# Thadomal Shahani Engineering College

# Certificate

This is to certify that Mr. _Tushar Nankani__ of Department of Computer Engineering, in Semester __V__ with Batch __C23__ and Roll No __1902112__ has completed the course of necessary assignments in the subject of 'Theory of Computer Science' successfully under my supervision in the 'Thadomal Shahani Engineering College' during academic year 2021 – 22

Prof. Vaibhav Ambhire
Teacher – in – charge

Dr. Tanuja Sarode
Head of the Department

Dr. G.T. Thampi
Principal

Date: ____24-10-2021____

# Content

TUSHAR NANKANI
1902112 C2

Page No.
Date

THEORY OF Computer Science.
⇒ ASSIGNMENT-I

Q1. Design Regular Expression-

(1) Set of all strings that end with 1 & has substring 00.

$\Sigma = \{0,1\}$
$r = (0+1)^*.(00).(0+1)^*.1$

∴ possible set = {001, 000101, 011...}

(2) language with all string beginning with aa.

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

$r = (aa).(b+a)^*$

possible set = {aa, aab, aaa...}

(3) Set of all strings ending with 101 or 111

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

$r_1 = (0+1)^*.(101)$
$r_2 = (0+1)^*.(111)$
$r_1 + r_2 = r$

$r = (0+1)^*.(101) + (0+1)^*.(111)$

possible set = {101, 111, 0101, 1111...}

Page No.
Date :

(4) set of strings containing 010 as substring

$\Sigma = \{0, 1, 2\}$

$r = (0+1+2)^* . (010) . (0+1+2)^*$

(5) set of all strings with odd 1's & even 0's

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

$r = (11+0)^* . (1) . (11)^* . (00)^* (00+1)^*$

(6) Set of substring that do not contain substring 110.

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

$r = (0+10)^* 1^*$

possible set = $\{0, 10, 01, 101 \ldots\}$

Page No.

Date :

Q2. Design a DFA :

(1) To accept a string which contains exactly 3 'a's

Soln

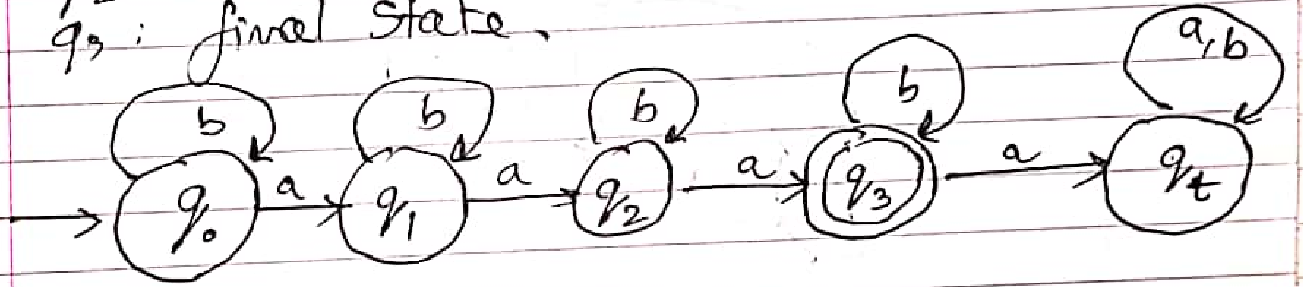
$q_0$ : initial

$q_1$ : state with 1 a

$q_2$ : state with 2 a

$q_3$ : final state.

$q_t$ = trap state




$\delta(q_0, a) = q_1$

$\delta(q_0, b) = q_0$

$\delta(q_1, a) = q_2$

$\delta(q_1, b) = q_1$

$\delta(q_2, a) = q_3$

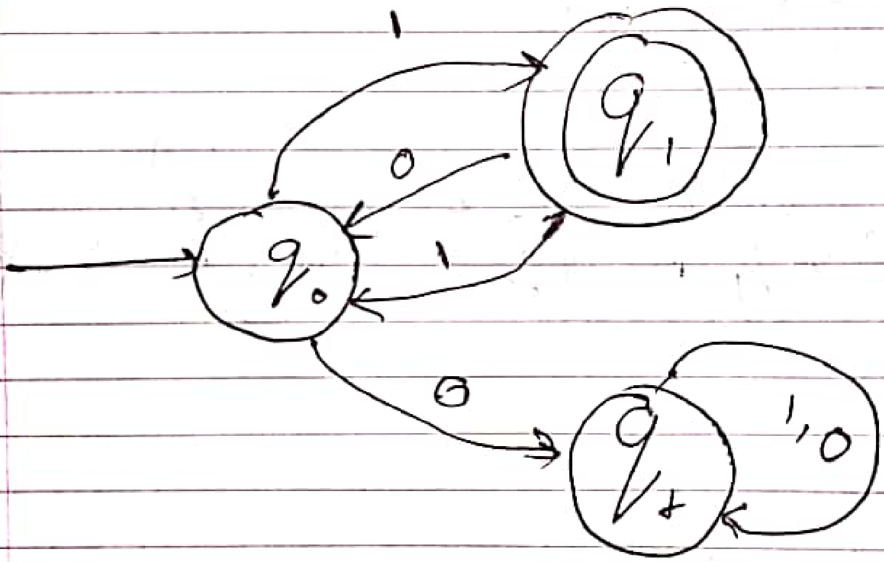
$\delta(q_2, b) = q_2$

$\delta(q_3, b) = q_3$

$\delta(q_3, a) = q_t$

② To accept the string in which every odd pos$^n$ is 1.

Ans: $q_0$ = initial $\quad$ $q_t$ = trap
$q_1$ = final




$\delta(q_0, 1) = q_1$
$\delta(q_0, 0) = q_t$
$\delta(q_1, 1) = q_0$
$\delta(q_t, 1) = q_t$
$\delta(q_1, 0) = q_0$

(3) To check whether given binary is divisible by 4

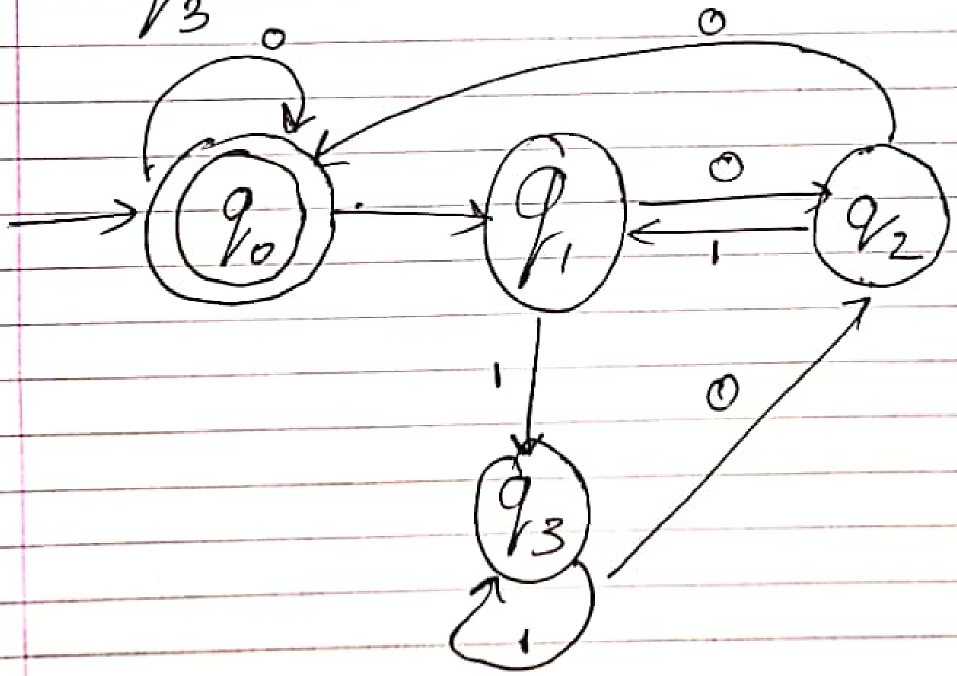
— last 2 ⇒ 00

⇒ $q_0$ = final /4

$q_1$ = remainder 1

$q_2$ = 2

$q_3$ = 3



Rishika Lulla
C2 - 1902087



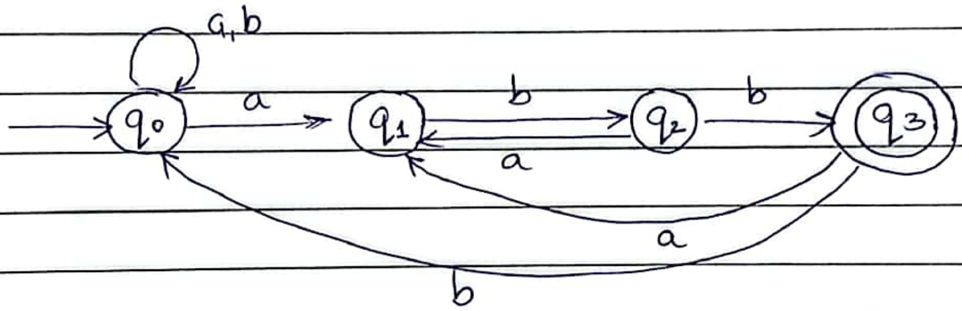
# ASSIGNMENT - 2

Q1.

I] Construct NFA that accepts a set of all strings over $\{a,b\}$ ending with "abb". Convert the NFA to equivalent DFA.

→ $\Sigma = \{a,b\}$

R.E. $= (a+b)^* . abb$




$Q = \{q_0, q_1, q_2, q_3\}$

$q_0 = \{q_0\}$

$F = \{q_3\}$

| Input/States | a | b |
|---|---|---|
| $q_0$ | $\{q_0, q_1\}$ | $\{q_0\}$ |
| $q_1$ | $\{q_1\}$ | $\{q_2\}$ |
| $q_2$ | $\{q_1\}$ | $\{q_3\}$ |
| $q_3$ | $\{q_1\}$ | $\{q_0\}$ |

Input string: aabb

$\delta(q_0, aabb) \vdash \delta(q_0, abb)$

$\vdash \delta(q_1, bb)$

$\vdash \delta(q_2, b)$

$\vdash \delta(q_3, \epsilon)$

$\vdash \underline{q_3}$

$\therefore$ $q_3$ is the final state.

Conversion to DFA:

We have new state : $\{q_0, q_1\}$

$\therefore \delta(\{q_0, q_1\}, a) = \delta(q_0, a) \cup \delta(q_1, a)$
$= \{q_0, q_1\} \cup \{q_1\}$
$= \{q_0, q_1\}$

$\therefore \delta(\{q_0, q_1\}, b) = \delta(q_0, b) \cup \delta(q_1, b)$
$= \{q_0\} \cup \{q_2\}$
$= \{q_0, q_2\}$ ... NS

$\therefore \delta(\{q_0, q_2\}, a) = \delta(q_0, a) \cup \delta(q_2, a)$
$= \{q_1, q_0\} \cup \{q_1\}$
$= \{q_0, q_1\}$

$\therefore \delta(\{q_0, q_2\}, b) = \delta(q_0, b) \cup \delta(q_2, b)$
$= \{q_0\} \cup \{q_3\}$
$= \{q_0, q_3\}$ ... NS

$\therefore \delta(\{q_0, q_3\}, a) = \delta(q_0, a) \cup \delta(q_3, a)$
$= \{q_0, q_1\} \cup \{q_1\}$
$= \{q_0, q_1\}$

$\therefore \delta(\{q_0, q_3\}, b) = \delta(q_0, b) \cup \delta(q_3, b)$
$= \{q_0\} \cup \{q_0\}$
$= \{q_0\}$

PAGE No
DATE / /

Transition table:

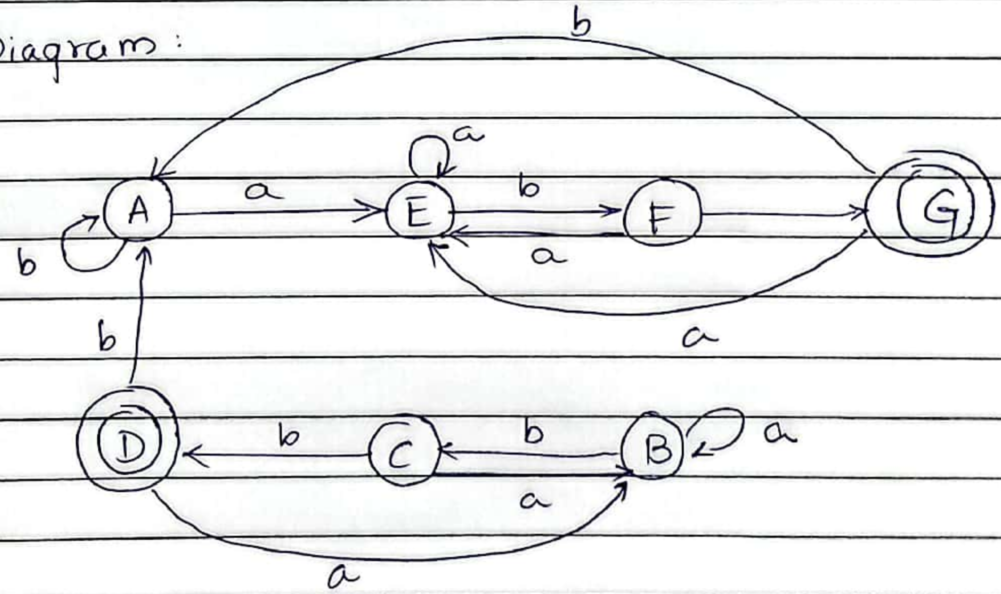
| Input/states | a | b |
|---|---|---|
| $\{q_0\}$ | $\{q_0, q_1\}$ | $\{q_0\}$ |
| $\{q_1\}$ | $\{q_1\}$ | $\{q_2\}$ |
| $\{q_2\}$ | $\{q_1\}$ | $\{q_3\}$ |
| $\{q_3\}$ | $\{q_1\}$ | $\{q_0\}$ |
| $\{q_0, q_1\}$ | $\{q_0, q_1\}$ | $\{q_0, q_2\}$ |
| $\{q_0, q_2\}$ | $\{q_0, q_1\}$ | $\{q_0, q_3\}$ |
| $\{q_0, q_3\}$ | $\{q_0, q_1\}$ | $\{q_0\}$ |

Rename the states as:

$\{q_0\} = A$ $\quad$ $\{q_0, q_1\} = E$
$\{q_1\} = B$ $\quad$ $\{q_0, q_2\} = F$
$\{q_2\} = C$ $\quad$ $\{q_0, q_3\} = G$
$\{q_3\} = D$

$\therefore$ In DFA, D and G are final states.

Diagram:

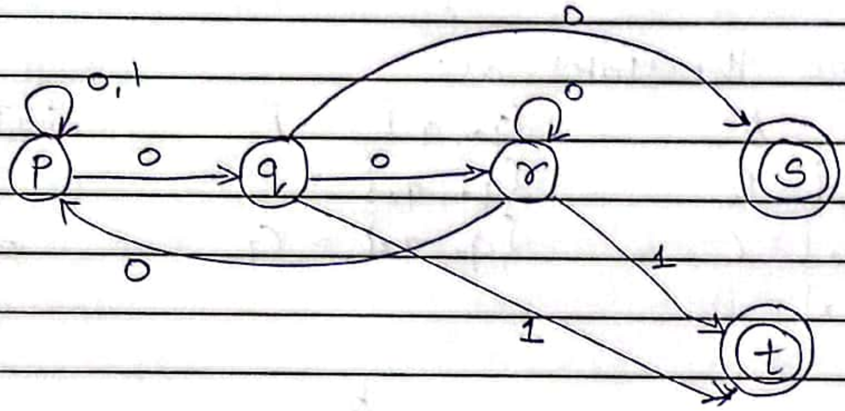



$M = \{Q, \Sigma', \delta, q_0', F'\}$
$Q = \{A, B, C, D, E, F, G\}$
$q_0 = A$
$F = \{D, G\}$

Q1
II) Convert the following NFA to DFA:
p is the intial state, s and t are final states.

| | 0 | 1 |
|---|---|---|
| p | {p,q} | {p} |
| q | {r,s} | {t} |
| r | {p,r} | {t} |
| s* | $\phi$ | $\phi$ |
| t* | $\phi$ | $\phi$ |

$\Rightarrow$ NFA




We have a new state {p,q}

$\therefore \delta(\{p,q\},0) = \delta(p,0) \cup \delta(q,0)$
$= \{p,q\} \cup \{r,s\}$
$= \{p,q,r,s\}$ ... NS

$\therefore \delta(\{p,q\},1) = \delta(p,1) \cup \delta(q,1)$
$= \{p\} \cup \{t\}$
$= \{p,t\}$ ... NS

$\therefore \delta(\{p,q,r,s\},0) = \delta(p,0) \cup \delta(q,0) \cup \delta(r,0) \cup (s,0)$
$= \{p,q\} \cup \{r,s\} \cup \{p,r\} \cup \phi$
$= \{p,q,r,s\}$

PAGE No.
DATE / /

$\therefore \delta(\{p,q,r,s\}, 1) = \delta(p,1) \cup \delta(q,1) \cup \delta(r,1) \cup \delta(s,1)$
$= \{p\} \cup \{t\} \cup \{t\} \cup \phi$
$= \{p,t\}$

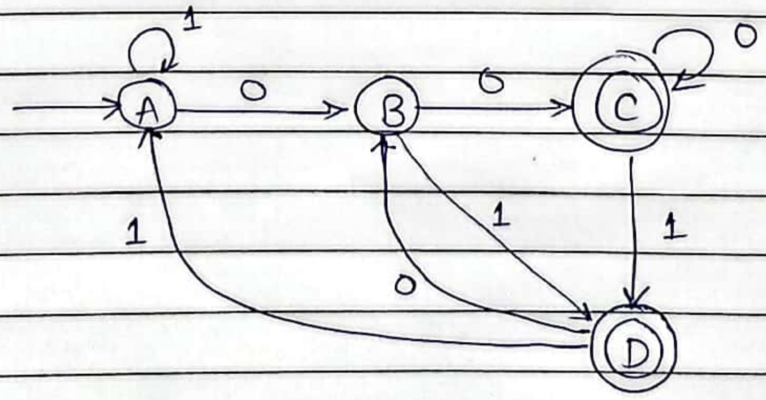
$\therefore \delta(\{p,t\}, 0) = \delta(p,0) \cup \delta(t,0)$
$= \{p,q\} \cup \phi$
$= \{p,q\}$

$\therefore \delta(\{p,t\}, 1) = \delta(p,1) \cup \delta(t,1)$
$= \{p\} \cup \phi$
$= \{p\}$

Rename the states as:
$\{p\} = A$
$\{p,q\} = B$
$\{p,q,r,s\} = C$
$\{p,t\} = D$

| | 0 | 1 |
|---|---|---|
| → A | B | A |
| B | C | D |
| * C | C | D |
| * D | B | A |

$\therefore$ C & D are final states




DFA

| PAGE No. | |
|---|---|
| DATE | / / |

Q2. Construct NFA with epsilon transition for following RE:

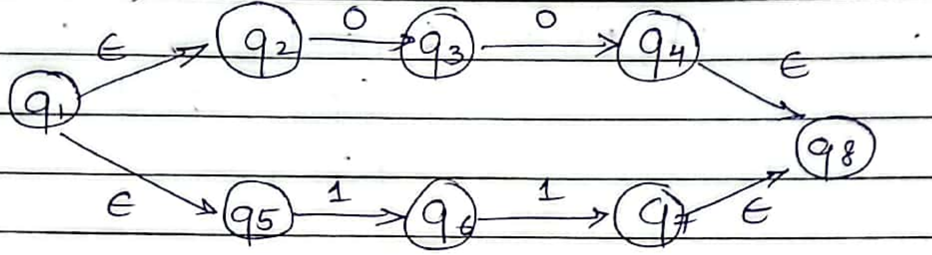
$$(00+11)^*.(10)^*$$

→ let

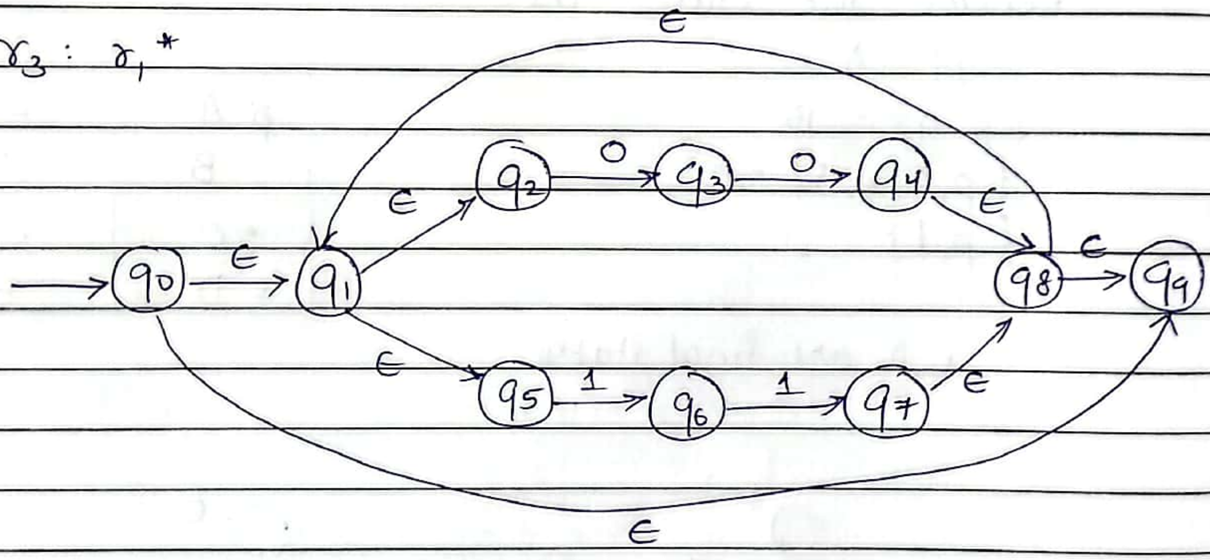
$r_1 = (00+11)$ $\quad$ $r_2 = (10)^*$

$r_3 = r_1^*$ $\quad$ R.E := $r_3 . r_2$

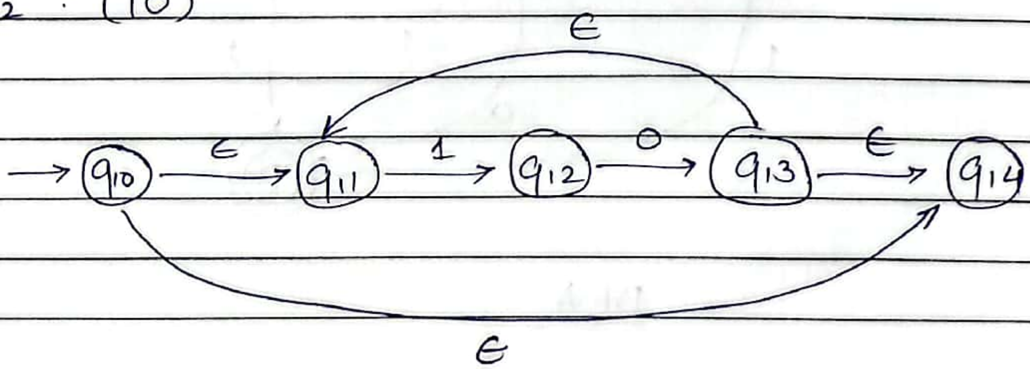
$r_1 : (00+11)$



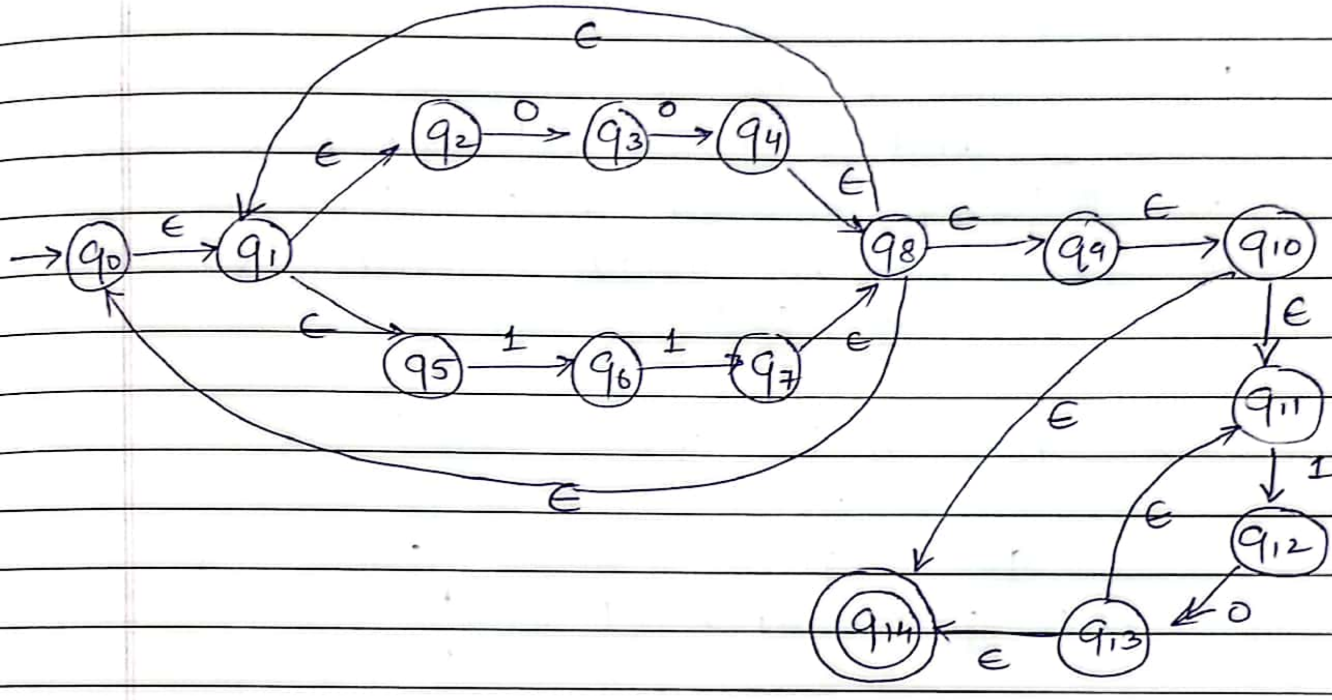
$r_3 : r_1^*$



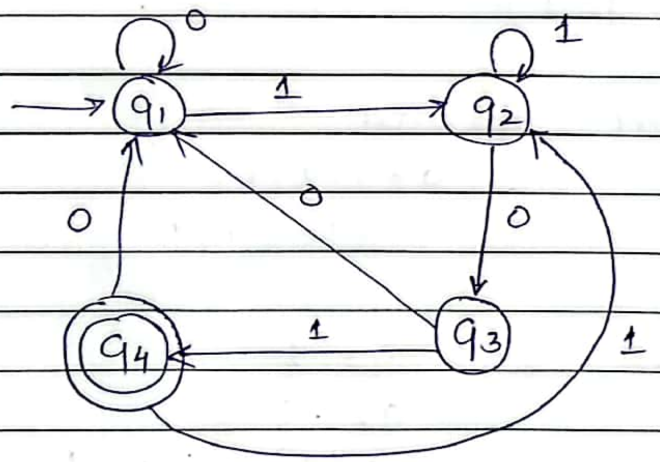
$r_2 : (10)^*$



$\therefore \quad RE = (00+11)^{*}.(10)^{*} = r_3 \cdot r_2$




II) State Arden's theorem and use it to construct a regular expression corresponding to the following automata:




$\rightarrow$

$q_1 = q_1.0 + q_3.0 + q_4.0 + \epsilon \longrightarrow (1)$

$q_2 = q_1.1 + q_2.1 + q_4.1 \longrightarrow (2)$

$q_3 = q_2.0 \longrightarrow (3)$

$q_4 = q_3.1 \longrightarrow (4)$

Arden's theorem states that:

"If P and Q are two regular expressions over $\Sigma$, and if P does not contain $\epsilon$, then the following equating in R given by $R = Q + RP$ has an unique solution, i.e.

$$\underline{R = QP^*}.$$

Now,
$$q_2 = q_1.1 + q_2.1 + q_3.1.1$$
$$= q_1.1.1 + q_2.1 + q_2.0.1.1$$
$$q_2 = q_1.1 + (1+011)\,q_2$$
$$\therefore q_2 = q_1.1\,(1+011)^*$$

And,
$$q_1 = q_1.0 + q_2.0.0 + q_2.0.1.0 + \epsilon$$
$$= q_1.0 + q_2(00+010) + \epsilon$$
$$= q_1.0 + (q_1.1)(1+011)^*.(00+010) + \epsilon$$

$$\therefore q_1 = q_1\left(0 + 1.(1+011)^*.(00+010)\right) + \epsilon$$

Using Arden's theorem,
$$q_1 = \epsilon.\left(0+1.(1+011)^*.(00+010)\right)$$
$$\therefore q_1 = \left(0+1.(1+011)^*.(00+010)\right)^*$$

$$\therefore RE = q_4 = q_3.1$$
$$= q_2.0.1$$
$$= (q_1.1)(1+011)^*.0.1$$

**Ans:** $\therefore RE = \left(0+1.(1+011)^*.(00+010)\right)^*.1\,(1+011)^*.0.1$

TCS Assignment No 3

Q1. Convert the following grammer in CNF

$S \rightarrow aABC$

$A \rightarrow aB \mid a$ The grammar is in reduced form.

$B \rightarrow bA \mid b$

$C \rightarrow a$

Soln

$R_1 \rightarrow a$ [CNF]

$R_2 \rightarrow BC$ [CNF]

$\therefore S \rightarrow R_1 A R_2$

$R_3 \rightarrow AR_2$ [CNF]

$\therefore S \rightarrow R_1 R_3$ [CNF]

$A \rightarrow R_1 B$ [CNF]

$A \rightarrow a$ [CNF]

$R_4 \rightarrow b$ [CNF]

$B \rightarrow R_4 A$ [CNF]

$B \rightarrow b$ [CNF]

$C \rightarrow a$ [CNF]

Ans

| | |
|---|---|
| $S \rightarrow R_1 R_3$ | $R_1 \rightarrow a$ |
| $A \rightarrow R_1 B$ | $R_4 \rightarrow b$ |
| $A \rightarrow a$ | $R_2 \rightarrow BC$ |
| $B \rightarrow R_4 A$ | $R_3 \rightarrow AR_2$ |
| $B \rightarrow b$ | |
| $C \rightarrow a$ | |

Q2 Reduce the following grammar in GNF

$S \rightarrow AB$

$A \rightarrow BSB \mid BB \mid b$

Sol^n The grammar is not reduced.

Symbol B is useless So we remove all the productions having symbol B.

~~Resultant Grammar~~.

$S \rightarrow AB$ These productions are discarded.

$A \rightarrow BSB \mid BB$

Resultant Grammar

$A \rightarrow b$ [GNF]

Q3 Obtain leftmost derivation, rightmost derivation and derivation tree for the String :- cccbaccba

The grammar is $S \rightarrow SSa \mid SSb \mid c$

Sol^n leftmost derivation.

$S \rightarrow SSa$

$\rightarrow SSaSa$

$\rightarrow cSaSa$

$\rightarrow cSSbaSa$

$\rightarrow ccSbaSa$

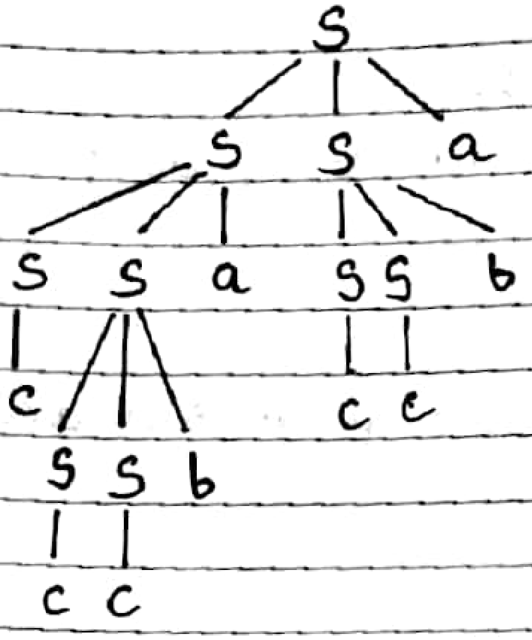
$\rightarrow cccbaSa$

$\rightarrow cccbaSSba$

$\rightarrow cccbacSba$

$\rightarrow cccbaccba$

derivation tree.




Right most derivation

S → SSa

→ SSSba

→ SScba

→ Sccba

→ SSaccba

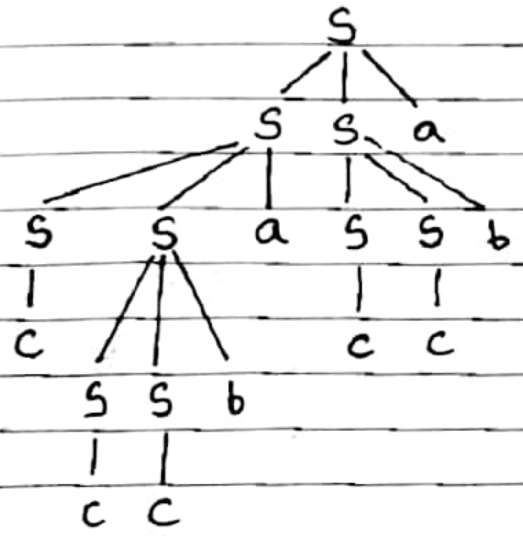
→ SSSbaccba

→ SScbaccba

→ Sccbaccba

→ cccbaccba

derivation tree.



**Q5** Consider the following grammar

$S \to iCtS \mid iCtSeS \mid a$

$C \to b$ String :- ibtibtaea

**Sol^n** leftmost derivation.

$S \to iCtSeS$

$\to ibtSeS$

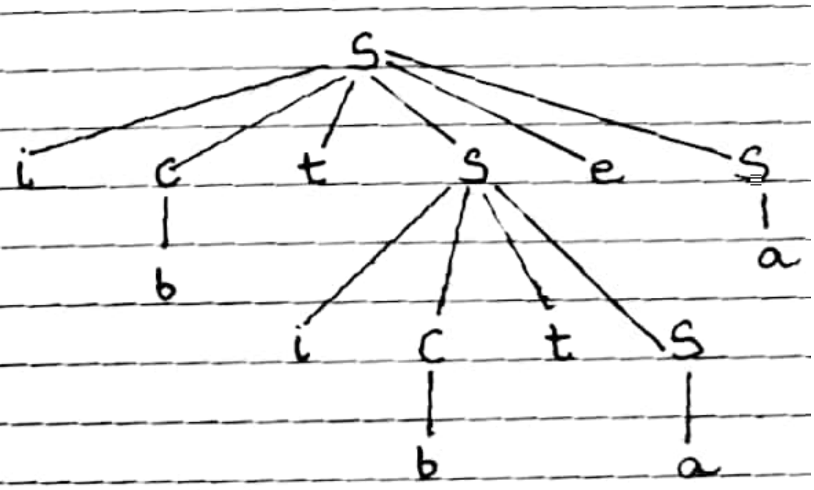
$\to ibtiCtSeS$

$\to ibtibtSeS$

$\to ibtibtaeS$

$\to ibtibtaea$

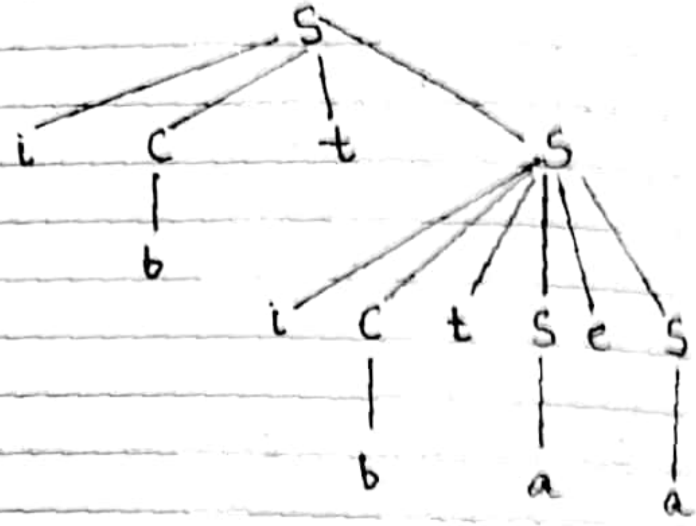
Parse tree



Right Most derivation

$S \rightarrow iCtS$
$\rightarrow iCtiCtSeS$
$\rightarrow iCtiCtSea$
$\rightarrow iCtiCtaea$
$\rightarrow iCtibtaea$
$\rightarrow ibtibtaea$

Parse tree



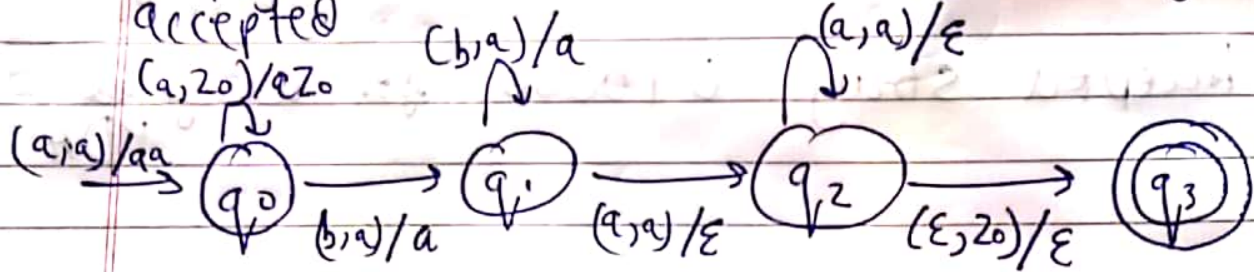

Since there are more than one parse tree for a given string The given grammar is ambiguous.

Q I

II Construct PDA for accepting following Language

$L(M) = \{a^n b^m a^n \mid m, n >= 1\}$

First All 'a' will be pushed onto stack
On reading b neither a or b will be pushed or popped.
On reaching 'a' one stack symbol 'a' will be popped out.
When string is ended and stack gets empty at the same time then string gets accepted




Transition Function

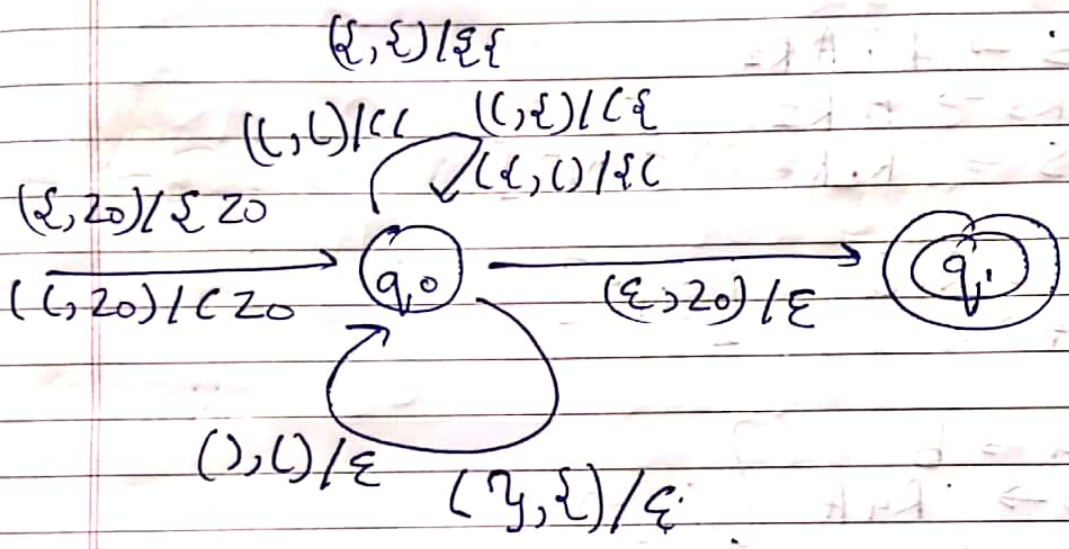
$\delta(q_0, a, Z_0) = (q_0, a, Z_0)$
$\delta(q_0, a, a) = (q_0, a, a)$
$\delta(q_0, b, a) = (q_1, a)$
$\delta(q_1, b, a) = (q_1, a)$
$\delta(q_1, a, a) = (q_2, \varepsilon)$
$\delta(q_2, a, a) = (q_2, \varepsilon)$
$\delta(q_2, \varepsilon, Z_0) = (q_3, \varepsilon)$

PDA defined $M = (\{q_0, q_1, q_2, q_3\}, \{a, b\}, \{a, Z_0\}, q_0, Z_0, \{q_3\}, \delta)$

IV Design PDA for well balanced parenthesis

If we read same or non complementary symbol we will push when checked with top of stack

If we read complementary symbol as top of stack then we perform pop operation




$\delta(q_0, (, Z_0) = (q_0, (Z_0)$
$\delta(q_0, \{, Z_0) = (q_0, \{Z_0)$
$\delta(q_0, (, () = (q_0, (()$
$\delta(q_0, \{, \{) = (q_0, \{\{)$
$\delta(q_0, (, \{) = (q_0, (\{)$
$\delta(q_0, \{, () = (q_0, \{()$
$\delta(q_0, ), () = (q_0, \varepsilon)$
$\delta(q_0, \}, \{) = (q_0, \varepsilon)$
$\delta(q_0, \varepsilon, Z_0) = (q_1, \varepsilon)$

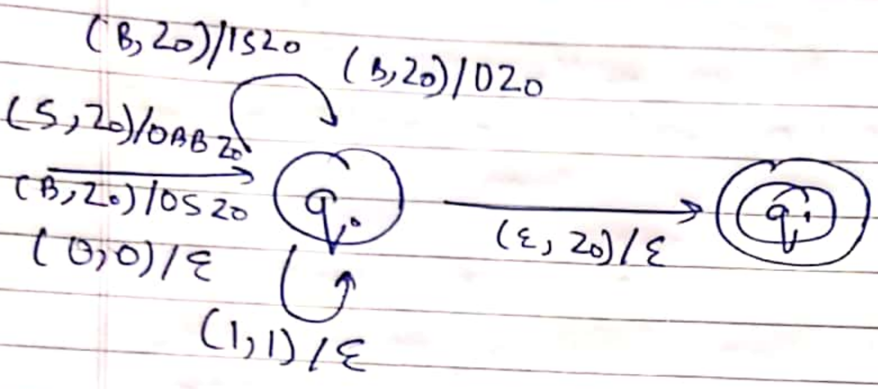
PDA defined $M = (\{q_0, q_1\}, \{, (, ), \{, \}\}, \{(, \{, Z_0\}, q_0, Z_0, \{q_1\}, \delta\}$

V Construct PDA equivalent to following grammar

$S \rightarrow 0BB$

$B \rightarrow 0S \mid 1S \mid 0$ .

Show that 010000 is accepted by PDA




$\delta(q_0, \varepsilon, S) = (q_0, 0BB)$

$\delta(q_0, \varepsilon, B) = (q_0, 0S)$

$\delta(q_0, \varepsilon, B) = (q_0, 1S)$

$\delta(q_0, \varepsilon, B) = (q_0, 0)$

$\delta(q_0, 1, 1) = (q_0, \varepsilon)$

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

PDA $= M = (\{q\}, \{0, 1\}, \{S, B, 0, 1\}, \delta, q, S, \phi\}$

$(q_0, 010000, S)$

$(q_0, 010000, 0BB)$ $\quad S = 0BB$

$(q_0, 10000, BB)$

$(q_0, 10000, 1SB)$ $\quad B = 1S$

$(q_0, 0000, SB)$

$(q_0, 0000, 0BBB)$

$(q_0, 000, BBB)$ $\quad B = 0$

$(q_0, 000, 0BB)$

$(q_0, 00, 0B)$.

$(q_0, 0, B)$

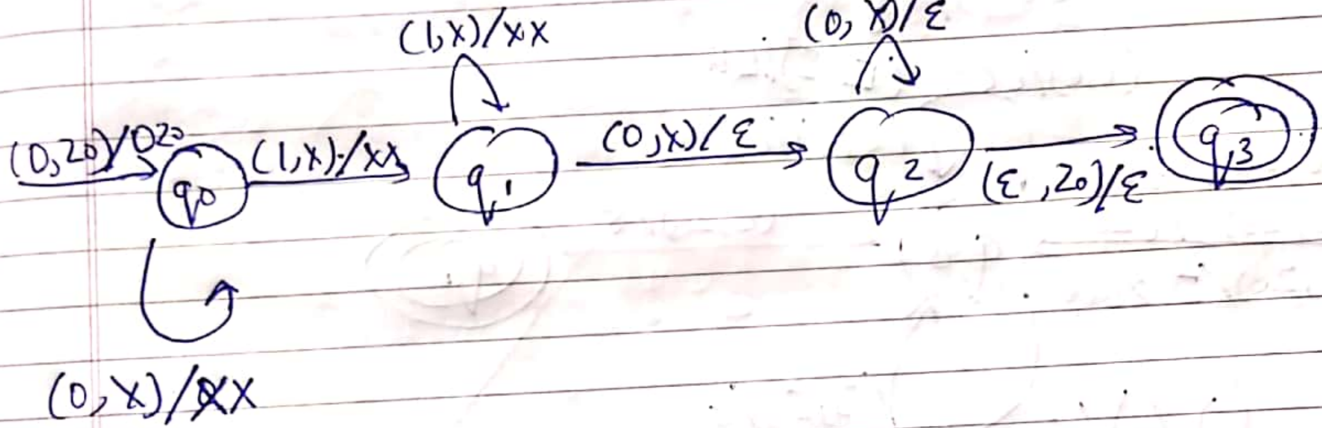
$(q_0, 0, 0)$

$(q_0, \epsilon, \epsilon)$

Accepted string 010000 for designed PDA

VI $L(M) = \{ 0^m 1^n 0^{m+n} \mid m, n >= 1 \}$

For initial '0' and '1' we will add 'X' of every occurrence of 0 and 1. After all '1' have been read, first '0' will be read and popping will start when string is ended and stack is empty, we will say that string is accepted.




Transition Function

$\delta(q_0, 0, Z_0) = (q_0, XZ_0)$
$\delta(q_0, 0, X) = (q_0, XX)$
$\delta(q_0, 1, X) = (q_1, XX)$
$\delta(q_1, 1, X) = (q_1, XX)$
$\delta(q_1, 0, X) = (q_2, \varepsilon)$
$\delta(q_2, 0, X) = (q_2, \varepsilon)$
$\delta(q_2, \varepsilon, Z_0) = (q_3, \varepsilon)$

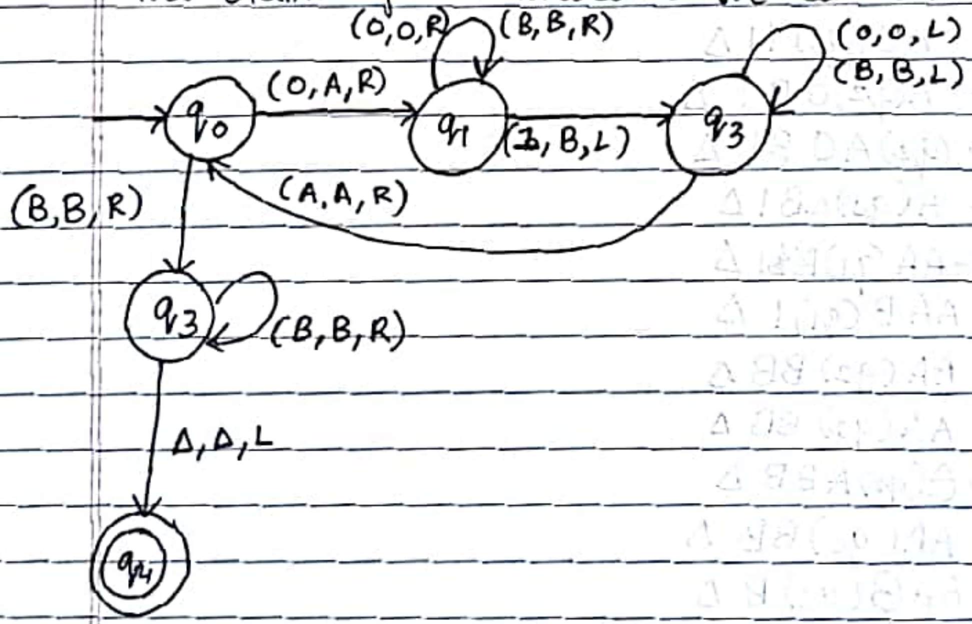
PDA defined $M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \{X, Z_0\}, q_0, Z_0, \{q_3\}, \delta)$

TCS assignment No. 5

Q1 Design Turing Machine to accept language
$L(M) = \{0^n 1^n \mid n > 0\}$

Sol^n Logic

Convert 0 to A and move to Right
Skip all 0 and search for 1
Convert first 1 into B and move to left to find A
Skip 0
when find A move right
Convert immediate 0 to A and move right
Skip B
Convert 1 to B and move left
No 0 present and No 1 present
The blank symbol indicates we can accept the string.



Transition Function

$\delta(q_0, a) = (q_1, A, R)$

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

$\delta(q_1, 1) = (q_2, B, L)$

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

$\delta(q_2, A) = (q_0, A, R)$

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

$\delta(q_2, B) = (q_2, B, L)$

$\delta(q_0, B) = (q_3, B, R)$

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

$\delta(q_3, \Delta) = (q_4, \Delta, L)$

Turing Machine can be defined as.

$M = (Q, \Sigma, \Gamma, \delta, q_0, \Delta, F) = (\{q_0, q_1, q_2, q_3, q_4\}, \{0, 1\}, \{0, 1, A, B\}, \delta, q_0, \Delta, \{q_4\})$

Simulation

$(q_0)0011\Delta \rightarrow A(q_1)011\Delta$

$\rightarrow A0(q_1)11\Delta$

$\rightarrow A(q_2)0B1\Delta$

$\rightarrow (q_2)A0B1\Delta$

$\rightarrow A(q_0)aB1\Delta$

$\rightarrow AA(q_1)BB1\Delta$

$\rightarrow AAB(q_1)1\Delta$

$\rightarrow AA(q_2)BB\Delta$

$\rightarrow AA(q_2)BB\Delta$

$\rightarrow A(q_2)ABB\Delta$

$\rightarrow AA(q_0)BB\Delta$

$\rightarrow AAB(q_3)B\Delta$

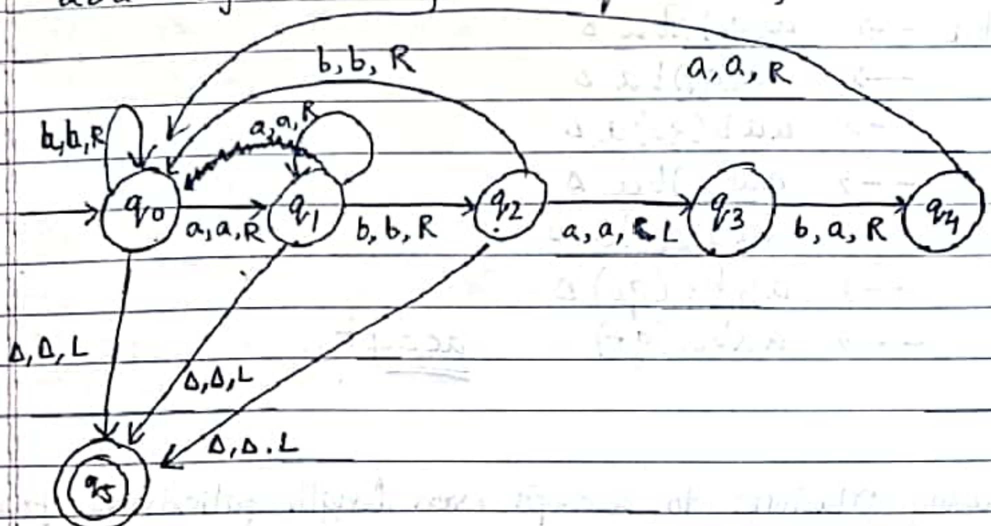
$\rightarrow AABB(q_3)\Delta$

$\rightarrow AAB(q_4)B\Delta$ Accept

Date

**Q2** Design Turing Machine that replaces all occurances of 'aba' by 'aaa' from sequence of a's & b's.

**Sol^n**




Transition function

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

$\delta(q_0, a) = (q_1, a, R)$

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

$\delta(q_1, b) = (q_2, b, R)$

$\delta(q_2, a) = (q_3, a, L)$

$\delta(q_2, b) = (q_0, b, R)$

$\delta(q_3, b) = (q_4, a, R)$

$\delta(q_4, a) = (q_0, a, R)$

$\delta(q_0, \Delta) = (q_5, \Delta, L)$

$\delta(q_1, \Delta) = (q_5, \Delta, L)$

$\delta(q_2, \Delta = (q_5, \Delta, L)$

Turing Machine can be defined as

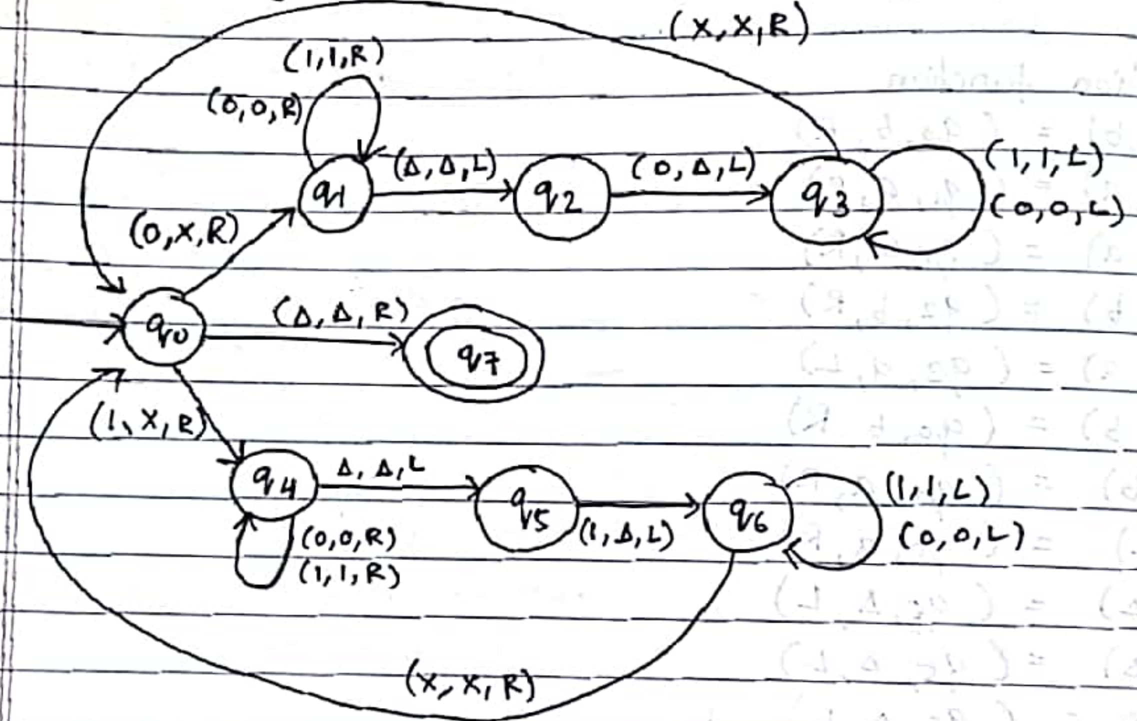
$M = (Q, \Sigma, \Gamma, \delta, q_0, \Delta, F)$

$= (\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{a, b\}, \{a, b\}, \delta, q_0, \Delta, \{q_5\})$

Simulation

$(q_0)aaba \dashrightarrow a(q_1)aba\Delta$
$\dashrightarrow aa(q_1)ba\Delta$
$\dashrightarrow aab(q_2)a\Delta$
$\dashrightarrow aa(q_3)ba\Delta$
$\dashrightarrow aab(q_4)a\Delta$
$\dashrightarrow aaba(q_0)\Delta$
$\dashrightarrow aaba\Delta(q_5)$ accept.

**Q3** Design Turing Machine to accept even length palindrome for {0,1}



Transition function

$\delta(q_0, 0) = (q_1, x, R)$
$\delta(q_0, 1) = (q_4, x, R)$
$\delta(q_1, 1) = (q_1, 1, R)$
$\delta(q_1, 0) = (q_1, 0, R)$
$\delta(q_1, \Delta) = (q_2, \Delta, L)$
$\delta(q_2, 0) = (q_3, \Delta, L)$
$\delta(q_3, 1) = (q_3, 1, L)$
$\delta(q_3, 0) = (q_3, 0, L)$
$\delta(q_3, x) = (q_0, x, R)$

$\delta(q_4, 1) = (q_4, 1, R)$
$\delta(q_4, 0) = (q_4, 0, R)$
$\delta(q_4, \Delta) = (q_5, \Delta, L)$
$\delta(q_5, 1) = (q_6, \Delta, L)$
$\delta(q_6, 1) = (q_6, 1, L)$
$\delta(q_6, 0) = (q_6, 0, L)$
$\delta(q_6, x) = (q_0, x, R)$

$\delta(q_0, \Delta) = (q_7, \Delta, R)$.

Turing Machine can be defined as

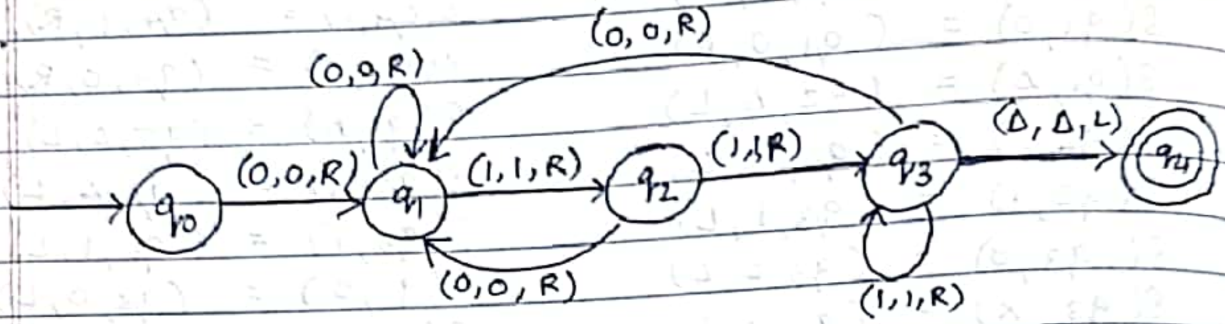
$M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}, \{0, 1\}, \{0, 1, x\}, \delta, q_0, \Delta, \{q_7\})$

Simulation.

$(q_0)0110\Delta \rightarrow x(q_1)110\Delta$
$\rightarrow x1(q_1)10\Delta$
$\rightarrow x11(q_1)0\Delta$
$\rightarrow x110(q_1)\Delta$
$\rightarrow \cancel{x11(q_2)0\Delta}$ $x110(q_2)\Delta$
$\rightarrow \cancel{x1(q_3)1\Delta\Delta}$ $x11(q_3)\Delta\Delta$
$\rightarrow x1(q_3)1\Delta\Delta$
$\rightarrow x(q_3)11\Delta\Delta$
$\rightarrow x(q_0)11\Delta\Delta$
$\rightarrow xx(q_4)1\Delta\Delta$
$\rightarrow xx1(q_4)\Delta\Delta$
$\rightarrow xx1q_5(\Delta)\Delta\Delta$
$\rightarrow xx\Delta q_6\Delta\Delta\Delta$
$\rightarrow xx(q_0)\Delta\Delta\Delta$
$\rightarrow xx\Delta(q_7)\Delta\Delta$ Accept

Q4 Design Turing Machine to accept the language given by regular expression $0(0+1)^*11$

Ans.




Transition Function.

$\delta(q_0, 0) = (q_1, 0, R)$
$\delta(q_1, 0) = (q_1, 0, R)$
$\delta(q_1, 1) = (q_2, 1, R)$
$\delta(q_2, 0) = (q_1, 0, R)$
$\delta(q_2, 1) = (q_3, 1, R)$
$\delta(q_3, 0) = (q_1, 0, R)$
$\delta(q_3, 1) = (q_3, 1, R)$
$\delta(q_3, \Delta) = (q_4, \Delta, L)$

Turing Machine can be defined as
$M = (Q, \Sigma, \Gamma, \delta, q_0, \Delta, F)$

$= (\{q_0, q_1, q_2, q_3, q_4\}, \{0,1\}, \{0,1\}, \delta, q_0, \Delta, \{q_4\})$.

Simulation.

$(q_0)0110111\Delta \longrightarrow 0(q_1)110111\Delta$

$\longrightarrow 01(q_2)10111\Delta$

$\longrightarrow 011(q_3)0111\Delta$

$\longrightarrow 0110(q_1)11\Delta$

$\longrightarrow 01101(q_2)1\Delta$

$\longrightarrow 011011(q_3)\Delta$

$\longrightarrow 011011(q_4)\Delta$ Accept

# TCS Assignment 6

Q1. Solve the following:

1. Explain concept of Decidability and undecidability in brief.

Ans: A problem is said to be decidable if there exists a Turing machine that gives the correct answer for every statement in the domain of the problem. Otherwise, the class of problems is said to be un-decidable

⇒ A language $L \leq \Sigma^*$ is turing decidable if there is a turing machine M which always halts on every $w \in \Sigma^*$. If $w \in L$ then M halts with answer "Yes" and if $w \notin L$ then M halts, with answer "No".

⇒ A problem P is said to be decidable/solvable if language $L \subseteq \Sigma^*$ representing the problem (set of solutions) is turing decidable.

⇒ Language is a set of solutions for any problem

⇒ If P is solvable/decidable then there is an algorithm for recognizing L, representing the problem.

⇒ The algorithm terminates on all inputs

2. What is recursive and recursively enumerable languages?

Ans: There is a difference between recursive and recursively enumerable language

⇒ Following statements are equivalent:
- Language L is Turing acceptable
- language L is recursively enumerable

⇒ Following statements are equivalent:
- language L is Turing decidable
- Language L is recursive
- There is an algorithm for recognizing L

⇒ Every Turing decidable language is Turing acceptable

⇒ Every Turing acceptable language need not be Turing decidable

⇒ A language L is turing acceptable if there is a Turing machine M which halts on every $w \in L$ with an answer 'yes'. However, if $w \notin L$ then M may not halt

⇒ Every language $L \subseteq \Sigma^*$ may not be turing acceptable and hence not turing decidable. Thus we cannot design turing machine / algorithm which halts for every $w \in L$

⇒ A turing enumerable language can be enumerated by some turing machine.

⇒ To enumerate a language means to list elements one at a time

⇒ Hence Recursive languages are decided by Turing machine i.e. it will enter into final state for strings of language and rejecting states for strings which are not part of language. On the other hand, Recursively Enumerable languages or type-0 languages can be accepted or recognized by Turing machine i.e. it will enter into final state for strings of language and may or may not enter into rejecting state for strings not part of language. TM can loop forever for strings not part of language

3. Explain halting problem in detail

Ans The halting problem of Turing machine states:

⇒ Given a Turing machine M and input w to machine, determine if machine M will eventually halt when it is given input w.

⇒ Halting problem of Turing machine is unsolvable / undecidable

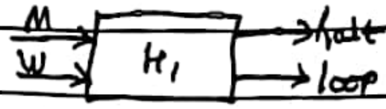
⇒ Proof:

- Moves of a turing machine can be represented using binary number. Thus machine can be represented using string over $\Sigma^* (0,1)$.

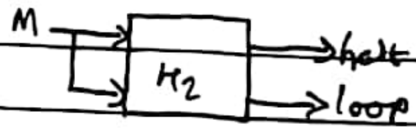
- Insolvability of halting problem can be proved via Contradiction

- Step 1: Assume halting problem is solvable. There exists machine $H_1$ taking two inputs:
  a. String describing M
  b. input w for machine M

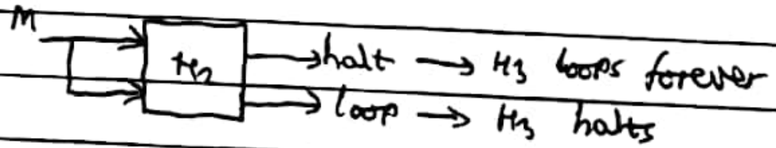
  $H_1$ Generates output "halt" if $H_1$ determines that M stops on input w otherwise H outputs "loop".




- Step 2: Revise $H_1$ as $H_2$ to take M as both inputs and $H_2$ should be able to determine if M will halt on M as its input. Machine can be described as string over 0 and 1.




- Step 3: Construct new Turing machine $H_3$ that takes output of $H_2$ as input and does following:
  a. If output of $H_2$ is "loop" then $H_3$ halts
  b. If output of $H_2$ is "halt" then $H_3$ [redacted] loops forever




  $H_3$ will do opposite of output of $H_2$

- Step 4: Give $H_3$ itself as inputs to $H_3$. If $H_3$ halts on $H_3$ as input, $H_3$ [redacted] loops. If $H_3$ loops forever on $H_3$ as input, $H_3$ halts. In either case, result is wrong. Hence, $H_3$ does not exist. If $H_3$ does not exist then $H_2$ does not exist. If $H_2$ does not exist then $H_1$ does not exist.

- ∴ Halting problem is unsolvable.

4. Write short note on Rice's theorem
Ans: "Every property that is satisfied by some but not all recursively enumerable language is undecidable"

⇒ Any property satisfied by some recursively enumerable langu but not all is known as nontrivial property.

⇒ Some properties of Recursively Enumerable that are undecidable are:

a. Given TM M, is L(M) non-empty?
b. Given TM M, is L(M) finite?
c. Given TM M, is L(M) regular?
d. Given TM M, is L(M) recursive?

⇒ Rice's theorem can be proved by reducing some other unsolvable problem to nontrivial property of recursively enumerable language

5. Explain Post correspondence problem with example
Ans: let A and B be two non-empty lists of strings of $\Sigma$. A and B are given as:

$$A = \{x_1, x_2, \dots x_k\}$$
$$B = \{y_1, y_2, \dots y_k\}$$

⇒ We say, there is post correspondence between A and B if there is a sequence of one or more integers i, j, ... m such that: string $x_i x_j \dots x_m$ is equal to $y_i y_j \dots y_m$

⇒ Example: Does PCP with 2 lists:

$$A = \{a, aba^3, ab\} \text{ and } B = \{a^3, ab, b\}$$

have a solution?

- We will have to find a sequence using which when elements of A, B are listed, will produce identical strings.
- The required sequence is $(\overline{2,1,1,3})$, $A_2A_1A_1A_3 = aba^3aaab = aba^6b$, $B_2B_1B_1B_3 = aba^3a^3b = \overline{aba^6b}$ $\therefore$ PCP has solution.

⇒ Post Correspondence Problem (PCP) is undecidable