D Y PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Theoretical Computer Science

Unit 5: Turing Machine

Faculty Name : Ms. Namita Pulgam

Index

Lecture No 45:

Turing Machine Examples (Part-2)

Example 3:

Q. Design Turing Machine for odd length palindrome over Σ= {0,1}.

Language L= {010, 101, 01010,…}

Logic :

-Take the first character (either 0 or 1), mark it as '*' and move right till the blank symbol

-After blank move left. If the last character matches with the first character then mark it as blank symbol

-In the same way, repeat above cycle to match second symbol with second last symbol and so on.

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

$\Gamma = \{ 0, 1, *, B \}$

Initial state : q_0 **Final state :** q_f

Logic in Detail

q0- Take the first character (either 0 or 1) mark it as ‘*’ and move right

q1- Search for blank symbol, keep as it is and then move left. While doing this skip all 0’s and 1’s

q2- Check whether the symbol from left is similar to symbol from right . If yes then mark it as ‘B’ **(If read 0 in state q0)**

q3- Move left to search for ‘*’. While doing this skip all 0’s and 1’s.

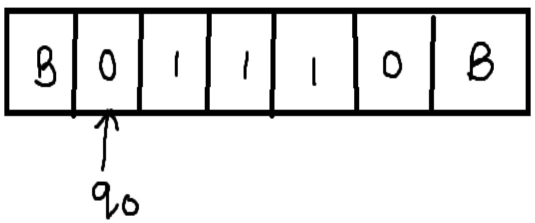
q4- Search for blank symbol, keep as it is and then move left. While doing this skip all 0’s and 1’s

q5- Check whether the symbol from left is similar to symbol from right. If yes then mark it as ‘B’. **(If read 1 in state q0)**

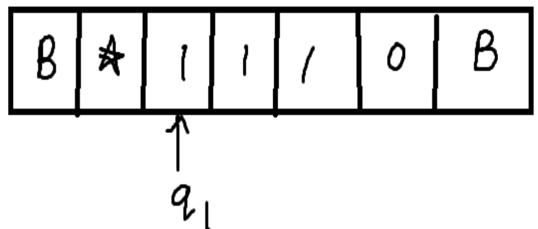
q6-Move left to search for ‘*’. While doing this skip all 0’s and 1’s.

qf- On q2 and q5 states after moving left if we get ‘*’ that means all symbols are over. then reach to final state i.e. qf.

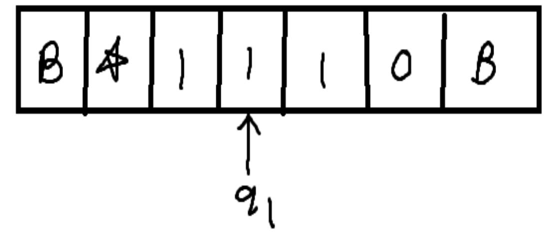
Consider input string– 01110



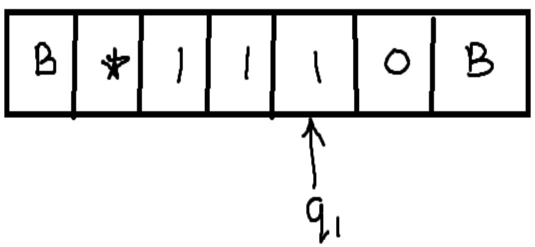
δ(q0, 0)-> (q1,*,R)



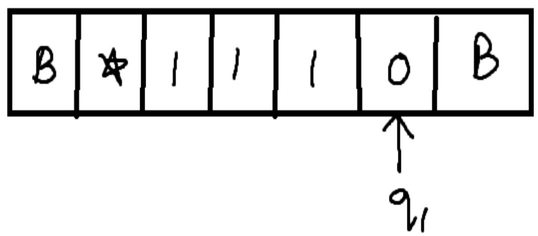
δ(q1, 1)-> (q1,1,R)



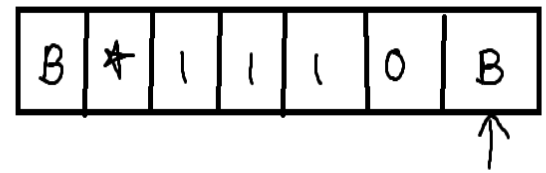
δ(q1, 1)-> (q1,1,R)



δ(q1, 1)-> (q1,1,R)

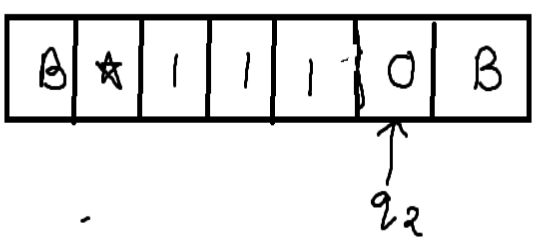


δ(q1, 0)-> (q1,0,R)

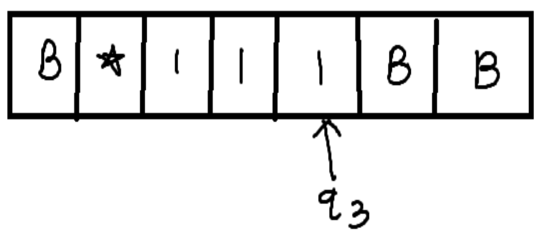


δ(q1, B)-> (q2,B,L)

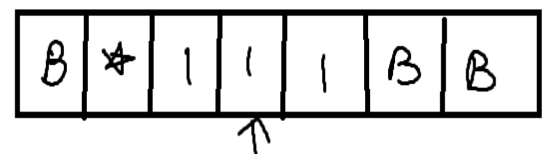
q1



δ(q2, 0)-> (q3,B,L)



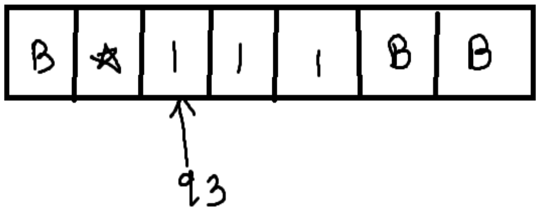
δ(q3, 1)-> (q3,1,L)



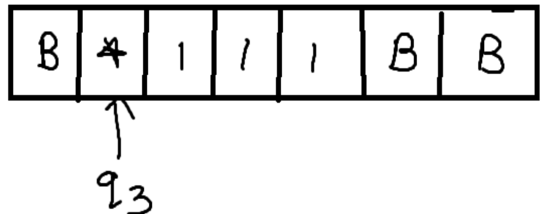
δ(q3, 1)-> (q3,1,L)

q3

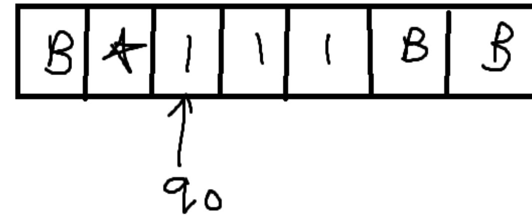
D Y PATIL DEEMED TO BE UNIVERSITY RAMRAO ADIK INSTITUTE OF TECHNOLOGY NAVI MUMBAI



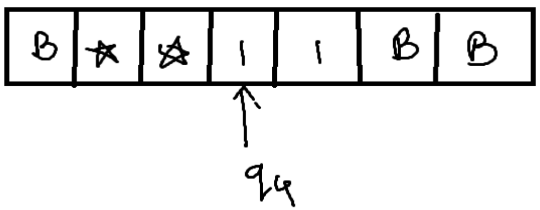
δ(q3, 1)-> (q3,1,L)



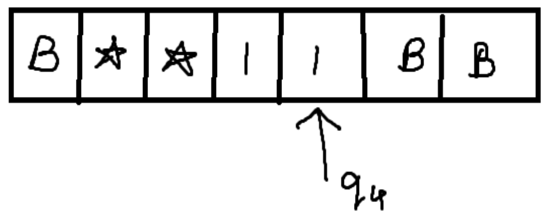
δ(q3, *)-> (q0,*,R)



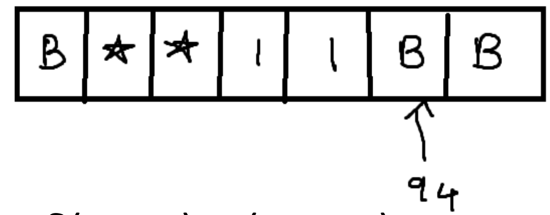
δ(q0, 1)-> (q4,*,R)



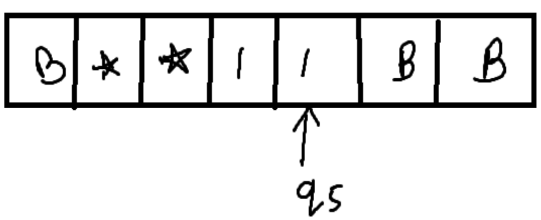
δ(q4, 1)-> (q4,1,R)



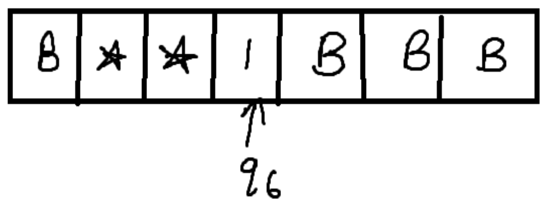
δ(q4, 1)-> (q4,1,R)



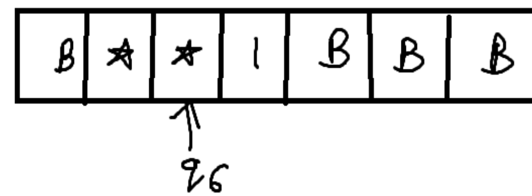
δ(q4, B)-> (q5,B,L)



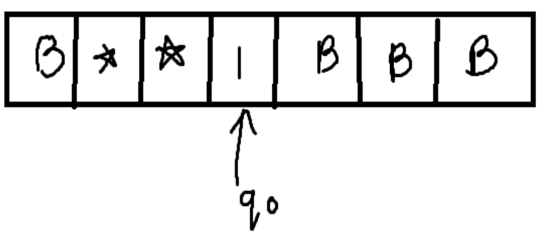
δ(q5, 1)-> (q6,B,L)



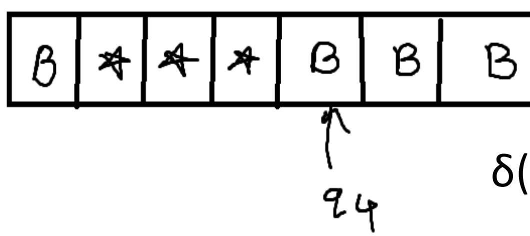
δ(q6, 1)-> (q6,1,L)



δ(q6, *)-> (q0,*,R)

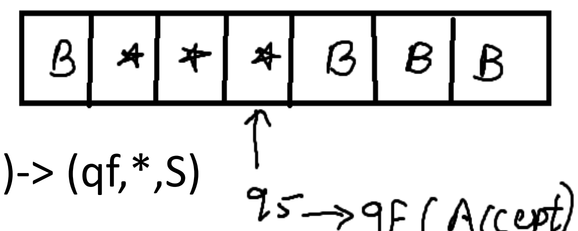


δ(q0, 1)-> (q4,*,R)



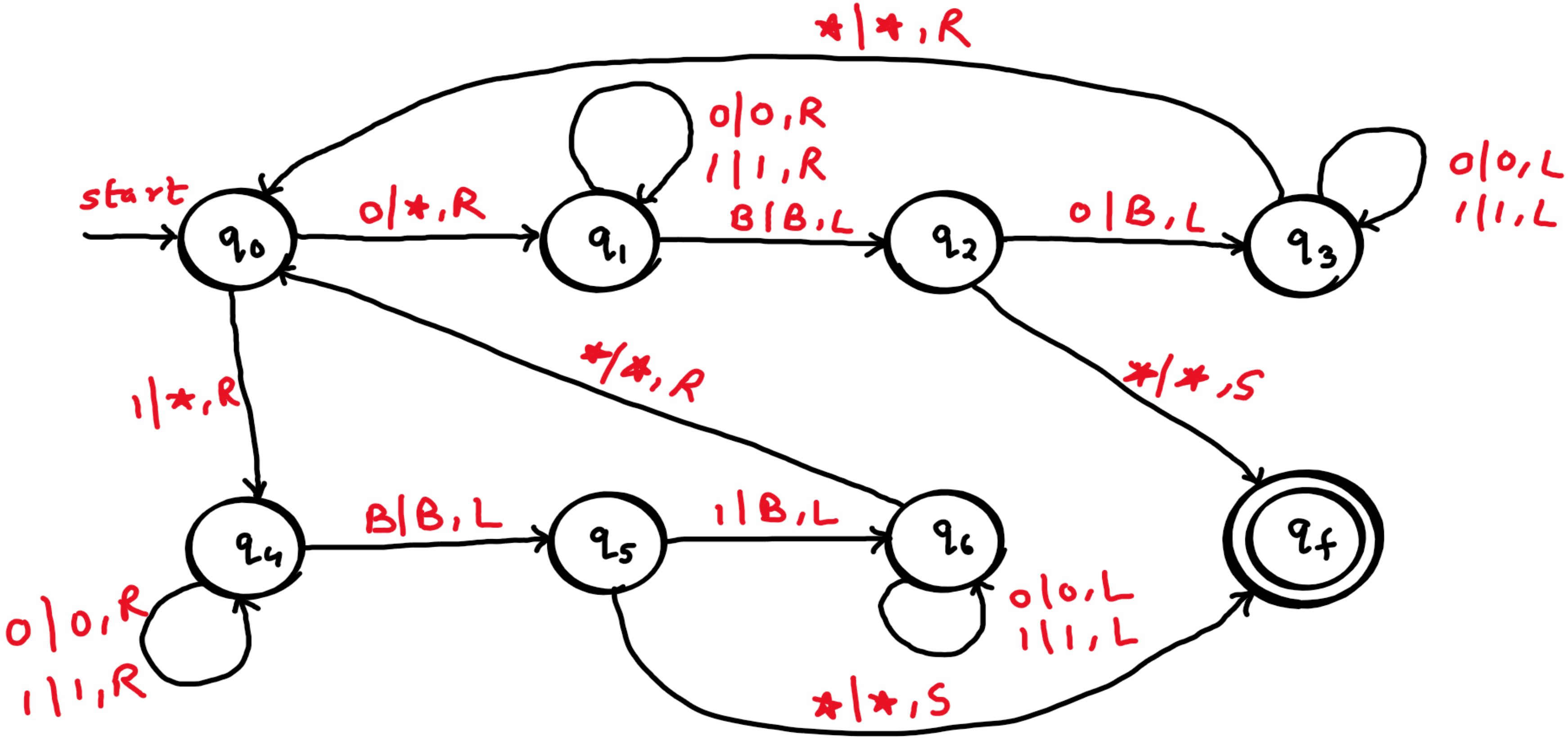
δ(q4, B)-> (q5,B,L)

δ(q5, *)-> (qf,*,S)



Q \ Γ	0	1	*	B
q0	(q1, *, R)	(q4, *, R)	-	-
q1	(q1, 0, R)	(q1, 1, R)	-	(q2, B, L)
q2	(q3, B, L)	-	(qf, *, S)	-
q3	(q3, 0, L)	(q3, 1, L)	(q0, *, R)	-
q4	(q4, 0, R)	(q4, 1, R)	-	(q5, B, L)
q5	-	(q6, B, L)	(qf, *, S)	-
q6	(q6, 0, L)	(q6, 1, L)	(q0, *, R)	-
qf*	Final State			

Transition Diagram



Example 4:

Q. Design Turing Machine for even length palindrome over Σ= {a,b}.

Language L= {abba, baab,…}

Logic :

-Take the first character (either a or b), mark it as '*' and move right till the blank symbol

-After blank move left. If the last character matches with the first character then mark it as blank symbol

-In the same way, repeat above cycle to match second symbol with second last symbol and so on.

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

$\Gamma = \{ a, b, *, B \}$

Initial state : q_0

Final state : q_f

Logic in Detail

q0- Take the first character (either a or b) mark it as '*' and move right

q1- Search for blank symbol, keep as it is and then move left. While doing this skip all a's and b's

q2- Check whether the symbol from left is similar to symbol from right . If yes then mark it as 'B' **(If read a in state q0)**

q3- Move left to search for '*'. While doing this skip all a's and b's.

q4- Search for blank symbol, keep as it is and then move left. While doing this skip all a's and b's

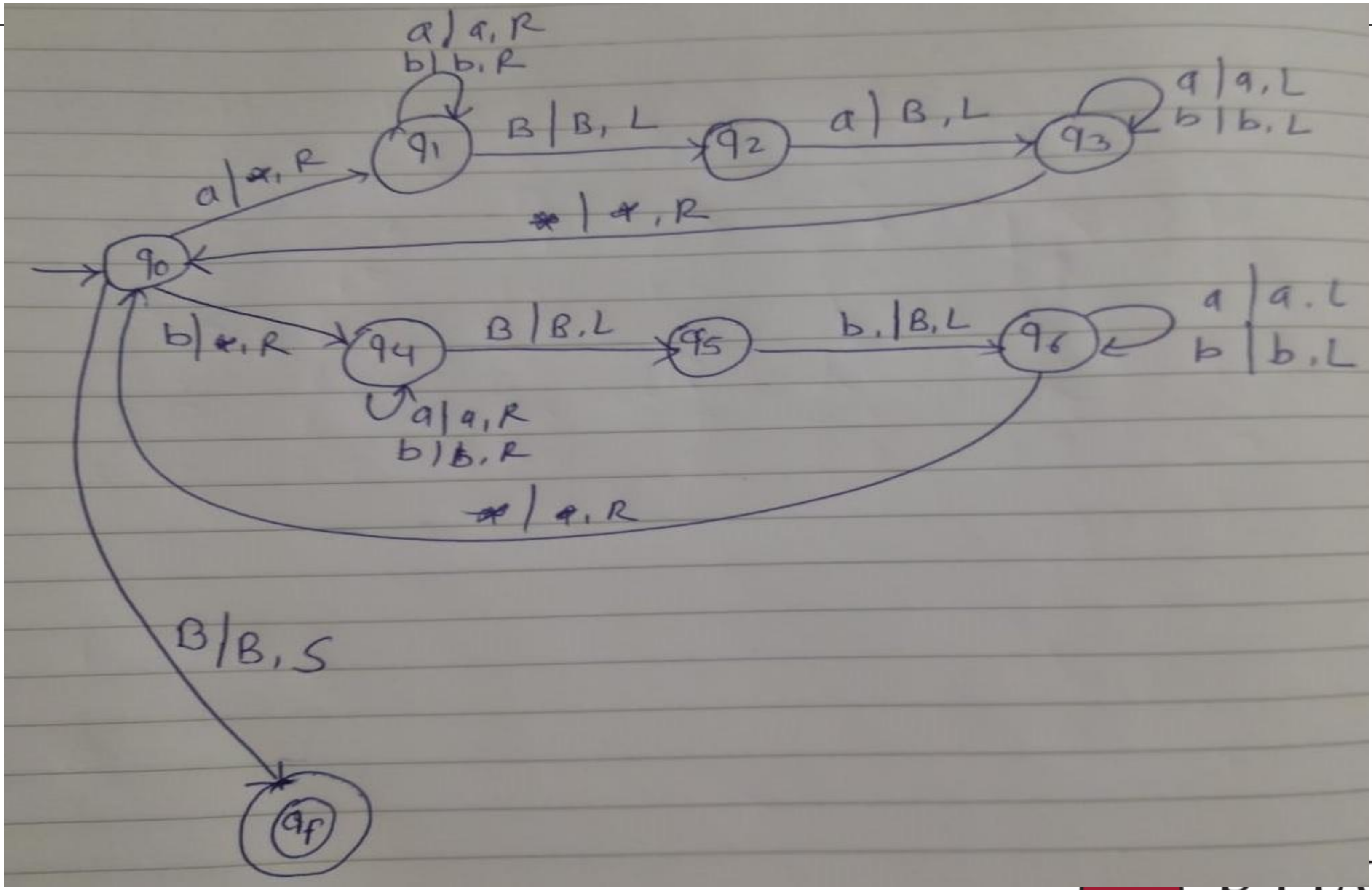
q5- Check whether the symbol from left is similar to symbol from right. If yes then mark it as 'B'. **(If read b in state q0)**

q6-Move left to search for '*'. While doing this skip all a's and b's.

qf- On q0 state after moving right if we get 'B' that means all symbols are over. then reach to final state i.e. qf.

Q \ Γ	a	b	*	B
q0	(q1, *, R)	(q4, *, R)	-	(qf, B, S)
q1	(q1, a, R)	(q1, b, R)	-	(q2, B, L)
q2	(q3, B, L)	-	-	-
q3	(q3, a, L)	(q3, b, L)	(q0, *, R)	-
q4	(q4, a, R)	(q4, b, R)	-	(q5, B, L)
q5	-	(q6, B, L)	-	-
q6	(q6, a, L)	(q6, b, L)	(q0, *, R)	-
qf*	Final State			

Transition Diagram



D Y PATIL
DEEMED TO BE
UNIVERSITY
RAMRAO ADIK
INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Thank You

Lecture No 46:

Turing Machine Examples (Part-3)

Turing Machine as Function Generator

- Turing machine can compute some functions.
- Its plays a role of function generator.
- TM can perform various arithmetic operations such as addition, subtraction, multiplication and so on.

Example 5:

Q. Design Turing Machine to perform addition of two unary numbers.

- In unary number system,

  Value of any number(unary) = Number of 0's

  where, 0 is used to represent the unary number.

- The required number will be represented in following format :

  - $(2)_{10} = 00$

  - $(3)_{10} = 000$

- Let us assume we want to add two numbers 2 and 3 i.e. (2+3)
- In order to separate the numbers on the tape we will use symbol "1" as a separator.
- Hence, the contents of input tape for numbers 2 and 3 will be :

B	0	0	1	0	0	0	B

- After addition of 2 and 3, the number of 0's on tape must be 5.
- Thus the output on tape will be as shown below:

B	0	0	0	0	0	B

D Y PATIL DEEMED TO BE UNIVERSITY
RAMRAO ADIK INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Example 5: (cont..)

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

Logic :

q0- Bypass all 0's of first number and move right to search '1'

q1- When we get '1' make it '0' and move right till blank symbol.

q2-After second step there will be one extra '0' on tape. Therefore, when we get blank symbol move left and make the last '0' as blank symbol.

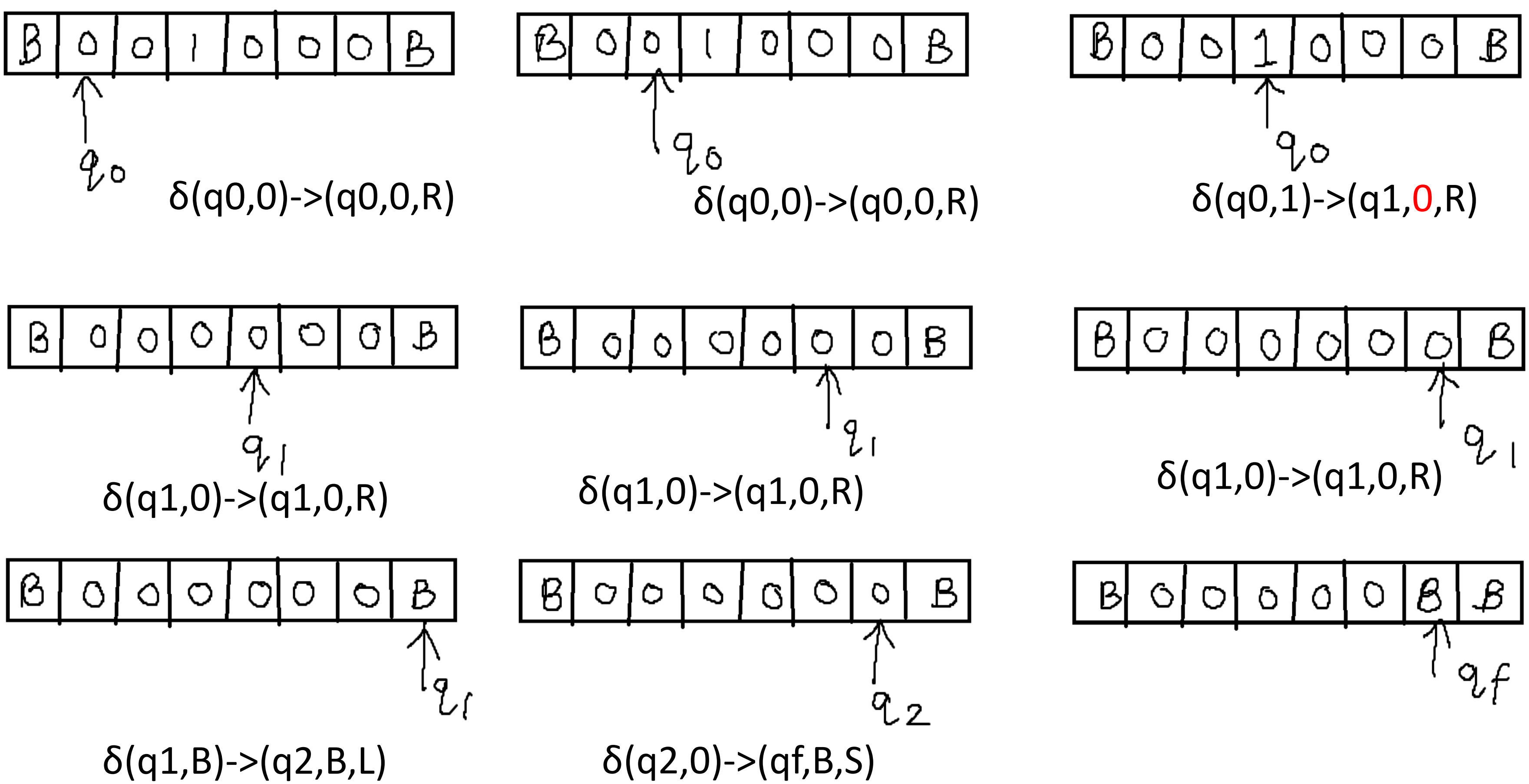
qf- Final state

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

Initial state : q_0

Final state : q_f

Example Processing : Addition of 2 and 3

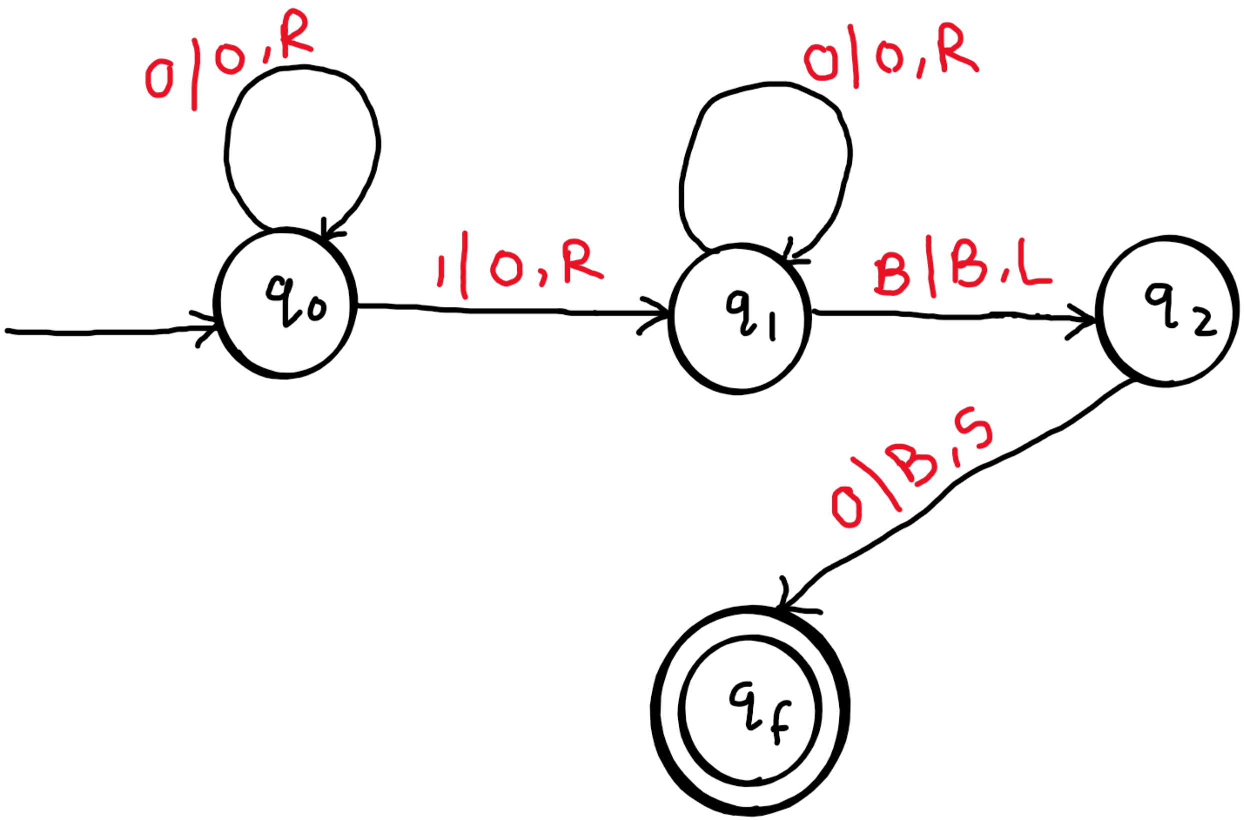


D Y PATIL DEEMED TO BE UNIVERSITY RAMRAO ADIK INSTITUTE OF TECHNOLOGY NAVI MUMBAI

Transition Table

Q \ Γ	0	1	B
→ q0	(q0, 0, R)	(q1, 0, R)	
q1	(q1, 0, R)		(q2, B, L)
q2	(qf, B, S)		
qf*	Final State		

Transition Diagram



Thank You

Lecture No 47:

Turing Machine Variants, Universal TM

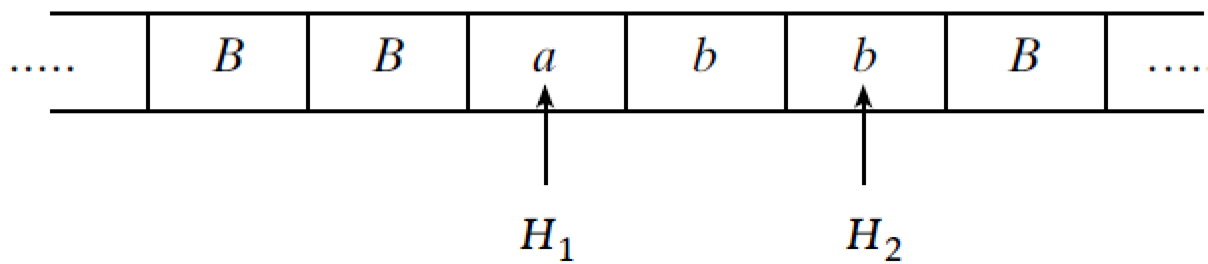
Variants of Turing Machine

- The Turing machine we have discussed so far is called as standard Turing machine
- In order to enhance the power of standard Turing machine many variations of it are suggested.
- Some of the variations are listed below:
  - Multi-Head Turing machine
  - Multi-Tape Turing machine
  - Non-deterministic Turing machine

Variants of Turing Machine (continue...)

Multi-head Turing Machine:

- A multi-head Turing machine contain two or more heads to read the symbols on the same tape.
- In one step all the heads sense the scanned symbols and move or write independently.

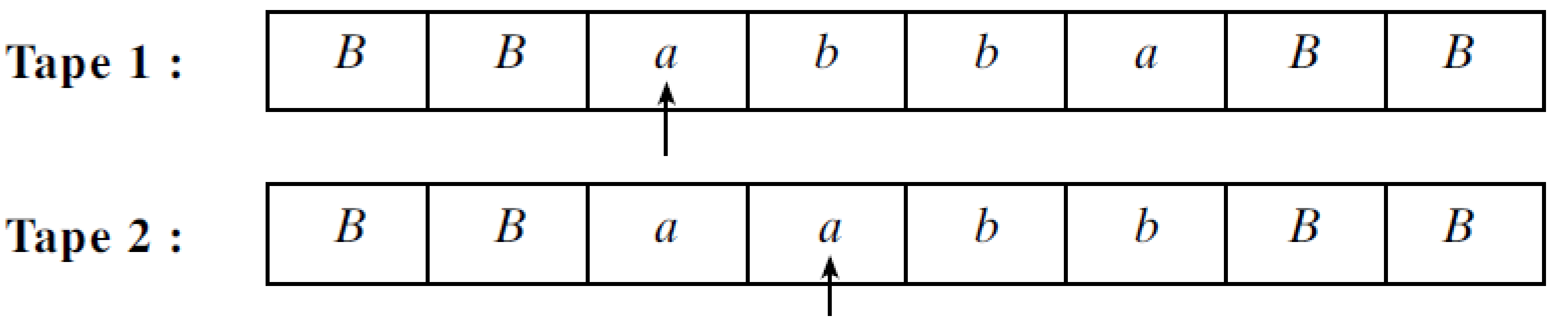


δ(the state, symbol under head H_1, symbol under head H_2) = (New State, (S_1, M_1), (S_2, M_2))

Variants of Turing Machine (continue...)

Multi-tape Turing Machine:

- This type of Turing machine consists of multiple tapes each having an independent head. Each head is capable to perform read/ write operations and the movement includes left or right or no movement.



$$\delta(\text{the state, symbol under head of Tape 1, symbol under head of Tape 2}) = (\text{New State}, (S_1, M_1), (S_2, M_2))$$

Variants of Turing Machine (continue...)

Non-deterministic Turing Machine:

- In case of standard Turing machine there exists only one possible transition from the current state for the current input. The standard Turing machine is also referred as Deterministic Turing machine.
- But, for non-deterministic Turing machine, it is possible that there are multiple transitions from the current state for the current input
- **Formal definition:** **$M = \{Q, \Sigma, \Gamma, \delta, q_0, F, B\}$**

  where,

  Q : Finite set of states

  Σ : Finite set of input symbols

  Γ: Finite set of tape symbols which include the blank symbol

  δ: The transition function, $Q X \Sigma \rightarrow 2^Q$ (Power set of Q)

  q_0 : Initial state

  F: Finite set of final states

  B: Blank symbol

D Y PATIL DEEMED TO BE UNIVERSITY RAMRAO ADIK INSTITUTE OF TECHNOLOGY NAVI MUMBAI

Universal Turing Machine

- A limitation of standard Turing Machines is that they are "hardwired"

  they execute only one program

– Real Computers are re-programmable

Solution: Universal Turing Machine

- We can construct a single Turing machine which can solve all sorts of problems.
- This type of Turing machine is called as Universal Turing Machine (UTM). Thus, **Universal Turing Machine is a Turing Machine which simulates any other Turing Machine for a given input.**

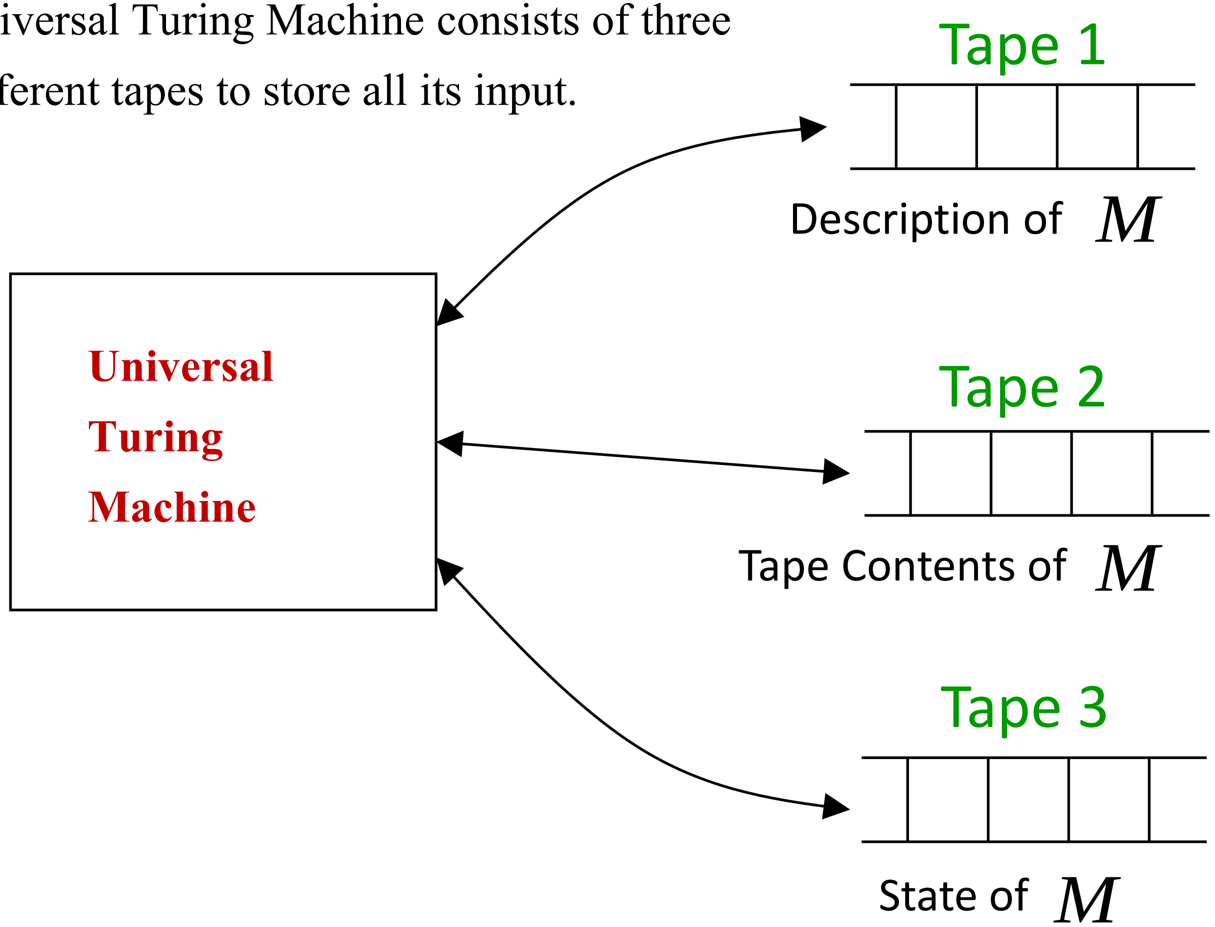
- The input of this Universal Turing Machine consists of:

  → Description of transitions of other Turing machine M

  → Input string of other Turing machine M

Three tapes

Universal Turing Machine consists of three different tapes to store all its input.



D Y PATIL DEEMED TO BE UNIVERSITY RAMRAO ADIK INSTITUTE OF TECHNOLOGY NAVI MUMBAI

Alphabet Encoding

Symbols: a b c d $\cdots$

Encoding: 1 11 111 1111

State Encoding

States: q_1 q_2 q_3 q_4 $\cdots$

Encoding: 1 11 111 1111

Head Move Encoding

Move: L R

Encoding: 1 11

Transition Encoding

Transition: $\delta(q_1, a) = (q_2, b, L)$

Encoding: $1\,0\,1\,0\,11\,0\,11\,0\,1$

separator

Alphabet Encoding (continue...)

Turing Machine Encoding

Transitions:

$\delta(q_1, a) = (q_2, b, L)$ $\delta(q_2, b) = (q_3, c, R)$

Encoding:

1 0 1 0 11 0 11 0 1 00 11 0 11 0 111 0 111 0 11

separator

Tape 1 contents of Universal Turing Machine:

binary encoding of the simulated machine M

Tape 1

1 0 1 0 11 0 11 0 10011 0 1 10 111 0 111 0 1100…

Working

It reads current input and current state from the tape 2 and 3.

Then checks description stored on tape1 and do the transitions according to it.

Thank You