COMPUTABLE FUNCTIONS

Dr. A. Beulah

AP/CSE

LEARNING OBJECTIVE

- To Design Turing machines for any Languages (K3)
  - To Understand the concept of Turing Machine

TURING-COMPUTABLE FUNCTIONS

- A total function f: $\Sigma^* \rightarrow \Sigma^*$ is Turing-computable if there exists a DTM M such that for every x in Σ^*,

  $$(s, BxB) \vdash^* (h, Bf(x)B).$$

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

$L = \{1^n 2^n 3^n\}$

A, R

- A partial f: $\Omega \rightarrow \Sigma^*$ is Turing-computable if there exists a DTM M such that $L(M)=\Omega$ and for every x in Ω,

  $$(s, BxB) \vdash^* (h, Bf(x)B).$$

TURING-COMPUTABLE FUNCTIONS

Construct a TM for successive function ?

$$f : N \rightarrow N,\ f(x) = x+1.$$

Assume that the input is encoded in UNARY form.

Let M = (Q, Σ, Γ, δ, q0, B, F)

Q = {q0,q1} q0=start state q1=final state

Σ = {0}

Γ = {0,B}

F = {q1}

$x \quad f(x)$

$N \quad x+1$

$5 \rightarrow 6$

$2 \rightarrow 3$

$4 \rightarrow 5$

0, 1, 2, 3 . . . 9

1	0
2	00
3	000
4	0000

$\delta(q_0, 0) = (q_0, 0, R)$

$\delta(q_0, B) = (q_1, 0, R)$

$2 \quad 2+1$

$00 \quad 000$

TURING-COMPUTABLE FUNCTIONS

Construct a TM for successive function ?

$$f : N \rightarrow N,\ f(x) = x+1.$$

Assume that the input is encoded in UNARY form.

Let M = (Q, Σ, Γ, δ, q0, B, F)

Q = {q0,q1} q0=start state q1=final state

Σ = {0}

Γ = {0,B}

F = {q1}

00 2 , 2+1 000

$q_0 00 \vdash 0 q_0 0$

$\vdash 00 q_0 B$

$\vdash 000 q_1 B$

States	Tape Symbols	
	0	B
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$
q_1	—	—

TURING-COMPUTABLE FUNCTIONS

Let us consider the input x=3, This is encoded as 000.

$$(q_0, \underline{0}00B) \vdash (q_0, 0\underline{0}0B) \vdash (q_0, 00\underline{0}B)$$

$$\vdash (q_0, 000\underline{B}) \vdash (q_1, 000\underline{0}B)$$

The machine halts in an accepting state q_1

by computing the successive of x

ADDITION

- We can represent numbers on a TM by using the unary representation, which just uses n 0's to represent the number n.

  Example: 00000 = 5

- We can construct a TM to concatenate two strings together.
- If we represent two numbers in unary form, then a TM which concatenates these two strings is actually performing addition!

→ m, n

& m+n

$f(m,n) = m+n$

m n

3+2 000100

000100

00000

3+0 B

2+2

00100 → 0000

ADDITION

- Addition of two numbers as simple as concatenation of two strings.
- Assume two integers are represented in unary form on the tape of a Turing machine.
- Assume the two integers are separated by a 1 between them.
- Design a Turing machine that will add these two numbers.

ADDITION

- f(m,n)=m+n

in '0' → B

$\delta(q_0, 0) = (q_1, B, R)$

$\delta(q_1, 0) = (q_1, 0, R)$

$\delta(q_1, 1) = (q_2, 0, R)$

$\delta(q_2, 0) = (q_2, 0, R)$

$\delta(q_2, B) = (q_3, B, R)$

000100

00000B

000100

B00000B

0010

2+1

000

$q_0 0010$

$\vdash B q_1 010$

$\vdash B 0 q_1 10$

$\vdash B 0 0 q_2 0$

$\vdash B 0 0 0 q_2 B$

$\vdash B 0 0 0 B q_3$

$q_3 \in F$

ADDITION

- 3+2

SUBTRACTION M - N

- For example, proper subtraction m – n is defined to be

f(m,n) =

m – n for m >= n, and '00.'

zero for m < n. 'B' '0' B

- The TM M = ({q0,q1,...,q6}, {0,1}, {0,1,B}, ∂, q0, B, {})

3-1

0001 0

B

00

The function ∂ is described below.

∂(q0,0) = (q1,B,R) Begin. Replace the leading 0 by B.

∂(q1,0) = (q1,0,R) Search right looking for the first 1.
∂(q1,1) = (q2,1,R)

∂(q2,1) = (q2,1,R) Search right past 1's until encountering a 0. Change that 0 to 1.
∂(q2,0) = (q3,1,L)

∂(q3,0) = (q3,0,L) Move left to a blank. Enter state q0 to repeat the cycle.
∂(q3,1) = (q3,1,L)
∂(q3,B) = (q0,B,R)

If in state q2 a B is encountered before a 0, we have situation i described above. Enter state q4 and move left, changing all 1's to B's until encountering a B. This B is changed back to a 0, state q6 is entered and M halts.

∂(q2,B) = (q4,B,L)
∂(q4,1) = (q4,B,L)
∂(q4,0) = (q4,0,L)
∂(q4,B) = (q6,0,R)

If in state q0 a 1 is encountered instead of a 0, the first block of 0's has been exhausted, as in situation (ii) above. M enters state q5 to erase the rest of the tape, then enters q6 and halts.

∂(q0,1) = (q5,B,R)
∂(q5,0) = (q5,B,R)
∂(q5,1) = (q5,B,R)
∂(q5,B) = (q6,B,R)

0010 2-1 2-1

0010

q_0 0010

BB11 B

m<n 2-3 3-1

00010

BB11

00BB

BB110

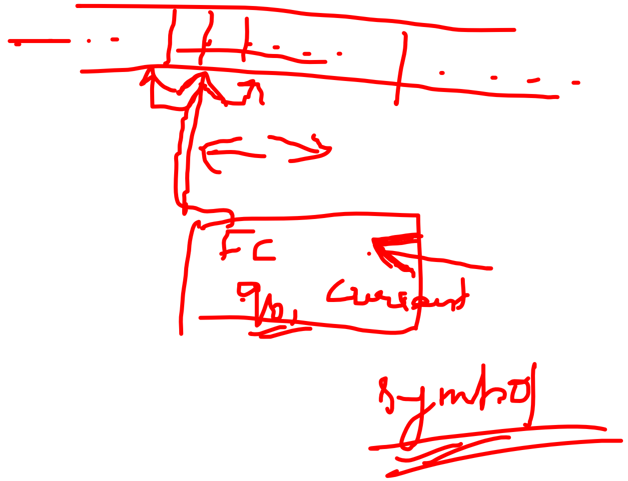
BBBB

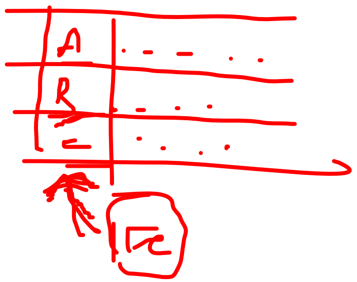
SUBTRACTION M - N

	symbol		
state	0	1	B
q_0	(q_1, B, R)	(q_5, B, R)	-
q_1	$(q_1, 0, R)$	$(q_2, 1, R)$	-
q_2	$(q_3, 1, L)$	$(q_2, 1, R)$	(q_4, B, L)
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	(q_0, B, R)
q_4	$(q_4, 0, L)$	(q_4, B, L)	$(q_6, 0, R)$
q_5	(q_5, B, R)	(q_5, B, R)	(q_6, B, R)
q_6	-	-	-

PROGRAMMING TECHNIQUES OF TURING MACHINES

- **Storage in the Finite Control**
- **Multiple Tracks**
- **Checking off Symbols**
- **Subroutines**





STORAGE IN THE FINITE CONTROL

- The finite control can be used to hold the finite amount of information.
- It is considered as a pair of elements, like (q_0,a), *where one exercising control* and second component stores a symbol in the finite control.
- Consider a turing machine M which accepts the language 01* + 10*
- Let M = (Q, {0,1}, {0,1,B}, δ, {q_0,B},B,F)
- Q = {q_0,q_1} x {0,1,B}

  = ([q_0,0], [q_0,1], [q_0,B], [q_1,0], [q_1,1], [q_1,B])
- F = {[q_1,B]}

[q_0, Symbol]

q_0

0.11111

1000000

L={0,1, 01, 011, 0111....
10, 100, 1000....}

STORAGE IN THE FINITE CONTROL

$$\delta([q_0,B],0) = ([q_1,0],\ 0,R)$$

$$\delta([q_0,B],1) = ([q_1,1],\ 1,\ R)$$

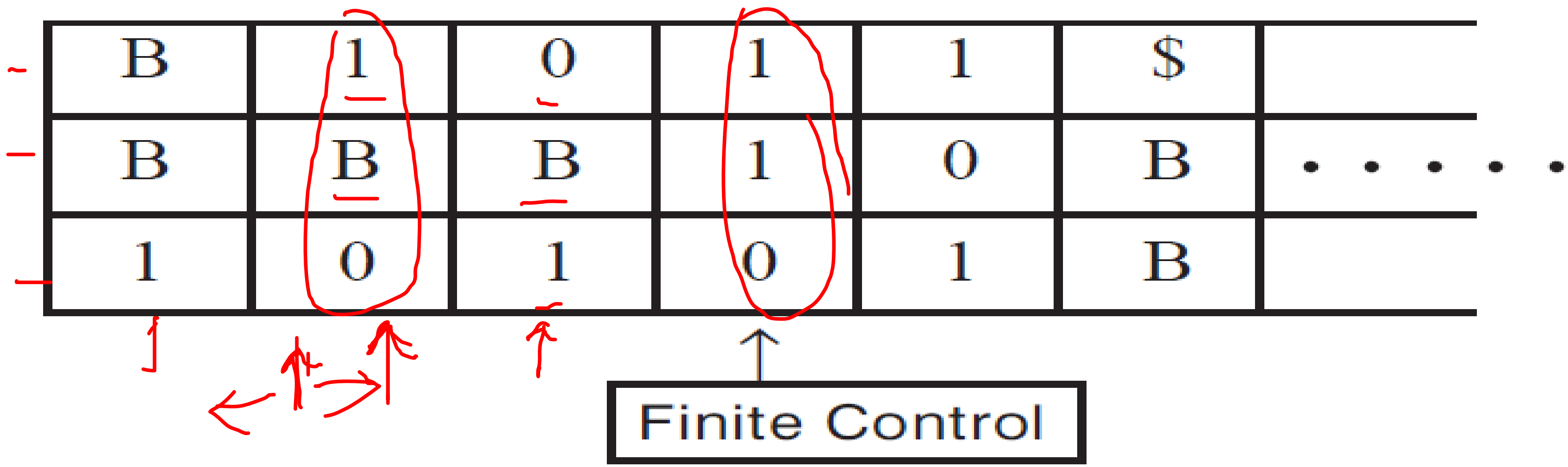
$$\delta([q_1,0],1) = ([q_1,0],\ 1,\ R)$$

$$\delta([q_1,1],0) = ([q_1,1],\ 0,\ R)$$

$$\delta([q_1,0],B) = ([q_1,B],\overset{B}{0},L)$$

$$\delta([q_1,1],\ B) = ([q_1,B],\ \overset{B}{1},\ L)$$

MULTIPLE TRACKS



$\delta(q_0, \underline{0}) = (q_1, \underline{X}, R)$

$[B, B, 1]$

$[1, 1, 0]$

$\delta(q_0, [1, 1, 0])$

CHECKING OFF SYMBOLS

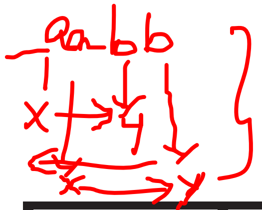
$\{wcw^R \,/\, w \text{ in } \Sigma^*\}$

$\{a^i b^i : i \geq 1\}$

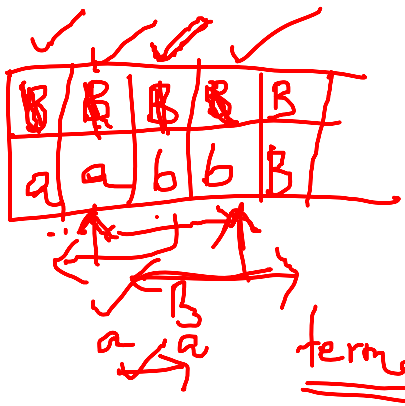
L = {a a, a, b, aaa, aba ....} (ε a | b)

Multiple track

storage in finite control.



B	B	B	B	✓	✓	....	extra Track
a	*a*	*b*	*c*	*a*	*b*	....	



CHECKING OFF SYMBOLS

Consider a turing machine $M = (Q, \Sigma, \Gamma, \delta, q0, B, F)$ for the language

$L = \{wcw \mid w \in \{a, b\}^+\}$

a) $Q = \{[q,d] \mid q = q_1\ q_2 \ldots q_9$ and $d = a, b$ or $B\}$

b) $\Sigma = \{[B, d] \mid d = a, b$ or $c\}$

c) $\Gamma = \{[x, d] \mid x = B$ or and $d = a, b, c$ or $B\}$

d) $q0 = [q_1, B]$

e) $F = \{[q_9, B]\}$

f) $B = [B, B]$

g) δ is defined for $d = a$ or b and $e = a$ or b.

CHECKING OFF SYMBOLS

Forward

d → a | b

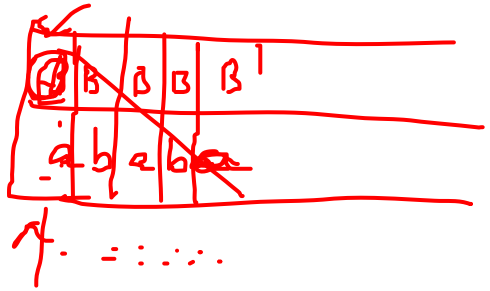
$\delta([q_1, B], [B, d]) = ([q_2, d], [\checkmark, d], R)$

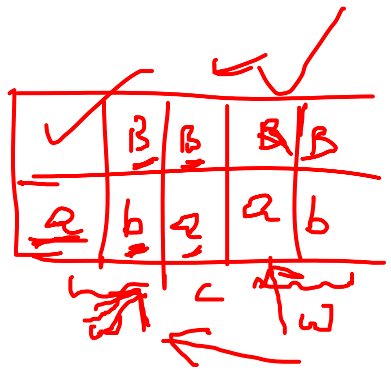
$\delta([q_2, d], [B, e]) = ([q_2, d], [B, e], R)$

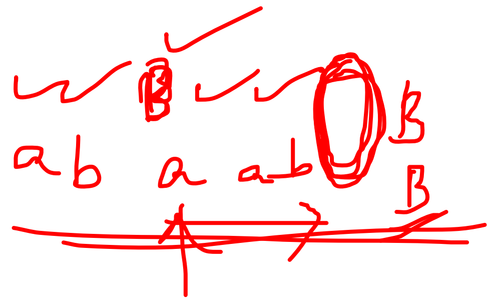
$\delta([q_2,d], [B, c]\} = ([q_3, d], [B, c], R)$

$\delta([q_3,d], [\checkmark, e]) = ([q_3,d], [\checkmark, e], R)$

$\delta([q_3,d], [B, d]) = ([q_4,B], [\checkmark, d], L)$







CHECKING OFF SYMBOLS

Backward

$\delta([q_4,B], [\checkmark, d]) = ([q_4,B], [\checkmark, d], L)$

$\delta([q_4,B], [B, c]) = ([q_5,B], [B, c], L)$

$\delta([q_5,B], [B, d]) = ([q_6,B], [B, d], L)$

$\delta([q_6,B], [B, d]) = ([q_6,B], [B, d], L)$

$\delta([q_6,B], [\checkmark, d]) = ([\underline{q_1},B], [\checkmark, d], R)$

CHECKING OFF SYMBOLS

Terminate

$\delta([q_5,B], [\checkmark, d]) = ([q_7,B], [\checkmark, d], R)$

$\delta([q_7,B], [B, c]) = ([q_8,B], [B, c], R)$

$\delta([q_8,B], [\checkmark, d]) = ([q_8,B], [\checkmark, d], R)$

$\delta([q_8,B], [B, B]) = ([q_9,B], [, B], L)$

SUBROUTINES

$\delta(q0,0) = (q6, B, R)$

$\delta(q6, 0) = (q6, 0, R)$

$\delta(q6, 1) = (q1, 1, R)$

$f(m,n) = m+n$

$f(m,n) = m*n$

m | n

000|00

m*n

00000

2*2 = 4

m n

00|00| result

3*2

B0100100

B100|0000

0000

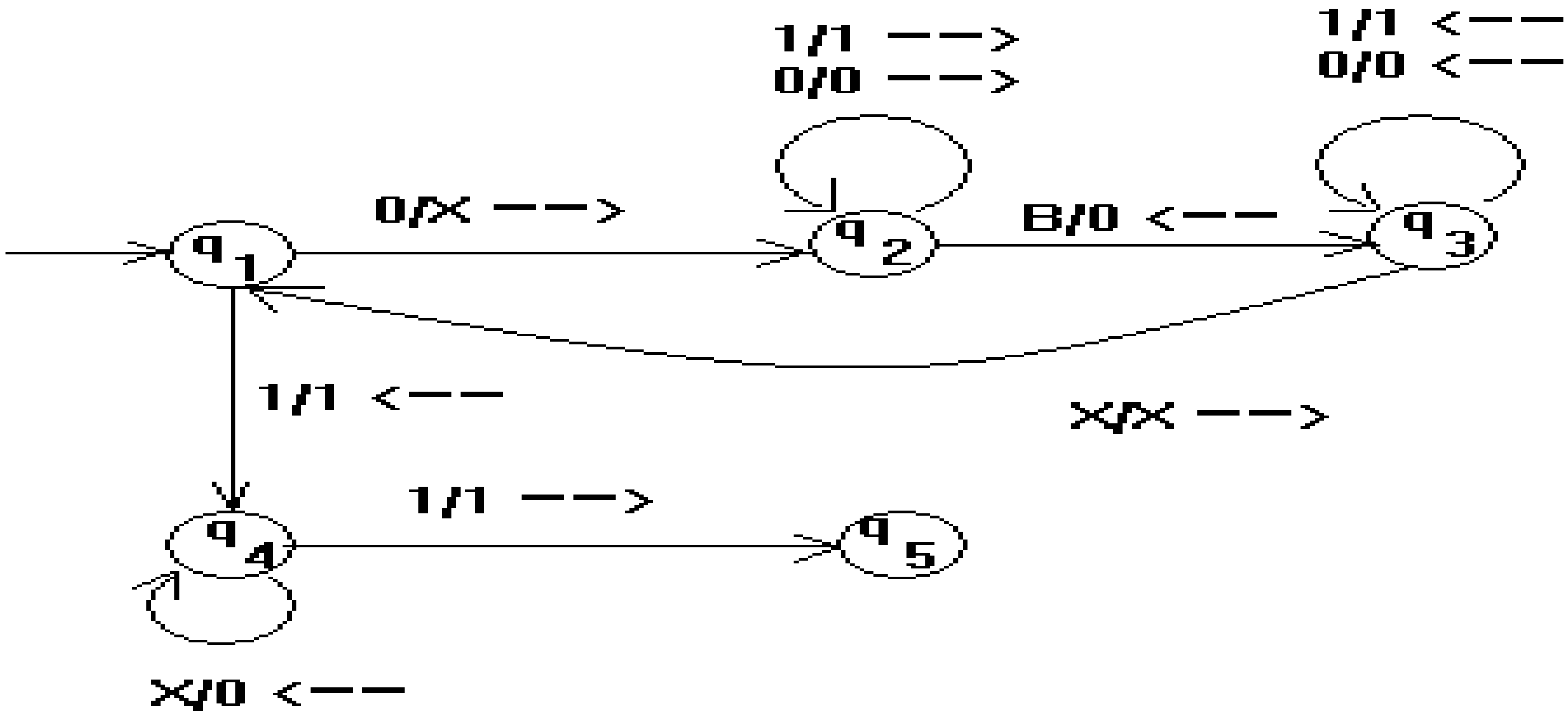
δ for subroutine COPY.

States	Inputs			
	0	1	2	B
q_1	$(q_2, 2, R)$	$(q_4, 1, L)$		
q_2	$(q_2, 0, R)$	$(q_2, 1, R)$		$(q_3, 0, L)$
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	$(q_1, 2, R)$	
q_4		$(q_5, 1, R)$	$(q_4, 0, L)$	

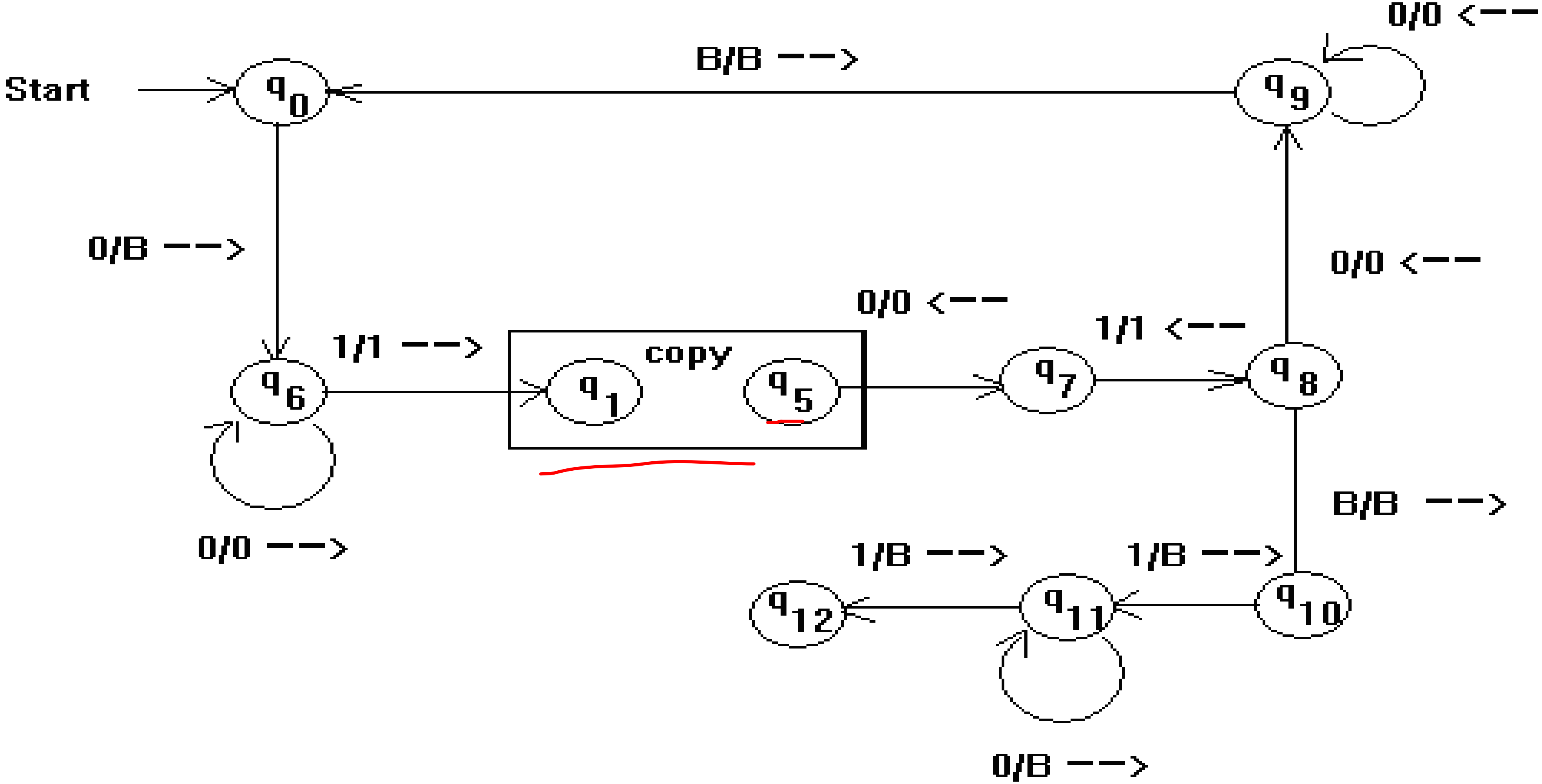
SUBROUTINES

States	Inputs			
	0	1	2	B
q_5	$(q_7, 0, L)$			
q_7		$(q_8, 1, L)$		
q_8	$(q_9, 0, L)$			(q_{10}, B, R)
q_9	$(q_9, 0, L)$			$(q_0, B, R$
q_{10}		(q_{11}, B, R)		
q_{11}	(q_{11}, B, R)	(q_{12}, B, R)		

SUBROUTINES



SUBROUTINES



$\delta(q_0, 001001)$ |– $Bq_6 01001B$

|– $B0q_6 1001B$

|– $B01q_1 001B$

|– $B01Xq_2 01B$

|– $B01X0q_2 1B$

|– $B01X01q_2 B$

|– $B01X0q_3 10$

|– $B01Xq_3 010$

|– $B01q_3 X010$

|– $B01Xq_1 010$

|– $B01XXq_2 10$

|– $B01XX1q_2 0$

|– $B01XX10q_2 B$

|– $B01XX1q_3 00$

|– $B01XXq_3 100$

|– $B01Xq_3 X100$

|– $B01XXq_1 100$

|– $B01Xq_4 X100$

|– $B01q_4 X0100$

|– $B0q_4 100100$

|– $B01q_5 00100$

|– $B0q_7 100100$

2x2

|– $Bq_8 0100100$

|– $q_9 B0100100$ (

|– $Bq_0 0100100$

|– $BBq_6 100100$

|– $BB1q_1 00100$

|– $BB1Xq_2 0100$

|– $BB1X0q_2 100$

|– $BB1X0q_2 100$

|– $BB1X01q_2 00$

|– $BB1X010q_2 0$

|– $BB1X0100q_2 B$

|– $BB1X010q_3 00$

|– $BB1X01q_3 000$

|– $BB1X0q_3 1000$

|– $BB1Xq_3 01000$

|– $BB1q_3 X01000$

|– $BB1Xq_1 01000$

|– $BB1XXq_2 1000$

|– $BB1XX1q_2 000$

|– $BB1XX10q_2 00$

|– $BB1XX100q_2 0$

|– $BB1XX1000q_2 B$

|– $BB1XX100q_3 00$

|– $BB1XX10q_3 000$

|– $BB1XX1q_3 0000$

|– $BB1XXq_3 10000$

|– $BB1Xq_3 X10000$

|– $BB1XXq_1 10000$

SUBROUTINES

$\vdash BB1Xq_4X10000$

$\vdash BB1q_4X010000$

$\vdash BBq_4 10010000$

$\vdash BB1q_5 0010000$

$\vdash BBq_7 10010000$

$\vdash Bq_8 B10010000$

$\vdash BBq_{10} 10010000$

$\vdash BBBq_{11} 0010000$

$\vdash BBBBq_{11} 010000$

$\vdash BBBBBq_{11} 10000$

$\vdash BBBBBBq_{12} 0000$

SUMMARY

- Definition of Total and partial function
- Computing a numerical function using Turing Machine
- Programming techniques of Turing Machine

TEST YOUR KNOWLEDGE

- Which of the following statements is/are FALSE?

1. For every non-deterministic Turing machine, there exists an equivalent deterministic Turing machine.
2. Turing recognizable languages are closed under union and complementation.
3. Turing decidable languages are closed under intersection and complementation.
4. Turing recognizable languages are closed under union and intersection.

A. 1 and 4 only

B. 1 and 3 only

C. 2 only

D. 3 only

TEST YOUR KNOWLEDGE

- Which of the following is true for the language

$$L=\{a^p / \ p \text{ is prime}\}$$

A. It is not accepted by a Turing Machine

B. It is regular but not context-free

C. It is context-free but not regular

D. It is neither regular nor context-free, but accepted by a Turing machine

REFERENCE

- Hopcroft J.E., Motwani R. and Ullman J.D, “Introduction to Automata Theory, Languages and Computations”, Second Edition, Pearson Education, 2008