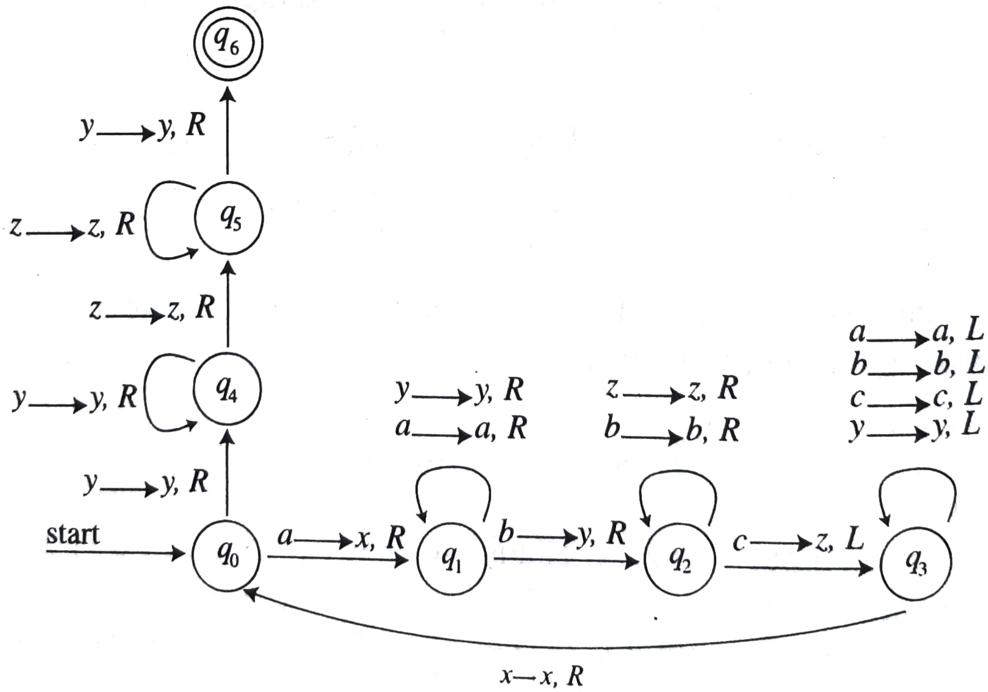


Figure 13.29. *Transition Diagram for M*

Transition table:

States \ Tape Symbols	a	b	c	x	y	z	B
q_0	(q_1, x, R)	–	–	–	(q_4, y, R)	–	–
q_1	(q_1, a, R)	(q_2, y, R)	–	–	(q_1, y, R)	–	–
q_2	–	(q_2, b, R)	(q_3, z, L)	–	–	(q_2, z, R)	–
q_3	(q_3, a, L)	(q_3, b, L)	–	(q_0, x, R)	(q_3, y, L)	(q_3, z, L)	–
q_4	–	–	–	–	(q_4, y, R)	(q_5, z, R)	–
q_5	–	–	–	–	–	(q_5, z, R)	(q_6, B, R)

Table 13.12 *Transition table for M.*

TM action for the string $w = abc$

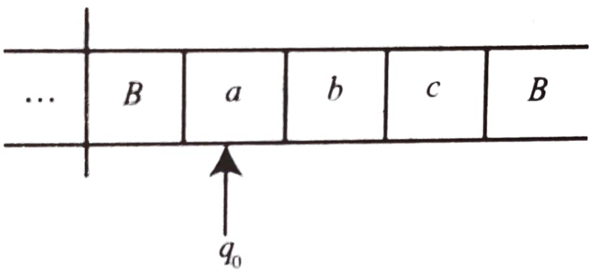


Figure 13.30. *The Tape*

ID is:

$$q_0abcB \vdash xq_1bcB \vdash xyq_2cB \vdash xq_3yzB \vdash q_3xyzB \vdash xq_0yzB \vdash xyq_4zB \vdash xyzq_5B \vdash xyzq_6.$$

Since the final state q_6 is reached, the string $w = abc$ is accepted.

EXAMPLE 13.10.5: Design a TM that recognises the language L of all strings, over $\{a, b\}$, with number of a's equal to the number of b's.

Design Strategy: Let q_0 be the initial state and the tape head points to the first symbol of the string to be scanned, which can either be a or b. The following cases are considered, based on the next input symbol to be scanned.

Transition table:

States \ Tape Symbol	a	b	x	y	B
q_0	(q_1, x, R)	(q_3, x, R)	–	(q_0, y, R)	(h, B, R)
q_1	(q_1, a, R)	(q_2, y, L)	–	(q_1, y, R)	–
q_2	(q_2, a, L)	–	(q_0, x, R)	(q_2, y, L)	–
q_3	(q_4, y, L)	(q_3, b, R)	–	(q_3, y, R)	–
q_4	–	(q_4, b, L)	(q_0, x, R)	(q_4, y, L)	–

Table 13.13 *Transition table.*

TM action for the string $w = ababab$

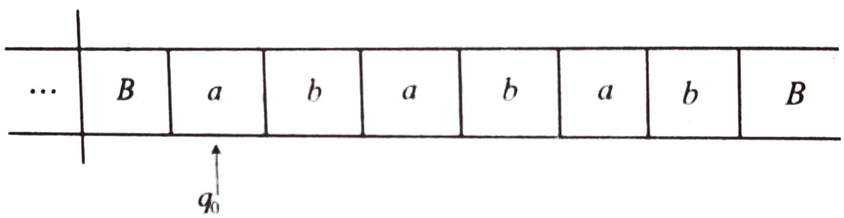


Figure 13.32. *The Tape*

ID is: $q_0abababB \vdash xq_1bababB$

$\vdash q_2xyababB$

$\vdash xq_0yababB$

$\vdash xyq_0ababB$

$\vdash xyxq_1babB$

$\vdash xyq_2xyabB$

$\vdash xyxq_0yabB$

$\vdash xyxyq_0abB$

$\vdash xyxyxq_1bB$

$\vdash xyxyq_2xyB$

$\vdash xyxyxq_0yB$

$\vdash xyxyxyq_0B$

$\vdash xyxyxyh.$

EXAMPLE 13.10.6: Design a TM 'parity counter' that outputs 0 or 1, depending on whether the number of 1's in the input sequence is even or odd respectively.

Design Strategy: Let q_0 be the start state and the tape head point to the first symbol of the string to be scanned. The TM goes from one state to another, replacing 0 by 1 and 1 by 0. The TM ends in 0 (or 1) if the number of 1's in the input sequence is even (or odd).

Thus, the TM, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ is

$$M = \big(\{q_0, q_1, h\}, \{0, 1\}, \{0, 1\}, \delta, q_0, B, h\big),$$

where δ is given by:

Transition diagram:

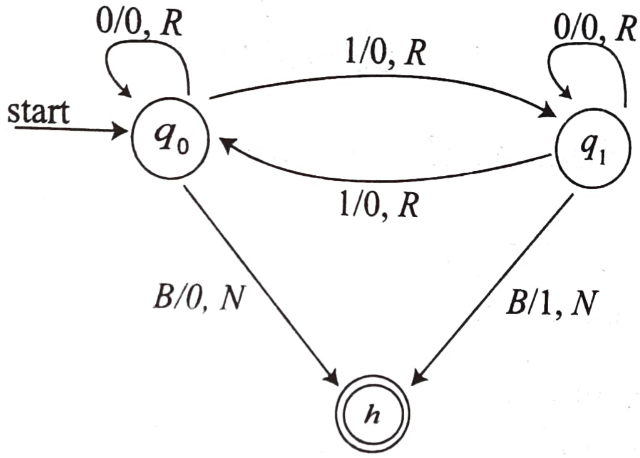


Figure 13.33. *Transition Diagram for TM*

Transition table:

States \ Tape Symbol	0	1	B
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$	$(h, 0, N)$
q_1	$(q_1, 0, R)$	$(q_0, 0, R)$	$(h, 1, N)$

Table 13.14 *Transition table*

TM action for the input $w = 10110101$

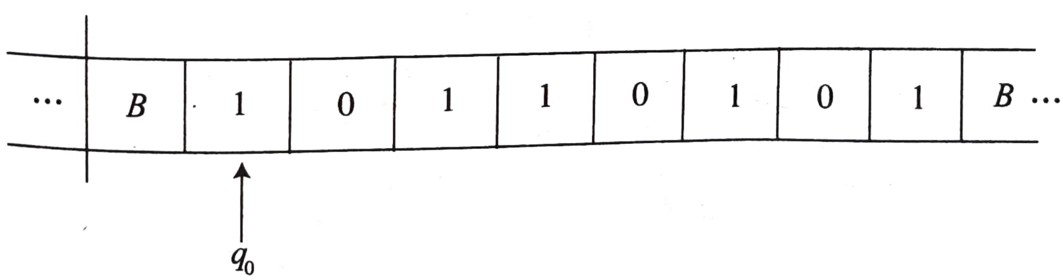


Figure 13.34. *The Tape*

ID is: $\quad q_0 10110101 \vdash 0q_1 0110101$

$\vdash 00q_1 110101$

$$\vdash 000q_0 10101$$
$$\vdash 0000q_1 0101$$
$$\vdash 00000q_1 101$$
$$\vdash 000000q_0 01$$
$$\vdash 0000000q_0 1$$
$$\vdash 00000000q_1 B$$
$$\vdash 000000001h$$

Since the final state h is reached, the TM halts with the output 1 (odd parity).

EXAMPLE 13.10.7: Design a TM 'parentheses checker', that outputs 1 or 0, depending on whether the sequence is properly formed or not.

Design Strategy: Let q_0 be the start state and the tape head point to the first symbol of the string to be scanned. (The machine always starts with the leftmost bracket symbol ('in' state q_0). Thus the input to TM is a sequence of left and right brackets. The output is a 1 or 0, depending on whether the sequence is properly formed or not.

Thus, the TM is $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ i.e.,

$$M = (\{q_0, q_1, q_2, h\}, \{(,)\}, \{(,), x, 0, 1\}, \delta, q_0, B, h),$$

where δ is given by:

Transition diagram:

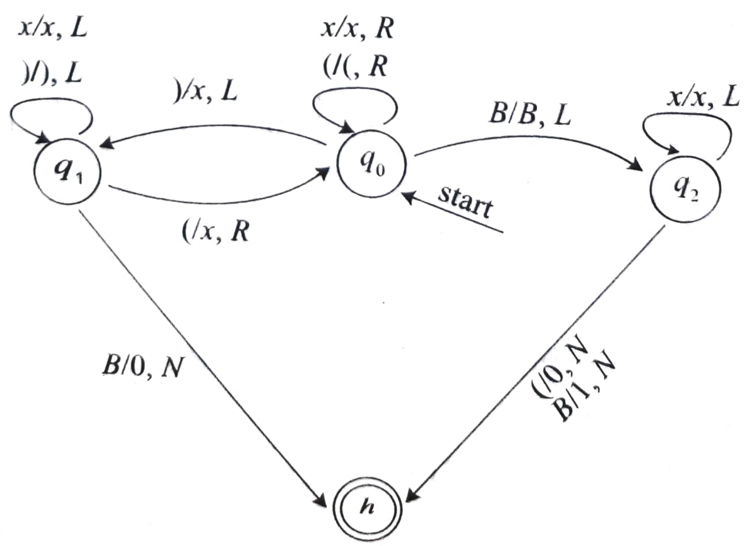


Figure 13.35. *Transition Diagram for TM*

Transition table:

Tape / States	(	)	x	B
q_0	$< q_0, (, R >$	$< q_1, x, L >$	$< q_0, x, R >$	$< q_2, B, L >$
q_1	$< q_0, x, R >$	$< q_1,), L >$	$< q_1, x, L >$	$< h, 0, N >$
q_2	$< h, 0, N >$	–	$< q_2, x, L >$	$< h, 1, N >$

Table 13.15 *Transition table for TM*

TM action for input string $w = (())$

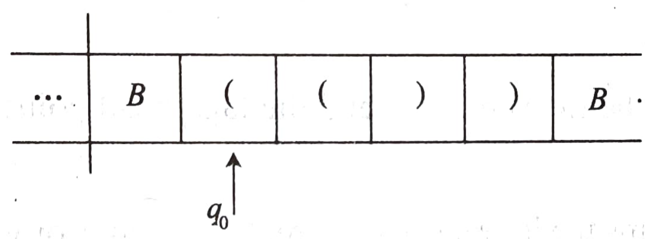


Figure 13.36. *The Tape*

ID is: $q_0(()) \vdash (q_0())$

$\vdash ((q_0))$

$\vdash (q_1(x)$

$\vdash (xq_0x)$

$\vdash (xxq_0)$

$\vdash (xq_1xx$

$\vdash (q_1xxx$

$\vdash q_1(xxx$

$\vdash xq_0xxx$

$\vdash xxq_0xx$

$\vdash xxxq_0x$

$\vdash xxxxq_0B$

$$\vdash xxxq_2xB$$
$$\vdash xxq_2xxB$$
$$\vdash xq_2xxxB$$
$$\vdash q_2xxxB$$
$$\vdash q_2BxxxxB$$
$$\vdash h1xxxxB[2pt]$$

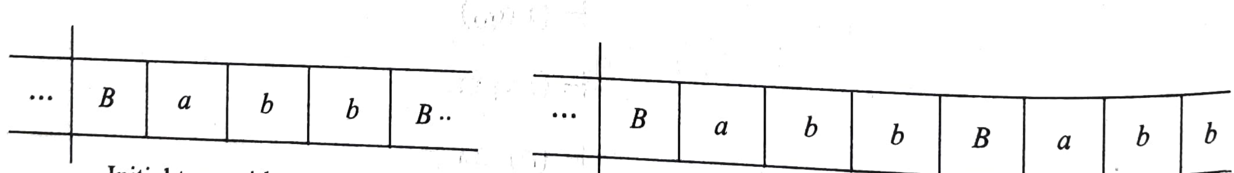
Since the final state h is reached, the TM halts with the output 1 (balanced parentheses).

EXAMPLE 13.10.8: Design a TM that copies a given string over $\{a, b\}$. Find the computation of TM for the string *abb*.

Design Strategy: Let q_0 be the start state and the tape head point to the blank symbol on the tape.

The TM in state q_1 scans the leftmost a or b, replaces it by x or y and copies it in the next available B (the first B on the right is left as marker and will not be available). If the symbol in the state q_1 is a, the TM (while skipping symbols) passes through the state q_2 and reaches q_4. However, if the symbol in state q_1 is b, then the TM (while skipping the symbols) passes through the state q_3 and reaches the state q_5. Then the TM copies the symbol and reaches the state q_6. Next, the TM starts its leftward scan, skipping over a's, b's, x's, y's and B, and meets x or y in q_7. At this stage, TM goes to the state q_1, then repeats the whole process until the whole string is copied in the second part of the tape.

Finally, TM goes from q_1 to state q_8 to replace each x by a and each y by b.



Initial tape with $w = abb$ Final tape after TM action

Figure 13.37. *TM to Copy $w = abb$*

Transition diagram:

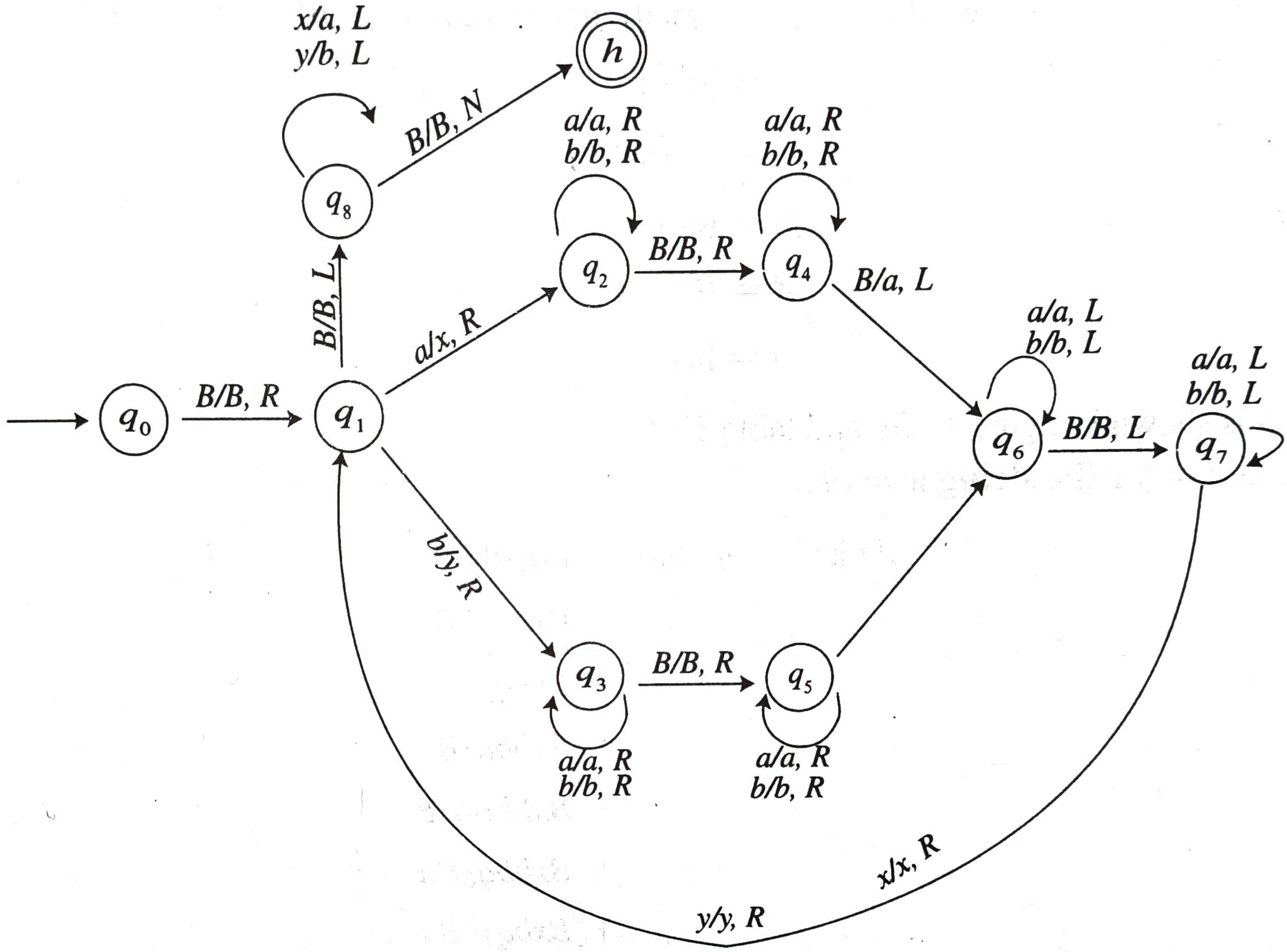


Figure 13.38. *Transition Diagram for TM*

Transition table:

Tape Symbol / States	a	b	x	y	B
q_0	–	–	–	–	(q_1, B, R)
q_1	(q_2, x, R)	(q_3, y, R)	–	–	(q_8, B, L)
q_2	(q_2, a, R)	(q_2, b, R)	–	–	(q_4, B, R)
q_3	(q_3, a, R)	(q_3, b, R)	–	–	(q_5, B, R)
q_4	(q_4, a, R)	(q_4, b, R)	–	–	(q_6, a, L)
q_5	(q_5, a, R)	(q_5, b, R)	–	–	(q_6, b, L)
q_6	(a_6, a, L)	(q_6, b, L)	–	–	(q_7, B, L)
q_7	(q_7, a, L)	(q_7, b, L)	(q_1, x, R)	(q_1, y, R)	
q_8	–	–	(q_8, a, L)	(q_8, b, L)	(h, B, N)

Table 13.16 *Transition table for the TM*

Thus the TM is $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$

$$\text{where} \quad Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, h\}$$

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

$$\Gamma = \{a, b, x, y\}$$

$$q_0 = \{q_0\}$$

$$B = B$$

$$h = \{h\}$$

and δ is shown in figure 13.38 and table 13.16.

TM action for the string $w = abb$

$$
\begin{aligned}
\text{ID is:} \quad q_0Babb &\vdash Bq_1abbB \\
&\vdash Bxq_2bbB \\
&\vdash Bxbq_2bB \\
&\vdash Bxbbq_2B \\
&\vdash BxbbBq_4B \\
&\vdash Bxbbq_6Ba \\
&\vdash Bxbq_7bBa \\
&\vdash Bxq_7bbBa \\
&\vdash Bq_7xbbBa \\
&\vdash Bxq_1bbBa \\
&\vdash Bxyq_3bBa \\
&\vdash Bxybq_3Ba \\
&\vdash BxybBq_5a \\
&\vdash BxybBaq_5B \\
&\vdash BxybBq_6ab \\
&\vdash Bxybq_6Bab \\
&\vdash Bxyq_7bBab \\
&\vdash Bxq_7ybBab \\
&\vdash Bxyq_1bBab \\
&\vdash Bxyyq_3Bab
\end{aligned}
$$

$$\vdash BxyyBq_5ab$$

$$\vdash BxyyBaq_5b$$

$$\vdash BxyyBabq_5B$$

$$\vdash BxyyBaq_6bb$$

$$\vdash BxyyBq_6abb$$

$$\vdash Bxyyq_6Babb$$

$$\vdash Bxyq_7yBabb$$

$$\vdash Bxyyq_1Babb$$

$$\vdash Bxyq_8yBabb$$

$$\vdash Bxq_8ybBabb$$

$$\vdash Bq_8xbbBabb$$

$$\vdash q_8BabbBabb$$

$$\vdash hBabbBabb$$

Since, the final state h is reached, the string abb is accepted.

EXAMPLE 13.10.9: Design a TM that accepts all the palindromes over the alphabet $\{a, b\}$.

Design strategy: Let q_0 be the start state and the tape head point to the first symbol of the string to be scanned.

Step-1: In state q_0, with the input as blank symbol B, the machine has found a palindrome of even length.

Step-2: In state q_0, with the input as non-blank symbol 'a', the machine replaces it by B and enters the state q_1, in which all the a's and b's are skipped. If B is encountered, then it goes from $q1$ to q_3 to find a (matching a in last non-blank symbol position). If there is a match, then it goes to q_5 and replaces 'a' by 'B'.

Step-3: In state q_2, if only B's are encountered, it means the previous 'a' was the middle symbol of the given string. Then, the machine has found the palindrome of odd length.

Step-4: For an input of non-blank symbol 'b' in state q_0, steps similar to 2 and 3 follow except that the next state is q_2 (with a's and b's interchanged).

Step-5: When the machine in state q_3 finds b or finds a when in state q_4, then the string under coorideration is not a palindrome.

Thus the TM is, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ i.e,

$$M = \left(\{q_0, q_1, q_2, q_3, q_4, q_5, h\}, \quad \{a, b\}, \quad \{a, b\}, \delta, q_0, B, h\right),$$

where δ is given by:

Transition diagram:

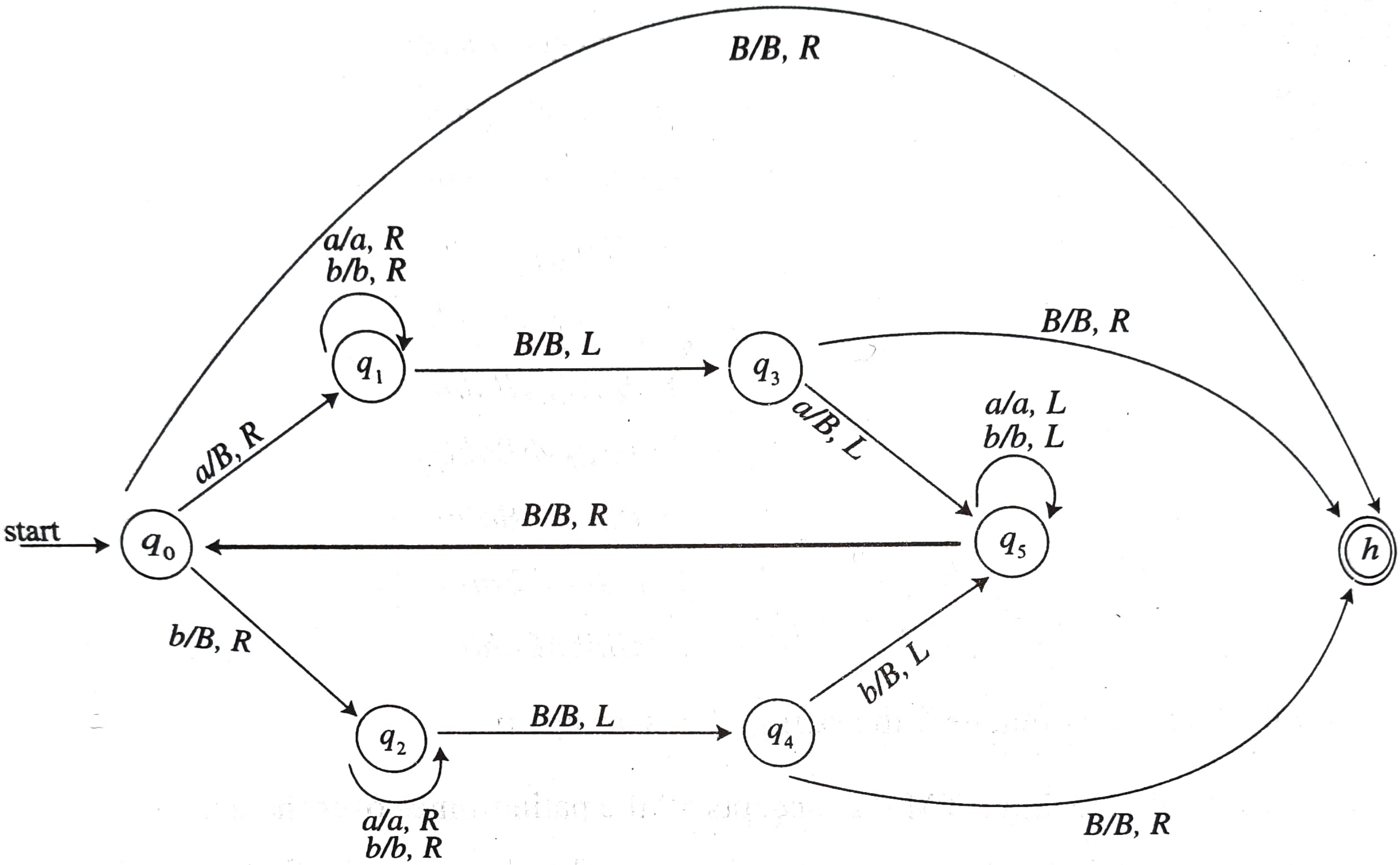


Figure 13.39. *Transition Diagram for M*

Transition table:

States \ Tape Symbol	a	b	B
q_0	(q_1, B, R)	(q_2, B, R)	(h, B, R)
q_1	(q_1, a, R)	(q_1, b, R)	(q_3, B, L)
q_2	(q_2, a, R)	(q_2, b, R)	(q_4, B, L)
q_3	(q_5, B, L)	–	(h, B, R)
q_4	–	(q_5, B, L)	(h, B, R)
q_5	(q_5, a, L)	(q_5, b, L)	(q_0, B, R).

Table 13.17 *Transition table for M*

TM action for the string $w = bab$

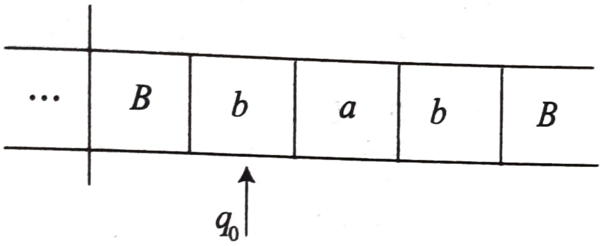


Figure 13.40. *The Tape*

ID is :

$$
\begin{aligned}
q_0bab &\vdash Bq_2ab \\
&\vdash Baq_2b \\
&\vdash Babq_2 \\
&\vdash Baq_4bB \\
&\vdash Bq_5aBB \\
&\vdash q_5BaBB \\
&\vdash Bq_0aBB \\
&\vdash BBq_1BB \\
&\vdash Bq_3BBB \\
&\vdash BBhBBB.
\end{aligned}
$$

Since the final state h is reached, the string is accepted.

EXAMPLE 13.10.10: Design a TM that recognises the language consisting of all strings of 0's, whose length is a power of 2. i.e. $L = \{0^{2^m} | m \geq 0\}$.

Design Strategy: Let q_0 be the start state and the tape head point to the first symbol of the string to be scanned.

Step-1: In state q_0 on input symbol 0, replace 0 by B and move the tape head towards right.

Step-2: Scan from left to right across the tape, replacing every 0 by x, whenever it is required.

Step-3: If the string contains a single 0, then the string is accepted by the TM.

Step-4: If the string contains more than a single 0 and the number of 0's are odd, then the TM goes to the state q_5 and the string is rejected.

Thus, the TM is,

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, h) \quad \text{i.e.,}$$

$$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, h\}, \{0\}, \{0, x\}, \delta, q_0, B, h),$$

where δ is given by:

Transition diagram:

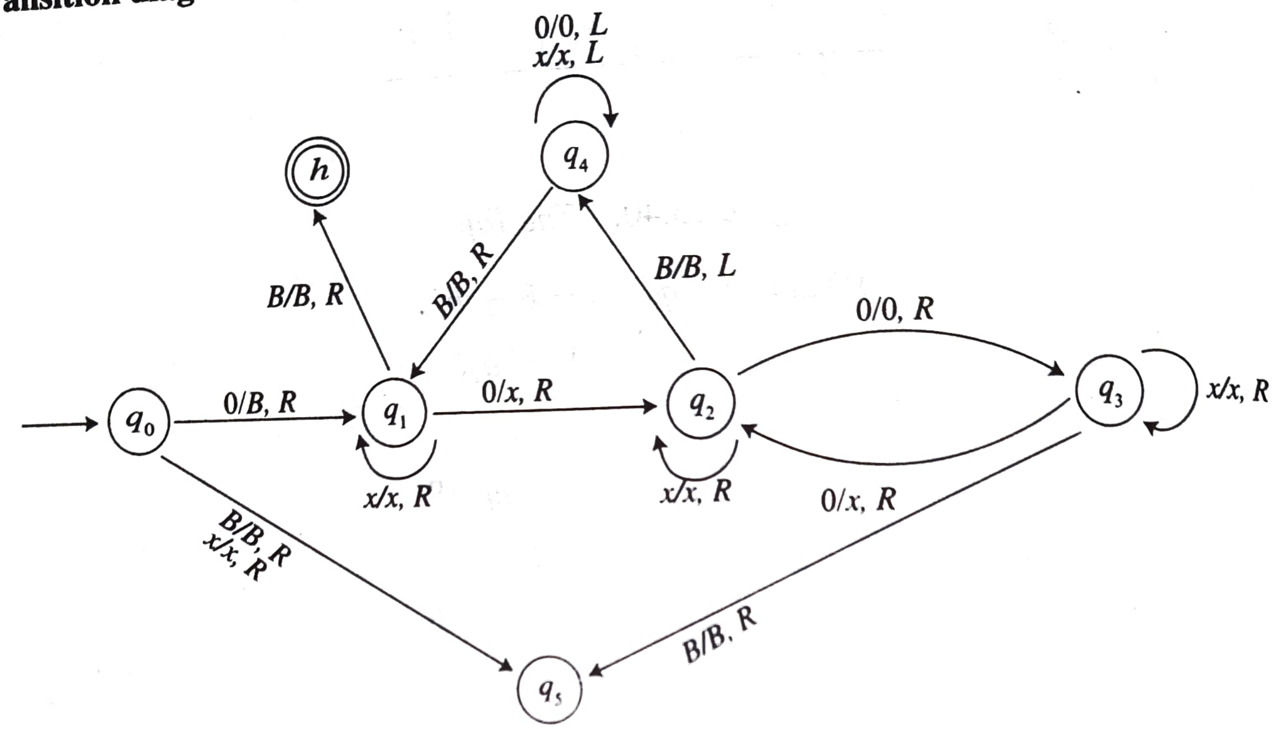


Figure 13.41. *Transition Diagram for M*

Transition Table:

Tape Symbol / States	0	x	B
q_0	(q_1, B, R)	(q_5, x, R)	(q_5, B, R)
q_1	(q_2, x, R)	(q_1, x, R)	(h, B, R)
q_2	(q_3, o, R)	(q_2, x, R)	(q_4, B, L)
q_3	(q_2, x, R)	(q_3, x, R)	(q_5, B, R)
q_4	(q_4, o, L)	(q_4, x, L)	(q_1, B, R)

Table 13.18 *Transition table for M.*

TM action for the string $w = 0000$.

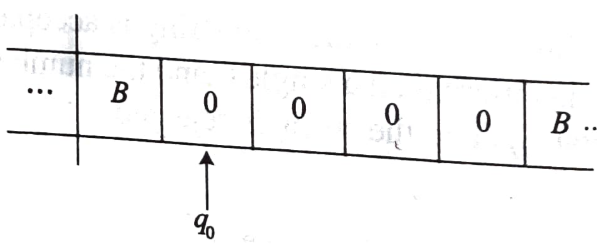


Figure 13.42. *The Tape*

$$
\begin{aligned}
\text{ID is:}\quad q_0 0000 &\vdash Bq_1 000\\
&\vdash Bxq_2 00\\
&\vdash Bx0q_3 0\\
&\vdash Bx0xq_2\\
&\vdash Bx0q_4 x\\
&\vdash Bxq_4 0x\\
&\vdash Bq_4 x0x\\
&\vdash q_4 Bx0x\\
&\vdash Bq_1 x0x\\
&\vdash Bxq_1 0x\\
&\vdash Bxxq_2 x\\
&\vdash Bxxxq_2\\
&\vdash Bxxq_4 xB\\
&\vdash Bxq_4 xxB\\
&\vdash Bq_4 xxxB\\
&\vdash q_4 BxxxB\\
&\vdash Bq_1 xxxB\\
&\vdash Bxq_1 xxB\\
&\vdash Bxxq_1 xB\\
&\vdash Bxxxq_1 B\\
&\vdash BxxxBh.
\end{aligned}
$$

Since the final state h is reaached, the string 0000 is accepted.

Theorem 1 Every regular expression has a TM that accepts it.

Proof. We know that there is a DFA for every regular language. Draw a DFA for the language. Change the edge labels from a and b to $a/a,R$ and $b/b,R$. Add a halt state. Take away the accept status of the accept states and add an edge from each one, labelled $B/B, N$, to the halt state.

Construction of a TM from the given DFA

The following are the steps required:

Step-1: Change the edge labels of a given DFA from a and b to $a/a, R$ and $b/b, R$.

Step-2: Add a halt state.

Step-3: Take all the 'accept' states of DFA and add an edge from each one, labelled $B/B, N$, to the halt state.

EXAMPLE 13.10.11: Construct a TM to accept the language containing strings of 0's and 1's ending with 00.

Solution: The transition diagram of DFA, which accepts the language consisting of strings of 0's and 1's ending with the string 00, is shown in figure 13.43.

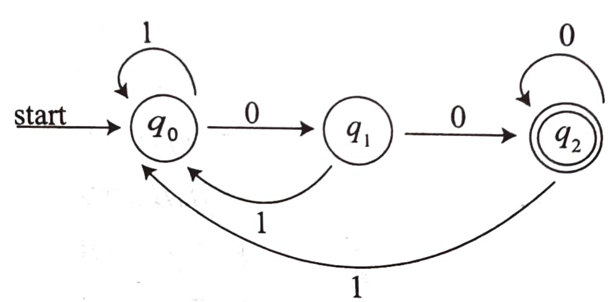


Figure 13.43. *DFA to Accept L* = (0 + 1)*00

Change the edge labels of the above DFA from 0 and 1 to 0/0,*R* and 1/1,*R*. Add the halt state h and add an edge lablelled $B/B, N$ from q_2 to h.

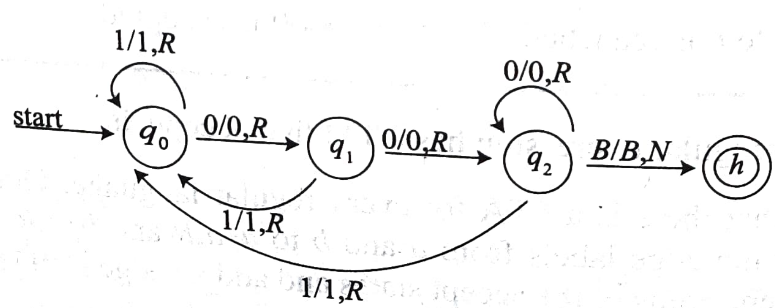


Figure 13.44. *Transition Diagram for TM to accept L* = (0 + 1)*00

Thus the TM is, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ i.e.,

$$M = (\{q_0, q_1, q_2, h\}, \{0, 1\}, \{0, 1\}, \delta, q_0, B, h).$$

The **transition** δ is given in table 13.19:

States \ Tape	0	1	B
q_0	$(q_1, 0, R)$	$(q_0, 1, R)$	–
q_1	$(q_2, 0, R)$	$(q_0, 1, R)$	–
q_2	$(q_2, 0, R)$	$(q_0, 1, R)$	(h, B, N)

Table 13.19 *Transition table for M*

EXAMPLE 13.10.12: Construct a TM over the alphabet {0,1}, that contains set of strings of 0's and 1's except those containing the substring 001.

Solution: The transition diagram of DFA, which accepts the language consisting of strings of 0's and 1's except those containing the substring 001, is shown in figure 13.45:

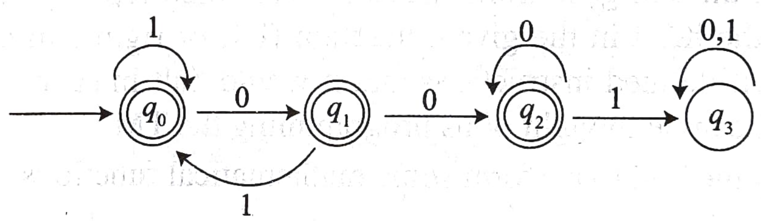


Figure 13.45. *DFA to accept all Strings of 0's & 1's except those containing the Substring 001*

Change the edge labels of the above DFA from 0 and 1 to 0/0,R and 1/1,R. Add the halt state h and add an edge labelled $B/B, N$ from q_0 to h, q_1 to h and q_2 to h.

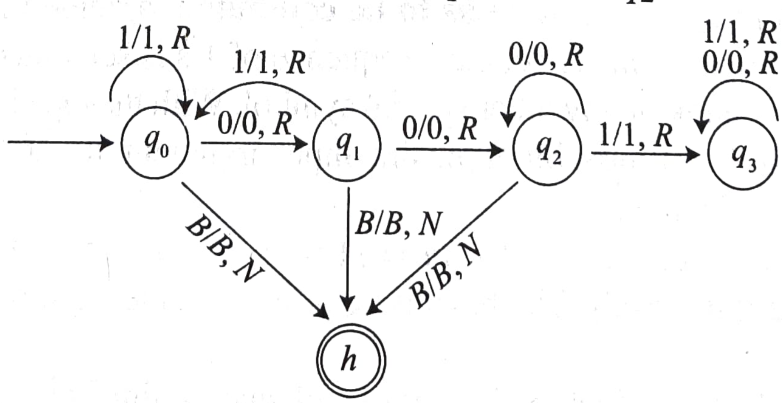


Figure 13.46. *TM to accept all Strings of 0's and 1's except those containing the Substring 001*

Thus, the TM is,

$$M = (Q, \Sigma, \Gamma, \delta, q_0, B, h) \quad \text{i.e.,}$$

$$M = (\{q_0, q_1, q_2, q_3, h\}, \{0, 1\}, \{0, 1\}, \delta, q_0, B, h).$$

The **transition** δ is given in table 13.20.

States \ Tape	0	1	B
q_0	$(q_1, 0, R)$	$(q_0, 1, R)$	(h, B, N)
q_1	$(q_2, 0, R)$	$(q_0, 1, R)$	(h, B, N)
q_2	$(q_2, 0, R)$	$(q_3, 1, R)$	(h, B, N)
q_3	$(q_3, 0, R)$	$(q_3, 1, R)$	—

Table 13.20 *Transition table for M*

13.10.2 *TM as a Computer of Functions*

Instructions of TM means that if the TM is in the current state and the head is pointing to a given symbol, then the TM goes from one state to another, replaces the symbol by a new symbol and moves the head in the given direction (left or right). In some situations, the TM will have no well-defined instructions and it would halt in such a situation. Defining instructions of a TM can be thought of as programming the TM.

For programming the TM to perform some mathematical functions, several conventions are commonly used:

a. **Interpretation of the symbols recorded on the tape** To describe how to interpret the ones and zeros appearing on the tape as numbers, the number is represented in unary notation. It means that the non-negative integer number n is represented by using successive 1's. For instance, 2 is 11, 3 is 111, 4 is 1111 and so on.

b. **If a function** $f(n_1, n_2, \ldots, n_k)$ **has to be computed**, assume that initially the tape consists of $n_1, n_2, \ldots, n_k$. Also, each sequence of 1's is separated from the previous one by a single blank or any other special symbol. With the tape head initially located at the rightmost or leftmost bit of the first input argument, the state of the TM is some initially specified value.

   The TM is said to have computed $m = f(n_1, n_2, \ldots, n_k)$, when it halts and the tape consists of the final result. The head is positioned at the rightmost or leftmost bit of the result.

c. To ascertain when the machine is started and when it finishes, some special symbols are included in the tape.

EXAMPLE: TM to compute the function $m = multiply(n_1, n_2) = n_1 \times n_2$. If $n_1 = 3$ and $n_2 = 2$, then the tape in the beginning of computation is given in figure 13.47.

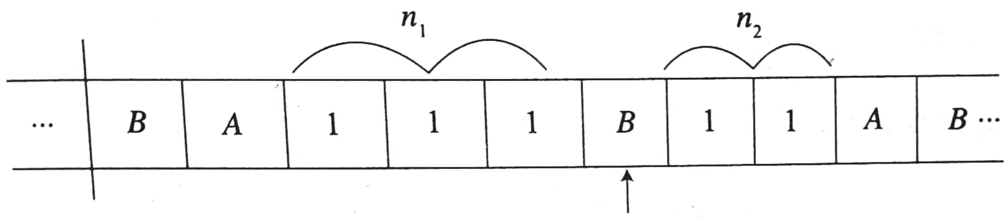


Figure 13.47. *The Tape in the Beginning of the Computation*

After the computation is over i.e $n_1 \times n_2 = 3 \times 2 = 6$, the TM halts, with its tape looking as shown in figure 13.48.

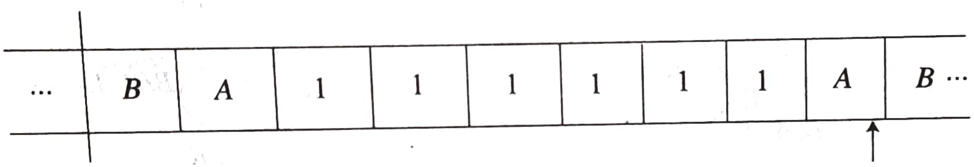


Figure 13.48. *The Tape After Computation*

EXAMPLE 13.10.13: Construct a TM that finds the difference of two natural numbers:

Design Strategy: We are to design a TM to compute the function $m = \text{SUB}(n_1, n_2)$ such that:

$$\text{SUB}(n_1, n_2) = \begin{cases} n_1 - n_2 & \text{if} \quad n_1 \geq n_2 \\ 0 & \text{if} \quad n_1 < n_2 \end{cases}$$

Initially, the unary sequence is written onto the tape with two numbers n_1 and n_2 separated by a blank symbol B. The symbol A on tape indicates the beginning and end points of the sequence. The tape head initially points to the symbol B, with q_0 as the start state. The initial configuration of the tape with $n_1 = 3$ and $n_2 = 2$ is shown in figure 13.49.

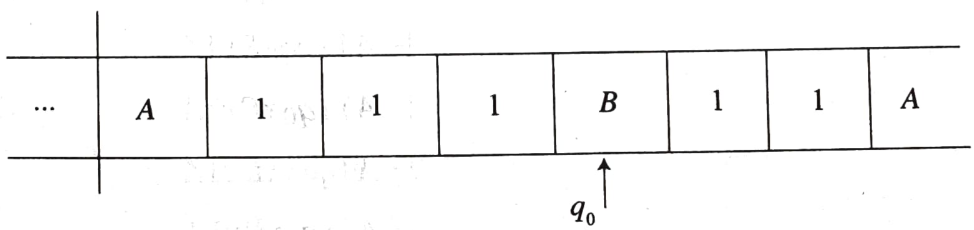


Figure 13.49. *The Tape's Initial Configuration*

Thus, the TM is $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ i.e.,

$$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, h\}, \{0\}, \{0, 1, x, A\}, \delta, q_0, B, h),$$

where δ is given by:

Transition diagram:

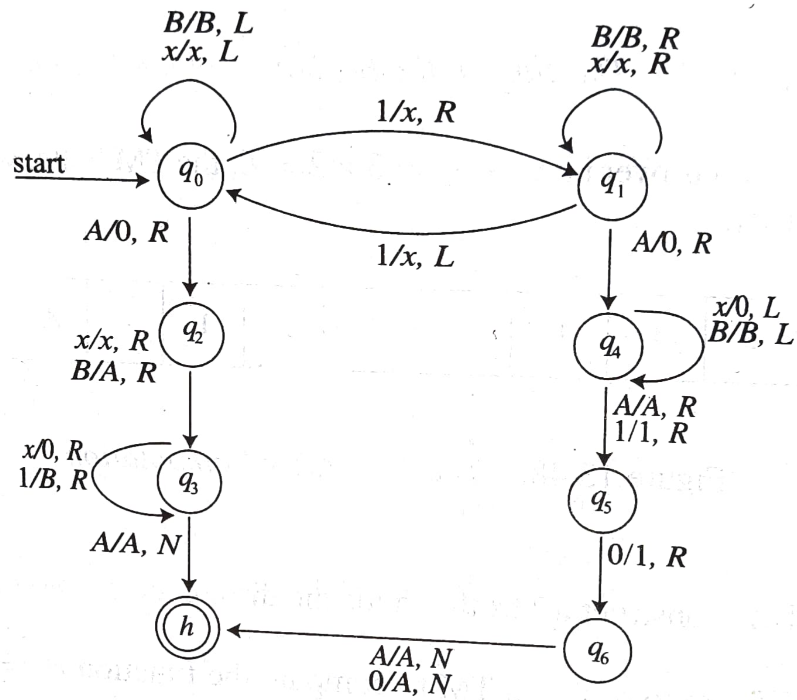


Figure 13.50. *TM to Compute SUB Function*

TM action for the input $n_1 = 3$ and $n_2 = 2$

ID is: $A111q_0B11A \vdash A11q_01B11A$

$\vdash A11xq_1B11A$

$\vdash A11xBq_111A$

$\vdash A11xq_0Bx1A$

$\vdash A11q_0xBx1A$

$\vdash A1q_01xBx1A$

$\vdash A1xq_1xBx1A$

$\vdash A1xxq_1Bx1A$

$$\vdash A1xxBq_1x1A$$
$$\vdash A1xxBxq_1 1A$$
$$\vdash A1xxBq_0xxA$$
$$\vdash A1xxq_0BxxA$$
$$\vdash A1xq_0BxxA$$
$$\vdash A1q_0xxBxxA$$
$$\vdash Aq_0 1xxBxxA$$
$$\vdash Axq_1xxBxxA$$
$$\vdash Axxq_1xBxxA$$
$$\vdash Axxxq_1BxxA$$
$$\vdash AxxxBq_1xxA$$
$$\vdash AxxxBxq_1xA$$
$$\vdash AxxxBxxq_1A$$
$$\vdash AxxxBxq_4x0$$
$$\vdash AxxxBq_4x00$$
$$\vdash Axxxq_4B000$$
$$\vdash Axxq_4xB000$$
$$\vdash Axq_4x0B000$$
$$\vdash Aq_4x00B000$$
$$\vdash q_4A000B000$$
$$\vdash Aq_5000B000$$
$$\vdash A1q_600B..$$
$$\vdash A1Ah..$$

Since the final state h is reached, the TM halts resulting into a difference of 3 and 2 as 1.

EXAMPLE 13.10.14: Construct a TM that finds the sum of two natural numbers.
Design Strategy: We are to design a TM to compute the function

$$M = \text{Sum}(n_1, n_2) = n_1 + n_2.$$

Initially, the unary sequence is written onto the tape with two numbers n_1 and n_2 separated by a symbol B, the symbol A on tape indicates the beginning and end points of the sequence. The tape head initially points to the leftmost bit of n_1. The initial configuration of the tape with $n_1 = 4$ and $n_2 = 3$ is shown in figure 13.51.

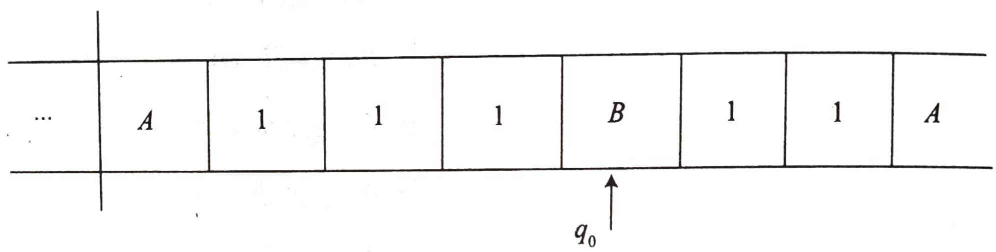


Figure 13.51. *The Tape's Initial Configuration*

Thus the TM is, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ i.e.,

$$M = (\{q_0, q_1, q_2, h\}, \{1\}, \{1, A\}, \delta, q_0, B, h)$$

where δ is given by:

Transition diagram:

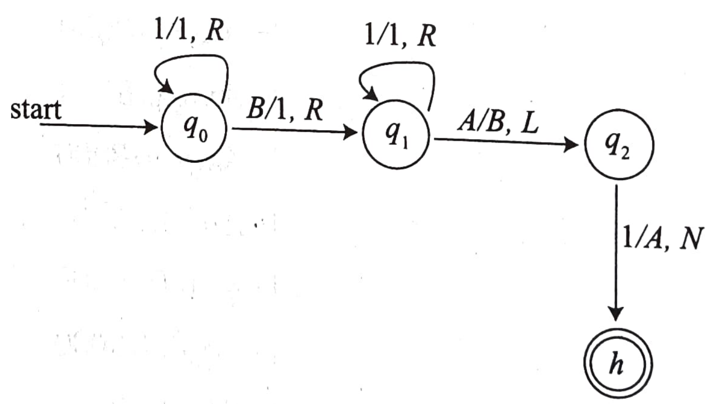


Figure 13.52. *TM to Compute the Sum Function*

TM action for the input $n_1 = 4$ and $n_2 = 3$

ID is: $\quad Aq_01111B111A \vdash A1q_0111B111A$

$$\vdash A11q_011B111A$$

$$\vdash A111q_01B111A$$

$\vdash A1111q_0B111A$

$\vdash A11111q_1111A$

$\vdash A111111q_11!A$

$\vdash A1111111q_11A$

$\vdash A11111111q_1A$

$\vdash A1111111q_21B$

$\vdash A1111111Ah..$

Since the final state h is reached, the TM halts resulting into the sum of 4 and 3 as 7.

EXAMPLE 13.10.15: Construct a TM to compute $f(n) = n + 2$.

Design Strategy: We are to design a TM to compute the function $m = f(n) = n + 2$.

Initially, the unary sequence for n is written onto the tape with the symbol A on both the ends of n, indicating the beginning and end points of the sequence. The tape head points to the leftmost bit of n, with q_0 as the start state. The initial tape configuration with n is shown in figure 13.53.

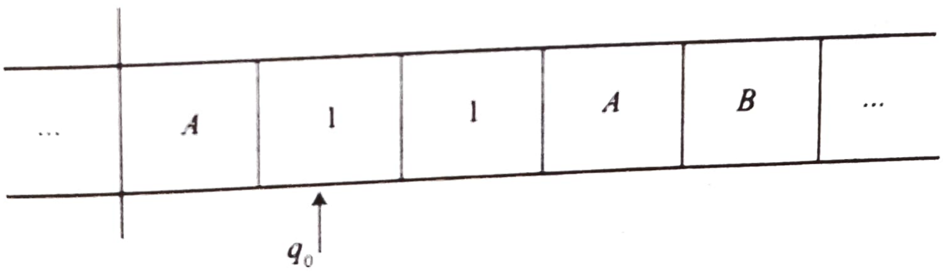


Figure 13.53. *The Tape's Initial Configuration*

Thus the TM is, $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ i.e.,

$$M = (\{q_0, q_1, q_2, h\}, \{1\}, \{1, A\}, \delta, q_0, B, h),$$

where δ is given by

Transition diagram:

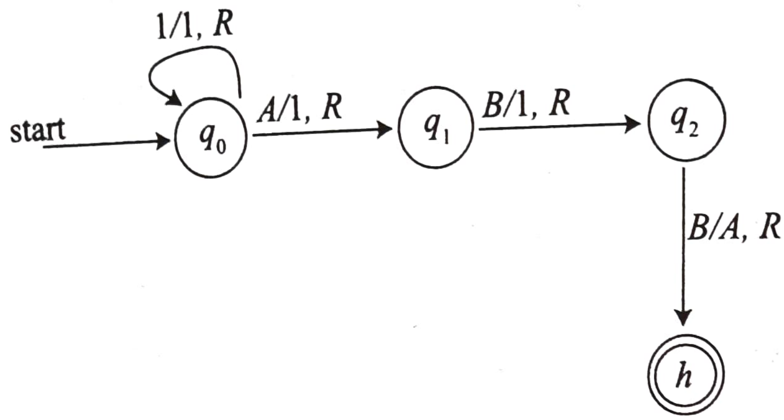


Figure 13.54. *TM to Compute $f(n)$*

TM action for input $n = 2$

$$\text{ID is:} \quad Aq_011A \vdash A1q_01A$$
$$\vdash A11q_0A$$
$$\vdash A111q_1B$$
$$\vdash A1111q_2B$$
$$\vdash A1111Ah$$

Since the final state h is reached, the TM halts, resulting into a sum of 2 and 2 as 4.

EXAMPLE 13.10.16: Design a TM to compute $\max(n_1, n_2)$.

Design Strategy: We are to design a TM to compute the finction $m = \max(n_1, n_2)$ such that:

$$\max(n_1, n_2) = \begin{cases} n_1 & \text{if} \quad n_1 \geqslant n_2 \\ n_2 & \text{if} \quad n_1 < n_2. \end{cases}$$

Initially, the unary sequence is written onto the tape with two numbers n_1 and n_2, separated by a blank symbol B. The symbol A on tape indicates the beginning and end points of the sequence. The head initially points to the symbol B, with q_0 as the start state. The initial tape configuration with $n_1 = 1$ and $n_2 = 2$ is shown in figure 13.55.

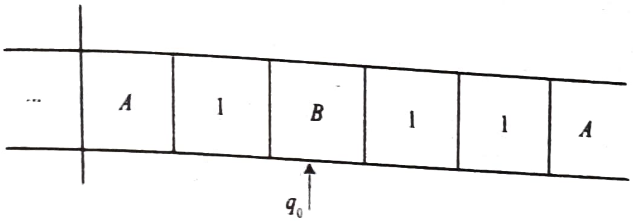


Figure 13.55. *The Tapes Initial Configuration*

Transition diagram:

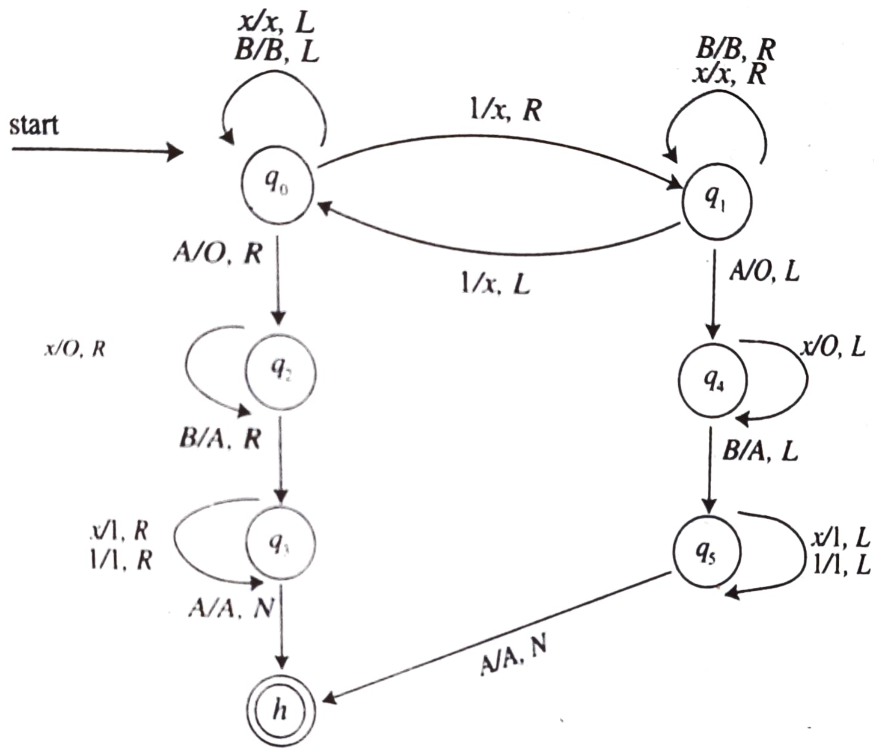


Figure 13.56. *TM to Compute* $max(n_1, n_2)$

TM action for input $n_1 = 1$ **and** $n_2 = 2$: With $\Gamma = \{0, 1, A, x\}$, $\Sigma = \{0, 1\}$ and $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, h\}$

$$
\begin{aligned}
\text{ID is:}\quad Aq_01B11A &\vdash Axq_1B11A \\
&\vdash AxBq_111A \\
&\vdash Axq_0Bx1A \\
&\vdash Aq_0xBx1A \\
&\vdash q_0AxBx1A \\
&\vdash 0q_2xBx1A \\
&\vdash 00q_2Bx1A \\
&\vdash 00Aq_3x1A \\
&\vdash 00A1q_31A \\
&\vdash 00A11q_3A \\
&\vdash ..A11Ah
\end{aligned}
$$

Since the final state h is reached, the TM halts, resulting into a maximum of 1 and 2 as 2.

EXAMPLE 13.10.17: Construct a TM that finds the product of two natural numbers.

Design Strategy: We are to design a TM to compute the function

$$m = multiply(n_1, n_2) = n_1 \times n_2.$$

Initially, the unary sequence is written onto the tape with two numbers n_1 and n_2, separated by a blank symbol B. The symbol A on the tape indicates the beginning and end points of the sequence. The tape head initially points to symbol B, with q_0 as the start state. The initial configuration of the tape with $n_1 = 2$ and $n_2 = 3$ is shown in figure 13.57.

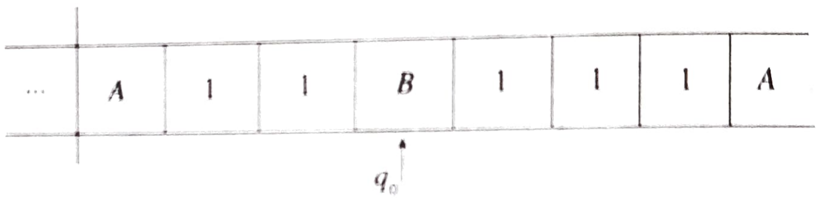


Figure 13.57. *Tape's Initial Configuration*

Thus, the TM is $M = (Q, \Sigma, \Gamma, \delta, q_0, B, h)$ i.e.,

$$M = (\{q_0, q_1, q_3, q_4, q_5, q_6, h\}, \{1\}, \{0, 1, A, x\}, \delta, q_0, B, h),$$

where δ is given by **Transition diagram:**

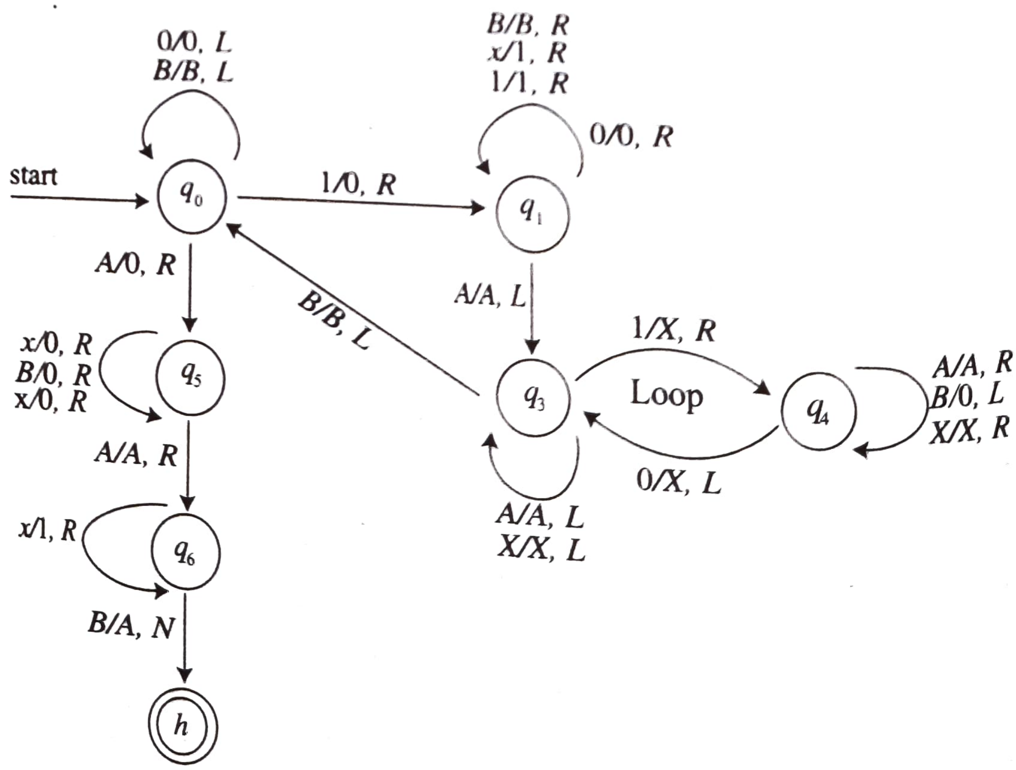


Figure 13.58. *TM to Compute the Multiply Function*

TM action for the input $n_1 = 2$ and $n_2 = 3$

i. $A11\underset{\substack{\uparrow\\ q_0}}{B}111A$

ii. $A10\underset{\substack{\uparrow\\ q_1}}{B}111A$

iii. Move right until $A10B111\underset{\substack{\uparrow\\ q_1}}{A}$

iv. After 3 operations we have $A10B11xA\underset{\substack{\uparrow\\ q_4}}{0}$

v. $A10B11x\underset{\substack{\uparrow\\ q_3}}{A}x$

vi. Move left until $A10B1\underset{\substack{\uparrow\\ q_3}}{1}xAx$

vii. Replace 1 by x and move right until $A10B1xxAx\underset{\substack{\uparrow\\ q_4}}{0}$

viii. Replace 0 by x and move left until $A10B\underset{\substack{\uparrow\\ q_3}}{1}xxAxx$

ix. Replace 1 by x and move right until $A10BxxxAxx\underset{\substack{\uparrow\\ q_4}}{0}$

x. Replace 0 by x and move left until $A10\underset{\substack{\uparrow\\ q_3}}{B}xxxAxxx$

This is a loop writing n 1's on the R.H.S. of the rightmost A.

xi. Move left until $A\underset{\substack{\uparrow\\ q_0}}{1}0BxxxAxxx$

(w to x) is a loop writing 'n' once on the RHS of the right most A.

xii. Replace 1 by 0 and move right, changing x's to 1's until

$$A00B111\underset{\substack{\uparrow\\ q_1}}{A}xxx$$

xiii. Now enter the loop again to write n once more, over the zeros on the right of the rightmost A until

$$A00\underset{\substack{\uparrow\\ q_3}}{B}xxxAxxxxxx$$

xiv. Now move left until

$$\underset{\substack{\uparrow\\ q_0}}{A}\,00BxxxAxxxxxx$$

xv. Move right until

$$000\,\underset{\substack{\uparrow\\ q_5}}{B}\,xxxAxxxxxx$$

xvi. Move right until

$$0000000\,\underset{\substack{\uparrow\\ q_5}}{A}\,xxxxxx$$

xvii. Move right until x replaces 1 to get the final output.

$$0000000A111111A$$

EXAMPLE 13.10.18: Construct a TM to compute n^2.
Design Strategy: We are to design a TM to compute the function

$$m = \text{square}(n) = n \times n.$$

Consider the initial configuration of tape with $n = 2$, as shown in figure 13.59.

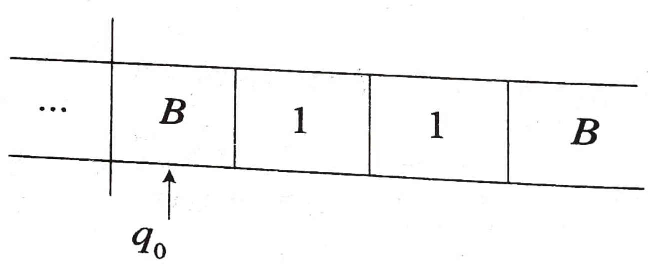


Figure 13.59. *Tape's Initial Configuration-1*

First convert the above tape into the form, shown in figure 13.60.

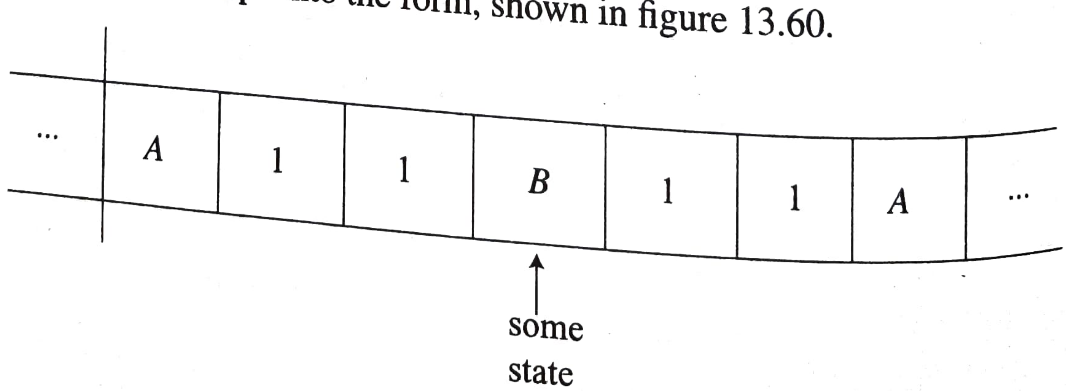


Figure 13.60. *Tape's Initial Configuration-2*

Thus, we require two TMs – one to convert from configuration-1 to configuration-2 and another to multiply n by n.

The transition diagram of TM_1 to convert from configuration-1 to configuration-2 is shown in figure 13.61.

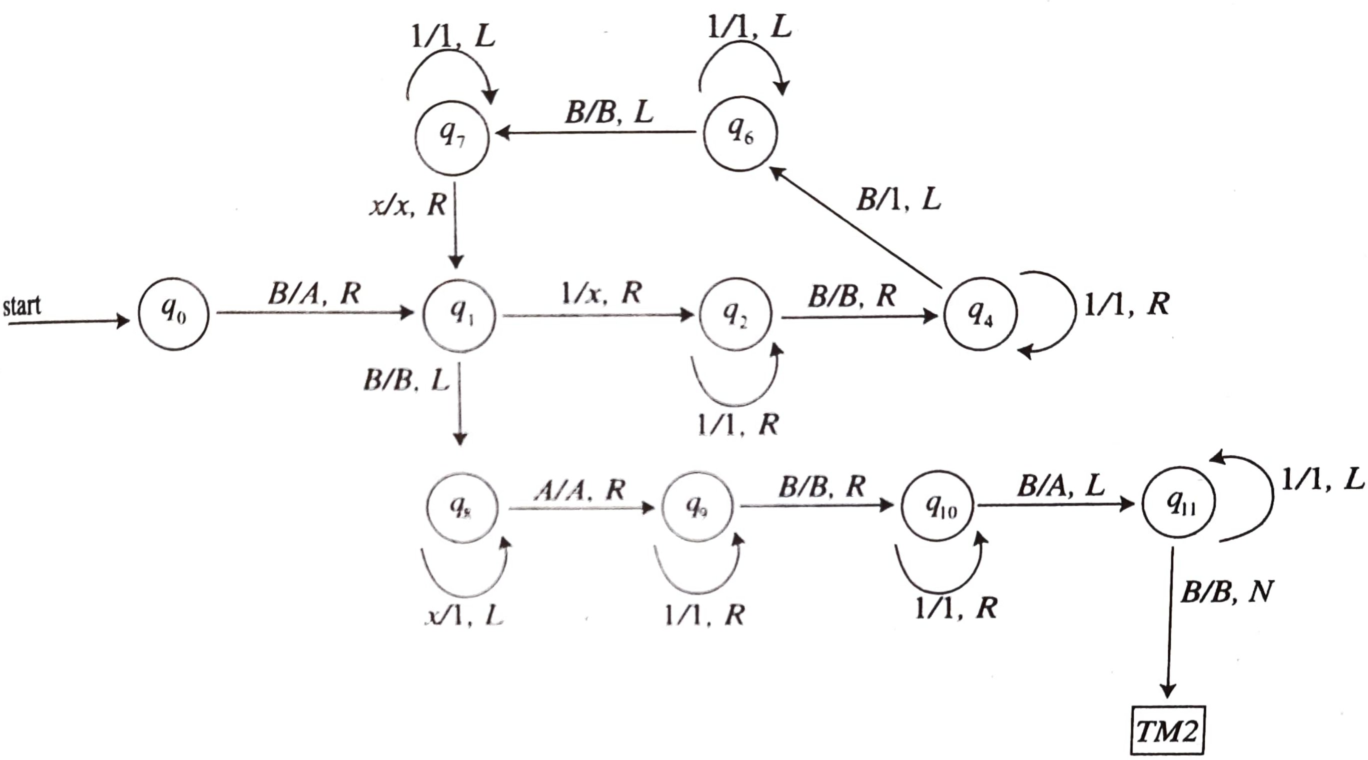


Figure 13.61. *Transition Diagram for TM1*

TM action for input $n = 2$ with $Q = \{q_0, q_1, q_2, q_4, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}$, $\Sigma = \{1\}$ and $\Gamma = \{1, X, A\}$.

ID is: $q_0B11 \vdash Aq_1 11$

$\vdash Axq_2 1$

$\vdash Ax1q_2$

$\vdash Ax1Bq_4$

$\vdash Ax1q_6B1$

$\vdash Axq_7 1B1$

$\vdash Aq_7x1B1$

$\vdash Axq_1 1B1$

$\vdash Axxq_2B1$

$$\vdash AxxBq_4 1$$
$$\vdash AxxB1q_4$$
$$\vdash AxxBq_6 11$$
$$\vdash Axxq_6 B11$$
$$\vdash Axq_7 xB11$$
$$\vdash Axxq_1 B11$$
$$\vdash Axq_8 xB11$$
$$\vdash Aq_8 x1B11$$
$$\vdash q_8 A11B11$$
$$\vdash Aq_9 11B11$$
$$\vdash A1q_9 1B11$$
$$\vdash A11q_9 B11$$
$$\vdash A11Bq_{10} 11$$
$$\vdash A11B1q_{10} 1$$
$$\vdash A11B11q_{10}$$
$$\vdash A11B1q_{11} 1A$$
$$\vdash A11Bq_{11} 11A$$
$$\vdash A11q_{11} B11A$$
$$\vdash A11\underset{\substack{\uparrow \\ TM_2}}{B}11A$$

We are now in the configuration-2. Thus, TM_2 is called to multiply n by n (TM_2 is already discussed in example 13.10.17).

13.11 Exercises

1. Define a TM. Present the formal definition of a TM.
2. Explain, with a neat diagram, the components of a TM.
3. Differentiate between FA/PDA vs. TM with respect to:

a. tape and head
b. halt state and final states.

4. Explain the different ways to describe a TM, with examples.
5. What is an infinite loop in TM? Explain with an example.
6. Present the formal definition to the ID of a TM.
7. Explain, with examples, the different string classes of TM.
8. Design a TM to accept the language $L = \{ww^R | w \in (a+b)^*\}$.
9. Design a TM to recognise the language of all strings of even length, over $\{a, b\}$.
10. Design a TM that accepts the language of all strings, which contain 101 as a substring over $\{0, 1\}$.
11. Design a TM that accepts twice as many 0's as 1's, over $\{0, 1\}$.
12. Design a TM to compute the function $f(n) = 2n$.
13. Design a TM to compute the function n mod 2.
14. Construct a TM that will search for and locate a symbol A on its tape (if there is one) and then halts.
15. Present a TM that decides the following languages, over $\{a, b\}$:
    (i) ϕ
    (ii) $\{a\}^*$
    (iii) $\{a^*ba^*b\}$.
16. Identify the language accepted by the TM given in figure 13.62.

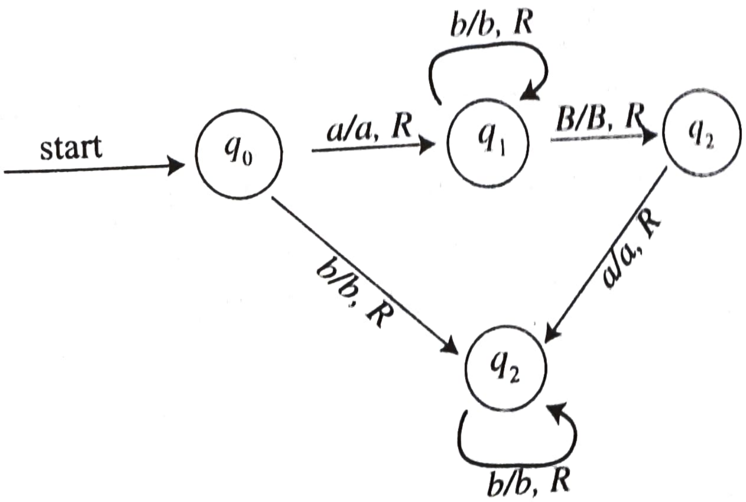


Figure 13.62. *Transition Diagram*