Turing Machines-Introduction

-- Sakshi Surve

Limitations of FA and FSM

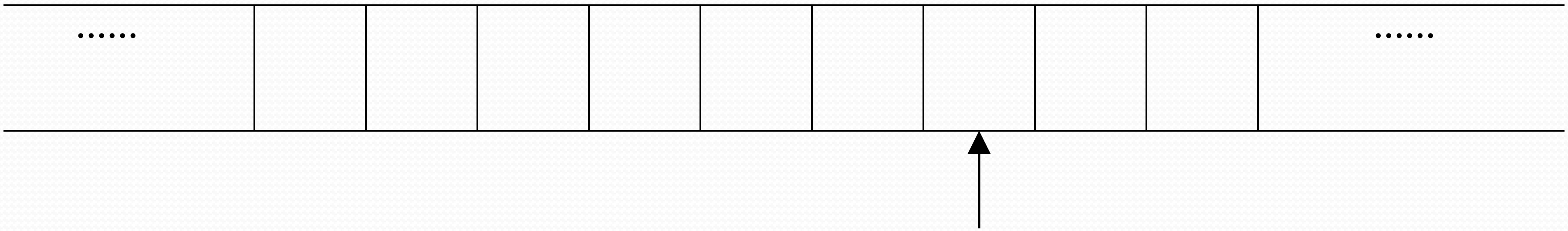
- Because an FA / FSM only has a finite number of states (say N states), it can only "remember" or "count" N things.
- There are therefore some strings that an FA / FSM cannot recognise. e,g. strings consisting of an equal number of 0s and 1s, strings with balanced parentheses such as ((()())).
- So, a more complex theoretical model is required, that can better incorporate the concept of "memory". One such model is the Turing machine.

TM consists of ….

- A head which can read or write a symbol or move to Left or Right or may stay in the same position.
- An infinite tape on which the symbols can be written.
- Finite set of symbols
- Finite set of states in one of which the machine can reside at a time.

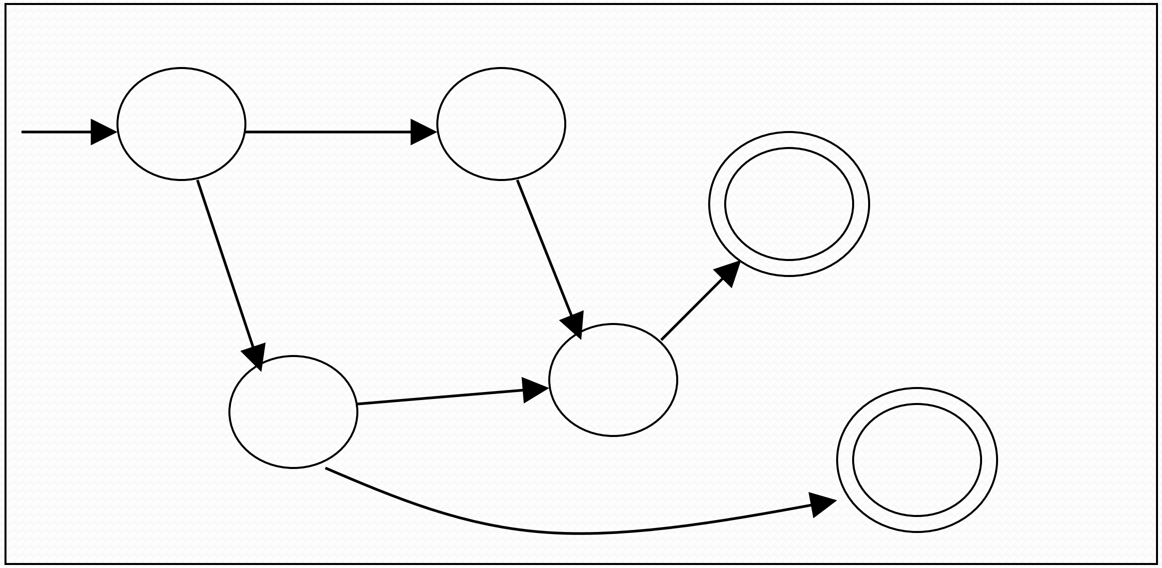
A Turing Machine :

Tape



Read-Write head

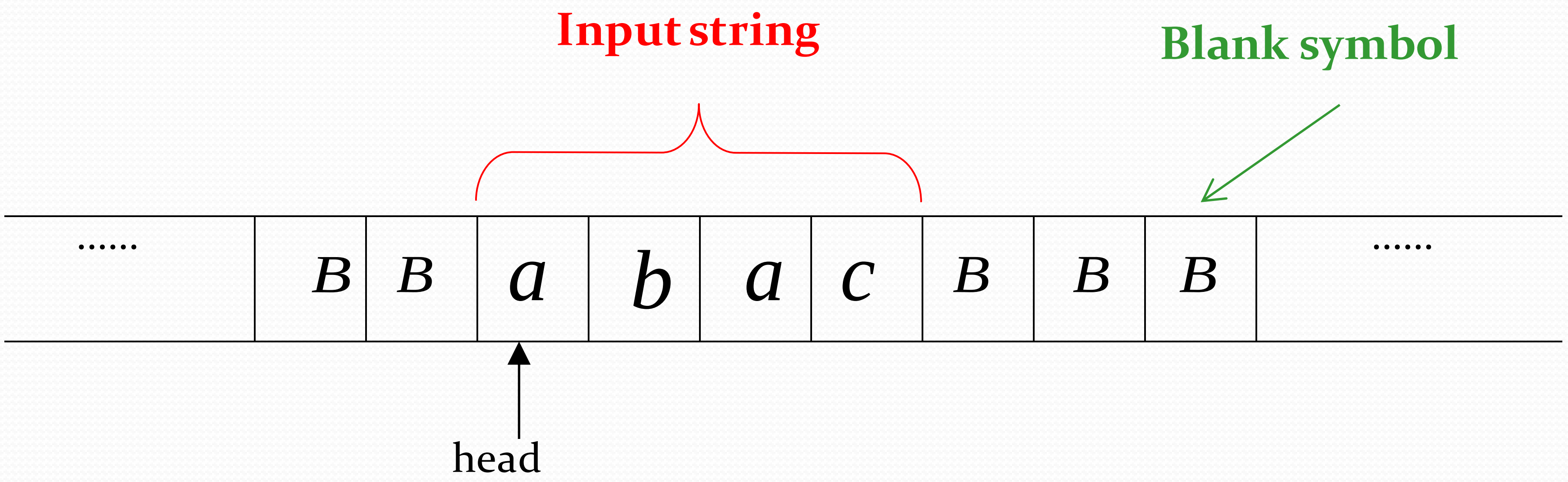
Control Unit



A Turing Machine (TM) has finite-state control (like PDA), and an infinite read-write tape.

- The tape serves as both input and unbounded storage device.
- The tape is divided into cells, and each cell holds one symbol from the tape alphabet.
- There is a special blank symbol B.
- At any instant, all but finitely many cells hold B....and the input string has B to its both sides
- Tape head sees only one cell at any instant.
- The contents of this cell and the current state determine the next move of the TM.

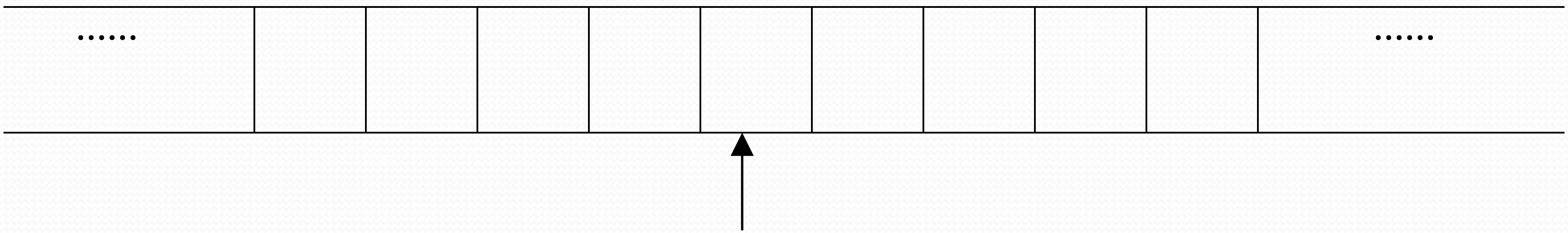
The Input String :



Head starts at the leftmost position of the input string

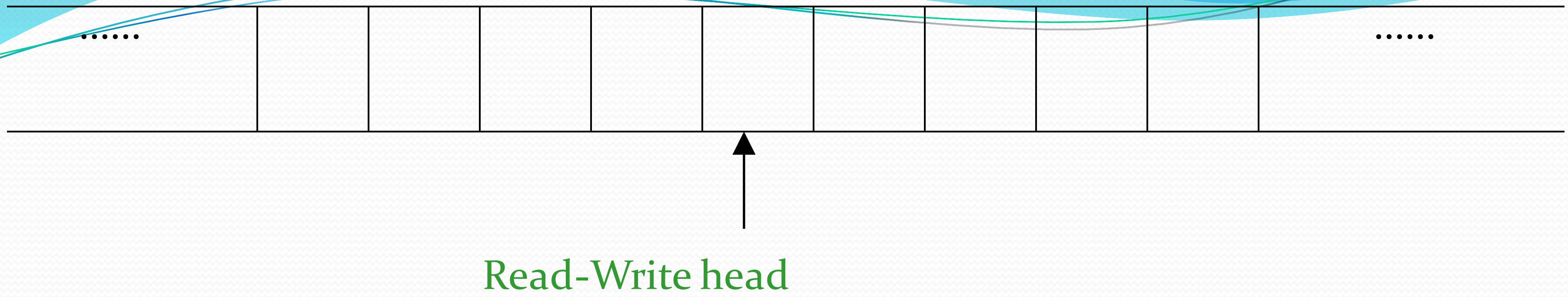
The Tape :

No boundaries -- infinite length



Read-Write head

The head moves Left or Right or stays at the same position

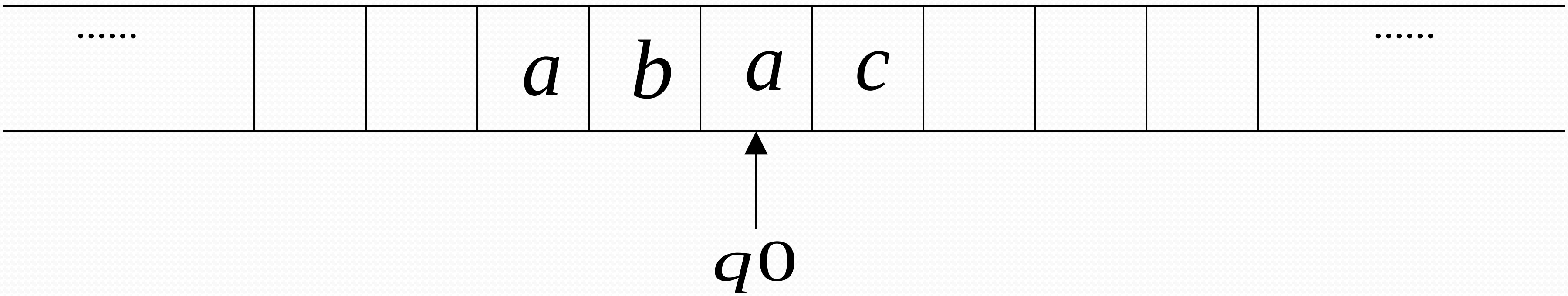


The head at each transition (time step):

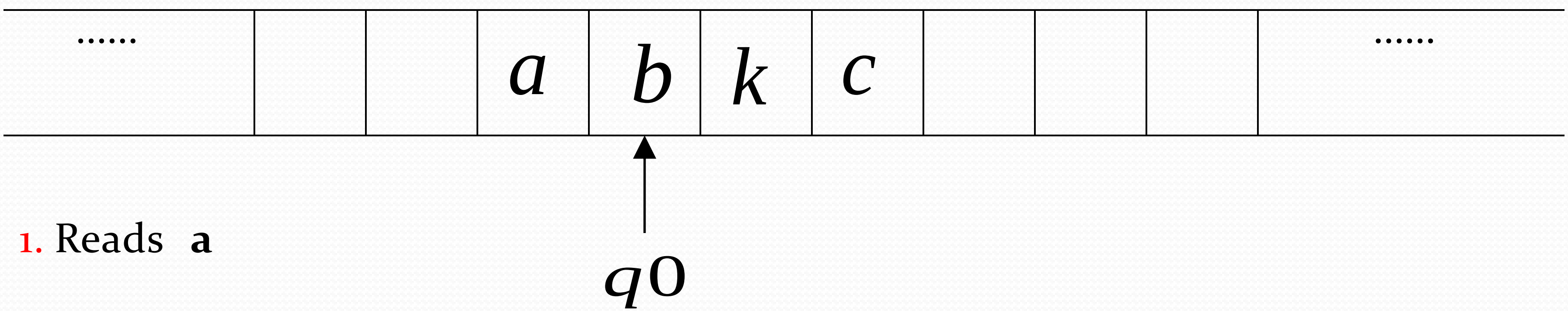
1. **Reads a symbol**
2. **Writes a symbol**
3. **Moves Left or Right or stays at same place**

Example:

Time 0



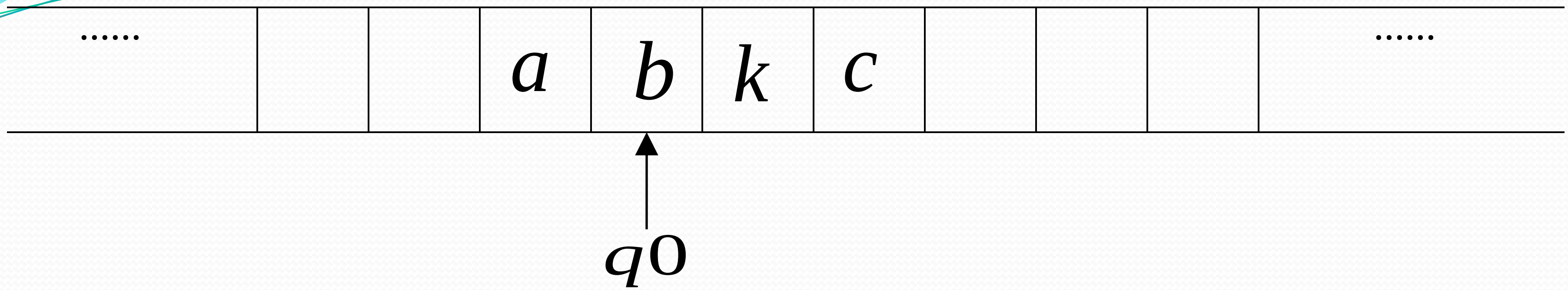
Time 1



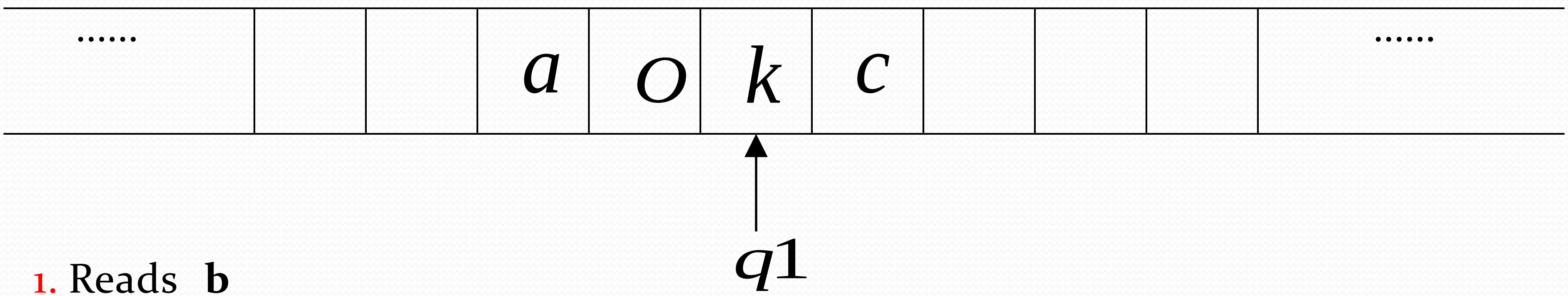
1. Reads **a**
2. Writes **k**
3. Moves Left

$$(q0, k, L)$$

Time 1



Time 2



1. Reads **b**
2. Writes **O**
3. Moves Right

$$(q1, k, R)$$

- A Turing Machine performs following three actions before the next interval begins :
  - Square being read is erased and is replaced by another symbol
  - The internal state is changed
  - The head moves one square left, right or stays at the same position

This can be summarized in a string of the form

$$(q_0 , a) \rightarrow (q_1 , b , L)$$

Where,

q_0 = Current state

a = Input symbol read in q_0

q_1 = Next state

b = symbol ‘a’ has to be replaced with

L = The direction of movement of the tape head

Differences between finite automata and Turing machine

1. A Turing machine can both write on the tape and read from it.
2. The read-write head can move both to the left and to the right.
3. The tape is infinite.
4. The special states for rejecting and accepting take immediate effect.

Representation of a Turing Machine :

Formally, a basic TM is defined as $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, where

Q	is a finite set of *states* (of the control)
Γ	is a finite set of allowable *tape symbols*
$B \in \Gamma$	is a special *blank symbol*
$\Sigma \subseteq (\Gamma - \{B\})$	is a finite set of *input symbols*
$q_0 \in Q$	is the *start* state
$F \subseteq Q$	is a finite set of final accepting states

and δ is the transition/action function, defined as

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, -\} \quad .$$

1. Design a TM to convert every occurrence of '11' to '10'   E = {0,1}

B 1 1 0 1 0 B

111000000

B 1 0 0 1 0 B

101000000

110011

B 0 1 1 0 1 0 0 1 B

100010

101

B 0 1 0 0 1 0 0 1 B

101

Transition Mapping Function :

	0	1	B
q0	(q0, 0, R)	(q1, 1, R)	(q2, B, N)
q1	(q0, 0, R)	(q0, 0, R)	(q2, B, N)
q2	Final State		

- Consider string

1 1 0 1 1
|
q0

1 1 0 1 1
|
q1

1 0 0 1 1
|
q0

1 0 0 1 1
|
q0

1 0 0 1 1
|
q1

1 0 0 1 0 B
|
q0

1 0 0 1 0 B
|
q2

Input String 1 1 0 1 1

Output String 1 0 0 1 0

- Consider string

1 0 0 1
|
q0

1 0 0 1
  |
  q1

1 0 0 1
    |
    q0

1 0 0 1
      |
      q0

1 0 0 1 B
        |
        q1

Input String 1 0 0 1

Output String 1 0 0 1

- **Tuples of the designed Machine:**

  $Q = \{ q0, q1, q2 \}$

  $E = \{0, 1 \}$

  $T = \{0, 1, B\}$

  $B = B$

  $q0 = q0$

  $F = \{ q2 \}$

- Answer to the questions related to designing a TM expects :
  - Transition Mapping Function
  - Example illustrated with steps
  - All tuples of the Machine designed

2. Design a TM to accept strings containing even number of ones. E = {0,1}

B 1 0 0 1 0 B

111000000

B 0 0 1 B

110011

1010

B 0 1 1 0 1 0 0 1 B

10

0

B 0 1 1 0 1 0 0 1 1 0B

1

Transition Mapping Function :

	0	1	B
q_0	$(q_0, 0, R)$	$(q_1, 1, R)$	(q_2, B, N)
q_1	$(q_1, 0, R)$	$(q_0, 1, R)$	--------

- Consider string

1 1 0 1 1
|
q0

1 1 0 1 1
|
q1

1 1 0 1 1
|
q0

1 1 0 1 1
|
q0

1 1 0 1 1
|
q1

1 1 0 1 1 B
|
q0

1 1 0 1 1 B
|
q2

String reaches the final state , so , string accepted

- Consider string

1 1 0 1

|

q0

1 1 0 1

|

q1

1 1 0 1

|

q0

1 1 0 1

|

q0

1 1 0 1 B

|

q1

String Rejected

- **Tuples of the designed Machine:**

  Q= { q0, q1, q2}

  E = {0, 1 }

  T = {0, 1, B}

  B = B

  q0 = q0

  F = { q2 }

3. Design a TM to find two's complement of a given binary input

B 1 1 0 1 0 B--------- i

B 0 0 1 1 0 B---------0

B 0 1 1 0 1 B --------i

B 1 0 0 1 1-----------0

Transition Mapping Function :

	0	1	B
q0	(q0, 0, R)	(q0, 1, R)	(q1, B, L)
q1	(q1, 0, L)	(q2, 1, L)	--------
q2	(q2, 1, L)	(q2, 0, L)	(q3, B, N)
q3	Final State		

- Consider string

B 1 1 0 B
|
q0

B 1 1 0 B
|
q0

B 1 1 0 B
|
q0

B 1 1 0 B
|
q0

B 1 1 0 B
|
q1

B 1 1 0 B
|
q1

B 1 1 0 B
|
q2

B 0 1 0 B
|
q2

B 0 1 0 B
|
q3

Input String : 1 1 0

Output String : 0 1 0

- Tuples of the designed Machine:

  Q= { q0, q1, q2, q3}

  E = {0, 1 }

  T = {0, 1, B}

  B = B

  q0 = q0

  F = { q3 }

4. Design a TM to check if a binary string contains ‘101’ as a substring

B 1 0 1 0 1 0 B

B 0 0 1 1 0 B

B 0 1 1 0 1 B

B 1 0 0 1 1 B

Transition Mapping Function :

	0	1	B
q0	(q0, 0, R)	(q1, 1, R)	--------
q1	(q2, 0, R)	(q1, 1, R)	--------
q2	(q0, 0, R)	(q3, 1, R)	--------
q3	(q3, 0, R)	(q3, 1, R)	(q4, B, N)
q4	Final State		

- Consider string

B 1 0 1 0 B
|
q0

1 0 1 0
|
q1

1 0 1 0
|
q2

1 0 1 0 B
|
q3

1 0 1 0 B
|
q3

1 0 1 0 B
|
q4

In the traversal, the input string reaches final state , so the string is accepted and is understood to contain '101' as a substring

- Consider string

B 1 0 0 0 B
|
q0
1 0 0 0
|
q1
1 0 0 0
|
q2
1 0 0 0 B
|
q0
1 0 1 0 B
|
q0

Halt

In the traversal, the input string does not reach the final state , so the string is rejected , it does not have '101' as a substring

- **Tuples of the designed Machine:**

Q= { q0, q1, q2, q3,q4}

E = {0, 1 }

T = {0, 1, B}

B = B

q0 = q0

F = { q4 }

5. Design a TM to replace occurrence of '111' by '101' in a binary string

B 1 1 1 1 0 B--------- i

B 1 0 1 1 0 B---------o

B 0 1 1 0 1 B ---------i

B 0 1 1 0 1 B -----------o

Transition Mapping Function :

	0	1	B
q0	(q0, 0, R)	(q1, 1, R)	(q5, B, N)
q1	(q0, 0, R)	(q2, 1, R)	(q5, B, N)
q2	(q0, 0, R)	(q3, 1, L)	--------
q3	--------	(q4, 0, R)	--------
q4	--------	(q0, 1, R)	--------
q5	Final State		

- **Tuples of the designed Machine:**

  Q= { q0, q1, q2, q3,q4, q5}

  E = {0, 1 }

  T = {0, 1, B}

  B = B

  q0 = q0

  F = { q5 }

Home Work :

- Design a TM to convert every occurrence of string ‘abb’ by ‘baa’

- Design a TM to accept language given by RE

$$0(0+1)^*11$$

6. Design a TM to recognize language $\{a^n b^n\}$

Transition Mapping Function :

	a	b	X	Y	B
q0	(q1, X, R)	--------	--------	(q3, Y, R)	--------
q1	(q1, a, R)	(q2, Y, L)	--------	(q1, Y, R)	--------
q2	(q2, a, L)	--------	(q0, X, R)	(q2, Y, L)	--------
q3	--------	--------	--------	(q3, Y, R)	(q4, B, N)
q4	Final State				

- Consider string

a a b b
|
q0
X a b b
|
q1
X a b b
|
q1
X a Y b
|
q2
X a Y b
|
q2
X a Y b
|
q2
X a Y b
|
q0

X X Y b
|
q1
X X Y b
|
q1
X X Y Y
|
q2
X X Y Y
|
q2
X X Y Y
|
q0

X X Y Y
|
q3
X X Y Y
|
q3
X X Y Y B
|
q3
X X Y Y B
|
q4

String Accepted

- Consider string

a a b
|
q0
X a b
|
q1
X a b
|
q1
X a Y
|
q2
X a Y
|
q2
X a Y
|
q2
X a Y
|
q0

X X Y
|
q1
X X Y B
|
q1
Halt

In the traversal, the string does not reach the final state hence ,the String is Rejected

- Consider string

a b b
|
q0
X b b
|
q1
X Y b
|
q2
X a Y
|
q0
X X Y
|
q1
X X Y
|
q1
X X Y B
|
q1

String Rejected

- **Tuples of the designed Machine:**

  Q= { q0, q1, q2, q3,q4}

  E = {a, b }

  T = {a, b, X, Y, B}

  B = B

  q0 = q0

  F = { q4 }

7. Design a TM to recognize language $\{a^n b^n c^n\}$

Transition Mapping Function :

	a	b	c	X	Y	Z	B
q0	(q1, X, R)	--------	--------	--------	(q4, Y, R)	--------	------
q1	(q1, a, R)	(q2, Y, R)	--------	--------	(q1, Y, R)	--------	-------
q2	--------	(q2, b, R)	(q3, Z, L)	--------	--------	(q2, Z, R)	------
q3	(q3, a, L)	(q3, b, L)	--------	(q0, X, R)	(q3, Y, L)	(q3, Z, L)	------
q4	--------	--------	--------	--------	(q4, Y, R)	(q4, Z, R)	(q5, B,
q5	Final State						

- Consider string

a a b b c c
|
q0
X a b b c c
|
q1
X a b b c c
|
q1
X a Y b c c
|
q2
X a Y b c c
|
q2
X a Y b c c
|
q2

X a Y b Z c
|
q3
X a Y b Z c
|
q3
X a Y b Z c
|
q3
X a Y b Z c
|
q3
X a Y b Z c
|
q0
X X Y b Z c
|
q1
X X Y b Z c
|
q1

X X Y Y Z c
|
q2
X X Y Y Z c
|
q2
X X Y Y Z Z
|
q3
X X Y Y Z Z
|
q0

X X Y Y Z Z B
|
q4
X X Y Y Z Z B
|
q5

String Accepted

- Consider string

a a b c
|
q0
X a b c
|
q1
X a b c
|
q1
X a Y c
|
q2
X a Y Z
|
q3
X a Y Z
|
q3
X a Y Z
|
q3

X a Y Z
|
q0
X X Y Z B
|
q1
X X Y Z B
|
q1

Halt

String Rejected

- Consider string

a b b
|
q0
X b b
|
q1
X Y b
|
q2
X a Y
|
q0
X X Y
|
q1
X X Y
|
q1
X X Y B
|
q1

String Rejected

- **Tuples of the designed Machine:**

  Q= { q0, q1, q2, q3,q4,q5}

  E = {a, b }

  T = {a, b, c, X, Y, Z, B}

  B = B

  q0 = q0

  F = { q5 }

Home Work :

- Design a TM to recognize language $\{a^n b^n a^n\}$

8. Design a TM to recognize even length binary palindrome strings

Transition Mapping Function :

	0	1	B
q0	(q1, B, R)	(q4, B, R)	(q6, B, R)
q1	(q1, 0, R)	(q1, 1, R)	(q2, B, L)
q2	(q3, B, L)	--------	--------
q3	(q3, 0, L)	(q3, 1, L)	(q0, B, R)
q4	(q4, 0, R)	(q4, 1, R)	(q5, B, L)
q5	--------	(q3, B, L)	--------
q6	Final State		

- Consider string

B 1 0 0 1 B
|
q0
B 0 0 1 B
|
q4
B 0 0 1 B
|
q4
B 0 0 1 B
|
q4
B 0 0 1 B
|
q4
B 0 0 1 B
|
q5

B 0 0 B B
|
q3
B 0 0 B B
|
q3
B 0 0 B B
|
q3

B 0 0 B B
|
q0
B B 0 B B
|
q1
B B 0 B B
|
q1
B B 0 B B
|
q1

B B 0 B B
|
q2
B B B B B
|
q3
B B B B B
|
q0
B B B B B
|
q6

String Accepted

- Consider string

B 0 0 1 B

|

q0

B B 0 1 B

|

q1

B B 0 1 B

|

q1

B B 0 1 B

|

q1

B B 0 1 B

|

q1

B B 0 1 B

|

q2

Halt

Sting Rejected

- **Tuples of the designed Machine:**

  Q= { q0, q1, q2, q3,q4,q5,q6}

  E = {0, 1 }

  T = {0, 1, B}

  B = B

  q0 = q0

  F = { q6}

9. Design a TM to recognize language L= { wcw^R | odd length binary palindrome strings }

Transition Mapping Function :

	0	1	c	B
q0	(q1, B, R)	(q4, B, R)	(q6, c , N)	--------
q1	(q1, 0, R)	(q1, 1, R)	(q1, c , R)	(q2, B, L)
q2	(q3, B, L)	--------	--------	--------
q3	(q3, 0, L)	(q3, 1, L)	(q3, c, L)	(q0, B, R)
q4	(q4, 0, R)	(q4, 1, R)	(q4,c, R)	(q5, B, L)
q5	--------	(q3, B, L)	--------	--------
q6	Final State			

- Consider string

B 1 0 c 0 1 B
|
q0
B B 0 c 1 B
|
q4
B B 0 c 1 B
|
q5
B B 0 c 0 B B
|
q3
B B B c 0 B B
|
q1
B B B c 0 B B
|
q2

B B B c B B B
|
q3
B B B c B B B
|
q0
B B B c B B B
|
q6

String Accepted

- Consider string

B 1 0 c 1 B

|

q0

B B 0 c B B

|

q5

B B 0 c B B

|

q3

B B 0 c B B

|

q0

B B B c B B

|

q1

B B B c B B

|

q2

Halt

String Rejected

- **Tuples of the designed Machine:**

  Q= { q0, q1, q2, q3,q4,q5,q6}

  E = {0, 1 }

  T = {0, 1, c, B}

  B = B

  q0 = q0

  F = { q6}

10. Design a TM to check for well formedness of parentheses

(( ))

( ) ( )

(( ) ( ))

Transition Mapping Function :

	(	)	X	Y	B
q0	(q0, (, R)	(q1, X, L)	(q0, X, R)	(q0, Y, R)	(q2, B, L)
q1	(q0, Y, R)	--------	(q1, X, L)	(q1, Y, L)	--------
q2	--------	--------	(q2, X, L)	(q2, Y, L)	(q3, B, N)
q3	Final State				

- Consider string

B (( )) B
|
q0
B (( )) B
|
q0
B ((X) B
|
q1
B (Y X) B
|
q0
B Y Y X X B
|
q2
B Y Y X X B
|
q3
String Accepted

B (( ) B
|
q3
B ((X B
|
q0
B (Y X B
|
q2
B (Y X B
|
q2

Halt

String Rejected

11. Design a TM to recognize the strings having equal number of a’s and b’s

a a b b

b b a a

a b a b

b a b a

Transition Mapping Function :

	a	b	X	Y	B
q0	(q1, X, R)	(q3, Y, R)	(q0, X, R)	(q0, Y, R)	(q5, B, L)
q1	(q1, a, R)	(q2, Y, L)	--------	(q1, Y, R)	--------
q2	(q2, a, L)	--------	(q0, X, R)	(q2, Y, L)	--------
q3	(q4, X, L)	(q3, b, R)	(q3, X, R)	--------	--------
q4	--------	(q4, b, L)	(q4, X, L)	(q0, Y, R)	--------
q5	Final State				

- Consider string

a a b b
|
q0
X a b b
|
q1
X a b b
|
q1
X a Y b
|
q2
X a Y b
|
q2
X a Y b
|
q2
X a Y b
|
q0

X X Y b
|
q1
X X Y b
|
q1
X X Y Y
|
q2
X X Y Y
|
q2
X X Y Y
|
q0

X X Y Y
|
q3
X X Y Y
|
q3
X X Y Y B
|
q3
X X Y Y B
|
q4

String Accepted

- Consider string

b b a a
|
qo
Y b a a
|
q3
Y b a a
|
q3
Y b X a
|
q4
Y b X a
|
q4
Y b X a
|
qo
Y b X a
|
qo

Y Y X a
|
q3
Y Y X a
|
q3
Y Y X X
|
q4
Y Y X X
|
q4
Y Y X X
|
qo

Y Y X X
|
qo
X X Y Y B
|
q5

String Accepted

- Consider string

a b a b
|
q0
X b a b
|
q1
X Y a b
|
q0
X Y a b
|
q0
X Y X b
|
q1
X Y X Y
|
q2
X Y X Y
|
q0

X X Y Y B
|
q0
X X Y Y B
|
q5

String Accepted

- Consider string

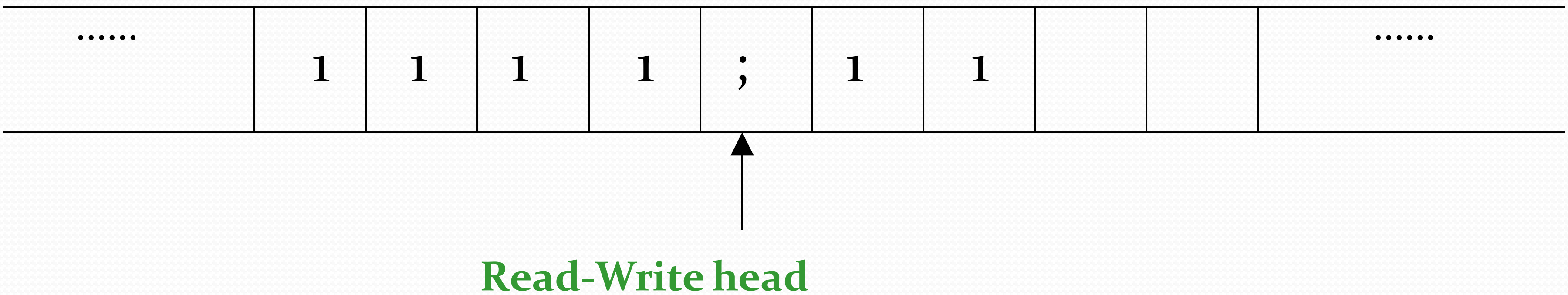
b a b a
|
qo
Y a b a
|
q3
Y X b a
|
q4
Y X b a
|
qo
Y X b a
|
qo
Y X Y a
|
q3
Y X Y X
|
q4

Y X Y X
|
qo
X X Y Y B
|
q5

String Accepted

12. Let x and y be two unary numbers. Construct a TM which will halt in the final state if x > y

The Tape :



- Consider string

B 1 1 ; 1 B
|
q0
B * 1 ; 1 B
|
q1
B * 1 ; 1 B
|
q1
B * 1 ; 1 B
|
q2
B * 1 ; * B
|
q3
B * 1 ; * B
|
q3

B * 1 ; * B
|
q3
B * 1 ; * B
|
q3

B * 1 ; * B
|
q0
B * 1 ; * B
|
q0
B * * ; * B
|
q1
B * * ; * B
|
q2
B * * ; * B
|
q2

B * * ; * B
|
q2
B * * ; * B
|
q4

String Accepted

- Consider string

B 1 ; 1 1 B
|
q0
B * ; 1 1 B
|
q1
B * ; 1 1 B
|
q2
B * ; * 1 B
|
q3
B * ; * 1 B
|
q3
B * ; * 1 B
|
q3

B * ; * 1 B
|
q3

B * ; * 1 B
|
q0
B * ; * B
|
q0
Halt

String Rejected

13 . Design a TM to reverse input string $\sum = \{a,b\}$

example:

B a a b B.
↑ ↑ ↑ ↑
q_0 . q_0 q_0 q_0.

B a a b B
↑q_1

B a a * B.
↑q_2

B a a * b B
↑q_3

B a a * b B
↑q_1

B a * * b B
↑q_4

B a * * b B
↑q_4

B a * * b B
↑q_4

B a * * b a
↑q_5

B a * * b a B
↑q_5

B a * * b a B
↑q_5

B a * * b a B
↑q_5

B a * * b a B
↑q_1

B a * * b a B
↑q_1

B * * * b a B
↑q_4 .... ↑q_4

B * * * b a B

B * * * b a a
↑q_5 .. ↑q_5

B * * * b a a
↑q_1

B. * * * b a a B.
↑q_6 (Final state)

I/p string ≡ a a b
o/p ≡ b a a.

Transition Mapping Function :

	a	b	*	B
q0	(q0, a, R)	(q0, b, R)	--------	(q1, B, L)
q1	(q4,*, R)	(q2, *, R)	(q1, *, L)	(q5, B, N)
q2	(q2, a, R)	(q2, b, R)	(q2, *, R)	(q3, b, L)
q3	(q3, a, L)	(q3, b, L)	(q1, *, L)	--------
q4	(q4, a, R)	(q4, b, R)	(q4, *, R)	(q3, a, L)
q5	Final State			

gn a TM which compares two unary
umbers a & b and if

i) $a < b$, m/c writes 'l' at the end

ii) $a = b$, m/c writes 'e' at the end,

iii) $a > b$, m/c writes 'g' at end.

* (Preferably give example of all 3 cases)

i) B 1 ; 1 B

B X ; 1 B
$\uparrow q_1$

B X ; 1 B
$\uparrow q_2$

B X ; X B
$\uparrow q_3$

B X ; X B
$\uparrow q_4$

B X ; X B
$\uparrow q_0$

B X ; X B
$\uparrow q_5$

B X ; X B
$\uparrow q_5$

B X ; X e
$\uparrow q_7$
(final state)

ii) B 1 1 ; 1 B

B X 1 ; 1 B

↑ q_1

B X 1 ; 1 B

↑ q_2

B X 1 ; X B

↑ q_3

B X 1 ; X B

↑ q_4

B X 1 ; X B

↑ q_0

B X X ; X B

↑ q_2

B X X ; X 'g'

↑ q_7.

(final state).

iii) B 1 ; 1 1 B

B X ; 1 1 B

↑ q_2

B X ; X 1 B

↑ q_3

B X ; X 1 B

↑ q_0

B X ; X 1 B

↑ q_5

B X ; X 1 B

↑ q_6

B X ; X 1 'λ'

↑ q_7.

(final stat

	1	;	X	B
q_0	(q_1, X, R)	$(q_5, ;, R)$	–	–
q_1	$(q_1, 1, R)$	$(q_2, ;, R)$	–	–
q_2	(q_3, X, L)	–	(q_2, X, R)	(q_7, g, N)
q_3	–	$(q_4, ;, L)$	(q_3, X, L)	–
q_4	$(q_4, 1, L)$	–	(q_0, X, R)	–
q_5	$(q_6, 1, R)$	–	(q_5, X, R)	(q_7, e, N)
q_6	$(q_6, 1, R)$	–	–	(q_7, l, N)

Final state:

Q.6 Design a TM to compute n mod 2, where, n is a unary no.

Sol^n:-

In unary i/p, odd no. of 1's will o/p a '1' on the tape (to indicate remainder is 1) & even no. of 1's should produce '0' on the tape (to indicate remain[der] if n mod 2 = 0).

Logic: • Keep converting at a time two ones to Blank.

• At end if one '1' remains before Blank, remainder is '1' or it is '0'.

	1	B
q_0	$(q_1, 1, R)$	$(q_4, 0, N)$
q_1	(q_2, B, L)	$(q_4, 1, N)$
q_2	(q_3, B, R)	–
q_3	–	(q_0, B, R)
q_4	final state	

B 1 1 1 B.
↑
q_0

B 1 1 1 B
↑
q_1

B 1 B 1 B
↑
q_2

B B B 1 B
↑
q_3

B B B 1 B
↑
q_0

B B B 1 B
↑
q_1

B B B 1 '1'
↑
q_4.
(final state).

$\therefore$ 111 mod 2 = 1.
(We focus on the symbol with final state)

B 1 1 B.
↑
q_0.

B 1 1 B
↑
q_1

B 1 B B
↑
q_2

B B B B
↑
q_3

B B B B
↑
q_0

B B B 0
↑
q_4
final state

$\therefore$ 11 mod 2 = 0

TM with Transition Diagram :