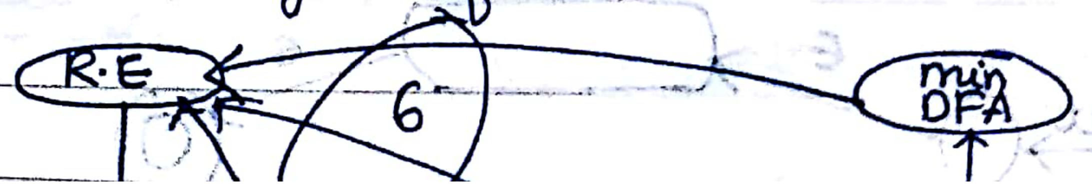
DFA

In DFA from each state each I/P symbol there is exactly one transition	In NFA from each state on each I/P symbol there can be zero one or more transitions
In DFA the transition function is defined as $\delta : Q \times \Sigma \rightarrow Q$	In NFA the transition function is defined as $\delta - Q \times \Sigma \rightarrow 2^Q$
The implementation of DFA with the help of computer programm is simple	The implementation of NFA with the help of a computer programm is difficult because of its non deterministic nature
In DFA there cannot be ϵ transition	In NFA there can be ϵ transition
eg of DFA	eg of NFA

Designs of DFA & NFA



$S \rightarrow C_1 A / C_2 A$

$A \rightarrow a / C_1 S / C_2 C_3$

$B \rightarrow b / C_2 C_4 / C_1 C_5$

$C_1 \rightarrow a$

$C_2 \rightarrow b$

$C_3 \rightarrow AA$

$C_4 \rightarrow SA$

$C_5 \rightarrow B/C$

$C_6 \rightarrow AB$

June 2010/ 10 marks

$A \rightarrow aBa / bBb$

$B \rightarrow aB / bB / \epsilon$

on Step 1

~~production~~ ~~solution~~

~~$A \rightarrow aBa$~~

Step 1 :- Elimination of null production

p | np

$\{B\}, \{\}$ $A \rightarrow aBa$ | $A \rightarrow aa$

$A \rightarrow bBb$ | $A \rightarrow bb$

$B \rightarrow aB$ | $B \rightarrow a$

$B \rightarrow \epsilon$ | -

p + np

$A \rightarrow aBa / aa / bBb / bb$

$B \rightarrow aB / a / bB / b / \epsilon$

after deleting null productions

$A \rightarrow aBa \mid aa \mid bBb \mid bb$

$B \rightarrow aB \mid a \mid bB \mid b$

<u>S2 & S3</u>

production	solution
$B \rightarrow a$	$B \rightarrow a$
$B \rightarrow b$	$B \rightarrow b$
$A \rightarrow aBa$	$C_1 \rightarrow a$
$A \rightarrow C_1BC_1$	$\boxed{A \rightarrow C_1C_2}$
$C_2 \rightarrow BC_1$	$C_2 \rightarrow BC_1$
$A \rightarrow aa$	
$A \rightarrow C_1C_1$	$\boxed{A \rightarrow C_1C_1}$
$A \rightarrow bBb$	$C_3 \rightarrow b$
$A \rightarrow C_3BC_3$	$\boxed{A \rightarrow C_3C_4}$
$C_4 \rightarrow BC_3$	$C_4 \rightarrow BC_3$
$A \cancel{B} \rightarrow bb$	
$A \cancel{B} \rightarrow C_3C_3$	$\boxed{A \cancel{B} \rightarrow C_3C_3}$
$B \rightarrow aB$	
$B \rightarrow C_1B$	$B \rightarrow C_1B$
$B \rightarrow bB$	
$B \rightarrow C_3B$	$B \rightarrow C_3B$

$A \rightarrow C_1C_2 / C_1C_1 / C_3C_4 / C_2C_2$

$B \rightarrow C_1B / a / C_2B / b$

② $S \to \sim S$
$S \to [S \supset S]$
$S \to p$
$S \to q$

S_1 (done)

S_2 & S_3

production	solution
$S \to p$	$S \to p$
$S \to q$	$S \to q$
$S \to \sim S$	$C_1 \to \sim$
$S \to C_1 S$	$S \to C_1 S$
$S \to [S \supset S]$	$C_2 \to [$, $C_3 \to \supset$, $C_4]$
$S \to C_2 S C_3 S C_4$	$S \to C_2 \cancel{S C_3 S C_4} C_5$
$S \to C_2 C_5$	
$C_5 \to S C_3 S C_4$	$C_5 \to S C_6$
$C_6 \to C_3 S C_4$	$C_6 \to C_3 C_7$
$C_7 \to S C_4$	$C_7 \to S C_4$

Normal forms

* [Greibach normal form (GNF)

Definition: any ε-free CFL can be generated by CFG in which all the productions are of the following

$A \to a\alpha$

γ = string of variables
(can be empty)

Such a CFG is said to be in $\boxed{\text{GNF}}$

Conversion procedure from CFG to GNF

Step1 perform elimination of null, unit & useless production

Step2 use any combination of Rule 1 & Rule 2 to get CFG in GNF

Rule1 let $A \rightarrow B\alpha$ be some A-production & let $B \rightarrow \beta_1 | \beta_2 | \cdots | \beta_n$ be B product then we can write A-produce as-

$$A \rightarrow \beta_1\alpha | \beta_2\alpha | \cdots | \beta_n\alpha$$

Rule2 Let $A \rightarrow A\alpha_1 / A\alpha_2 | \cdots | A\alpha_5$ be some A-productions (starting with A) & let $A \rightarrow \beta_1 | \beta_2 | \cdots | \beta_3$ be the remaining A-production (not starting with A) then

Express CFG in GNF

(1)

$S \rightarrow AB \mid BA \mid AA$

$A \rightarrow aA \mid b$

$B \rightarrow bB \mid a$

$B \rightarrow bB \mid a$

$A \rightarrow aA \mid b$

$S \rightarrow AB$ $\quad S \rightarrow aAB \mid bA$ (R1)

$S \rightarrow BA$ $\quad S \rightarrow bBA \mid aA$ (R1)

$S \rightarrow AA$ $\quad S \rightarrow aAA \mid bA$ (R1)

(2) $A_1 \rightarrow A_2A_2 \mid a$

$A_2 \rightarrow A_1A_1 \mid b$

$A_2 \rightarrow A_1A_1 \mid b$

$A_2 \rightarrow A_2A_2A_1 \mid aA_1 \mid b$ (R1)

$A \rightarrow A\overline{\alpha_1} \quad \overline{\beta_1} \quad \beta_2$

assumption

$B \rightarrow \underset{\alpha_2}{A_2A_1} \mid \underset{\alpha_1}{A_2A_1B} \leftarrow \alpha_1 \mid \alpha_1 B$

$A_2 \rightarrow aA_1 \mid aA_1B \mid b \mid bB \quad \beta_1 \mid \beta_1 B$

$A_1 \rightarrow aA_1A_2 \mid aA_1BA_2 \mid bA_2 \mid bBA_2 \mid a$ (R1)

$B \rightarrow aA_1A_1 \mid aA_1BA_1 \mid bA_1 \mid bBA_1$

$aA_1A_1B \mid aA_1BA_1B \mid bBA_1B$

ii) $S \to AA / 0$ — express the following in CNF & GNF

$A \to SS / 1$

$A \to SS / 1$

$A \to AAS / 0S / 1$ (R1)

$A \to A\alpha_1 \;\; \beta_1 \;\; \beta_2$

$\to \alpha_i / \alpha_i B$

$\to \beta_i / \beta_i B$

$B \to AS / ASB$

$A \to 0S / 0SB / 1 / 1B$ (R2)

$S \to 0SA / 0SBA / 1A / 1BA / 0$ ✓ (R1)

$B \to 0SS / 0SBS / 1S / 1BS /$

$0SSB / 0SBSB / 1SB / 1BSB$ ✓ (R1)

express CFG in GNF $A \to a\gamma$

$S \to asa / bsb / \epsilon$

$S \to aSC_1 / bSC_2 / \epsilon$

$C_1 \to a$

$C_2 \to b$

$S \to SS / asb / ab$

$S \to \underline{SS} / \underline{asb} / \underline{ab}$

$A \to A\alpha_1 \;\; \beta_1 \;\; \beta_2$

$/ \alpha_i B$

$/ \beta_i B$

$B \to S / SB$

$S \to asb / aSbB / ab / abB$ (R2)

$S \to aSC_1 / aSC_1B / aC_1 / aC_1B$ ✓ (S.R)

$C_1 \to b$

Scanned by CamScanner

Express the following language in CNF & GNF

$L = \{a^n b^n\}$ $n \geq 0$

CFG $S \rightarrow aSb / \epsilon$

Simplified CFG $S \rightarrow aSb / ab$

$A \rightarrow BC$

$A \rightarrow a$

$A \rightarrow a\gamma$

CNF

$C_1 \rightarrow a$

$C_2 \rightarrow b$

$S \rightarrow C_1C_2$

$S \rightarrow C_1C_3$

$C_3 \rightarrow SC_2$

GNF

$S \rightarrow aSC_1$

$S \rightarrow aC_1$

$C_1 \rightarrow b$

Chap 7. Push down automata (PDA)

PDA is used for recognizing context free language which is generated by context free grammar

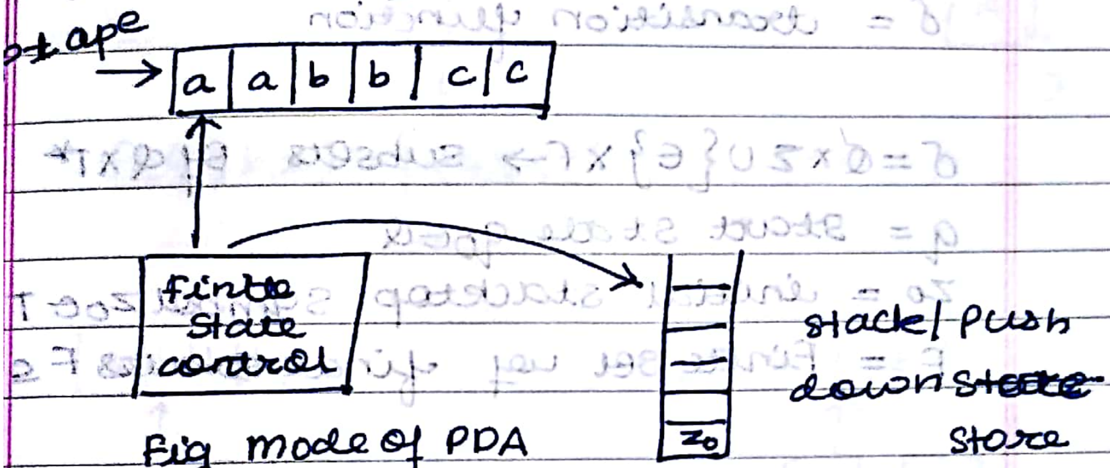


Fig model of PDA

Components of PDA

PDA consists of finite set of states, I/P tape, read head & a stack

working of PDA

Depending on the state, I/P symbol & stack top symbol

PDA can change the state / remain in same state

PDA moves head to right of current cell

PDA can perform some stack operation

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

(PDA)

where

Q = finite set of states

Σ = I/P alphabet

Γ = stack alphabet

δ = transition function

$\delta = Q \times \Sigma \cup \{\epsilon\} \times \Gamma \rightarrow$ subsets of $Q \times \Gamma^*$

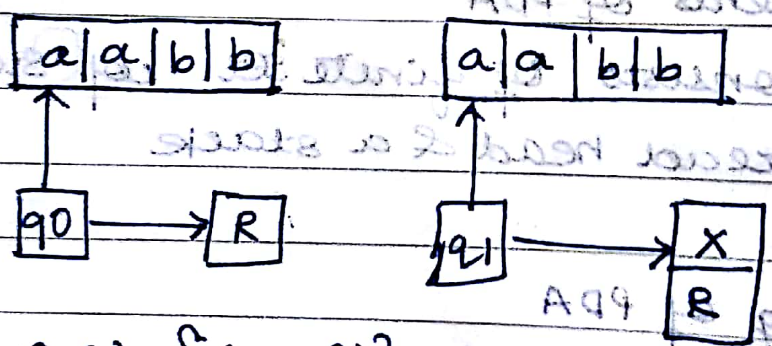
q = start state $q_0 \in Q$

Z_0 = initial stacktop symbol $Z_0 \in \Gamma$

F = Finite set of final states $F \subseteq$

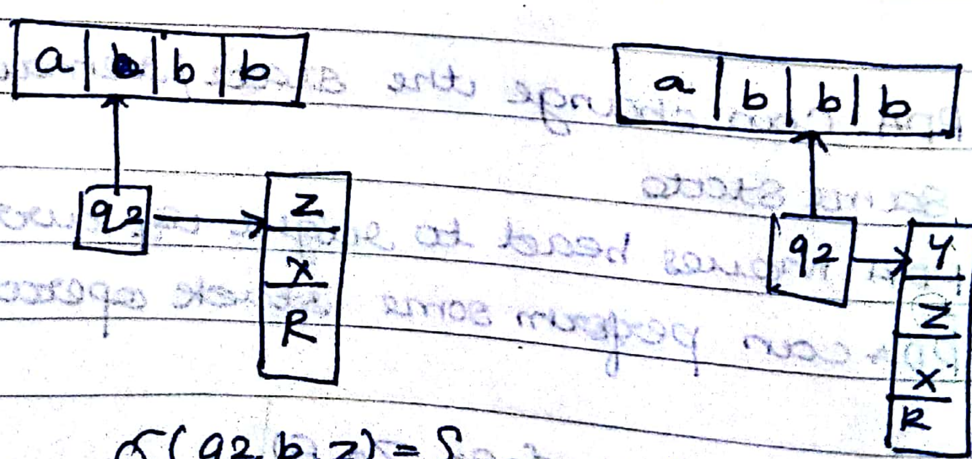
Some eg of transition function

①

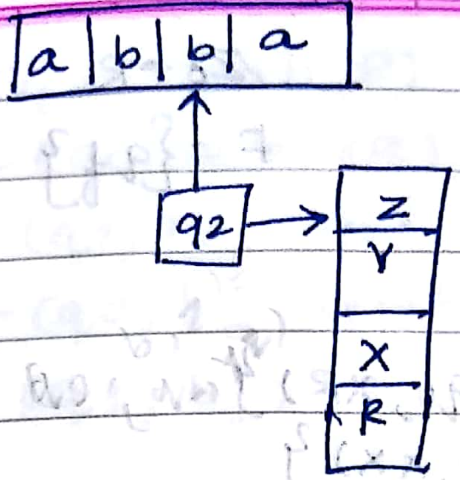


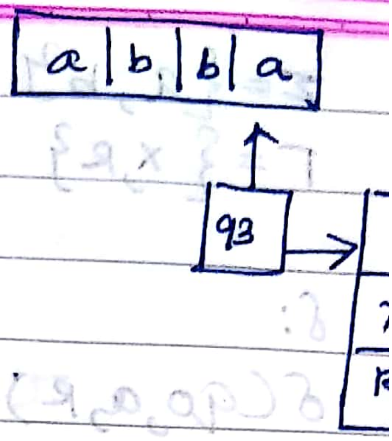
$\delta(q_0, a, R) = \{(q_1, XR)\}$

②

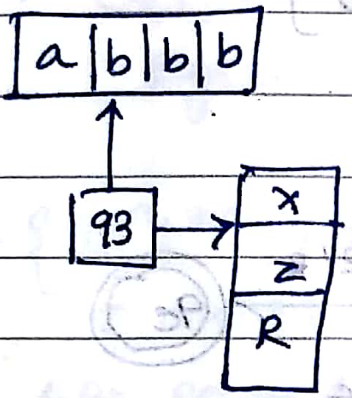


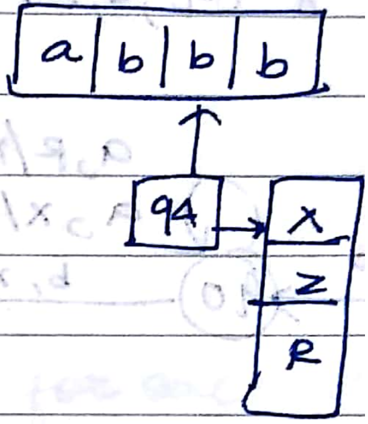
$\delta(q_2, b, Z) = \{(q_2, YZ)\}$





$\delta(q_2, b, Z) = \{(q_3, \epsilon)\}$





$\delta(q_3, b, X) = \{(q_4, X)\}$

Design a PDA for recognising
$L = \{a^n b^n \mid n \geq 1\}$

on **Step 1** Theory

Step 2 Logic
For each 'a' push 1X for each 'b

Step 3 Implementation
$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$\Sigma = \{a, b\}$ $\qquad\qquad$ $Z_0 = R$

$\Gamma = \{x, R\}$ $\qquad\qquad$ $F = \{q_f\}$

δ:

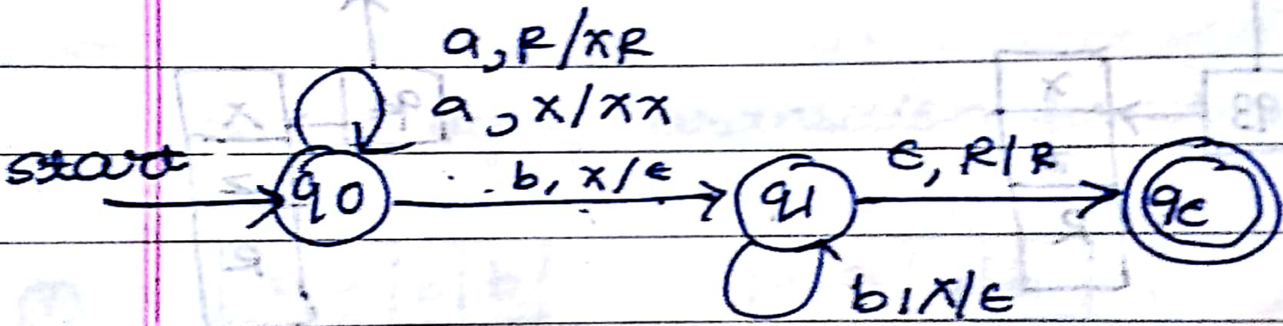
$\delta(q_0, a, R) = \{(q_0, xR)\}$

$\delta(q_0, a, x) = \{(q_0, xx)\}$

$\delta(q_0, b, x) = \{(q_1, \epsilon)\}$

$\delta(q_1, b, x) = \{(q_1, \epsilon)\}$

$\delta(q_1, \epsilon, R) = \{(q_f, R)\}$



Steps

$(q_0, aaabbb, R)$ ↓stack

$\vdash (q_0, aabbb, xR)$

$\vdash (q_0, abbb, xxR)$

$\vdash (q_0, bbb, xxxR)$

$\vdash (q_1, bb, xxR)$

$\vdash (q_1, \epsilon, R)$

$\vdash (q_f, \epsilon, R)$

accept

$\vdash (q_0, ab, xxR)$

$\vdash (q_0, b, xxxR)$

$\vdash (q_1, \epsilon, xxR)$ Reject

$(q_0, abab, R)$

$\vdash (q_0, bab, xR)$

$\vdash (q_1, ab, R)$

Reject

$L = \{0^n 1^{2n} / n \geq 1\}$

000111111

for each '0' push 2x for each '1' pop 1x

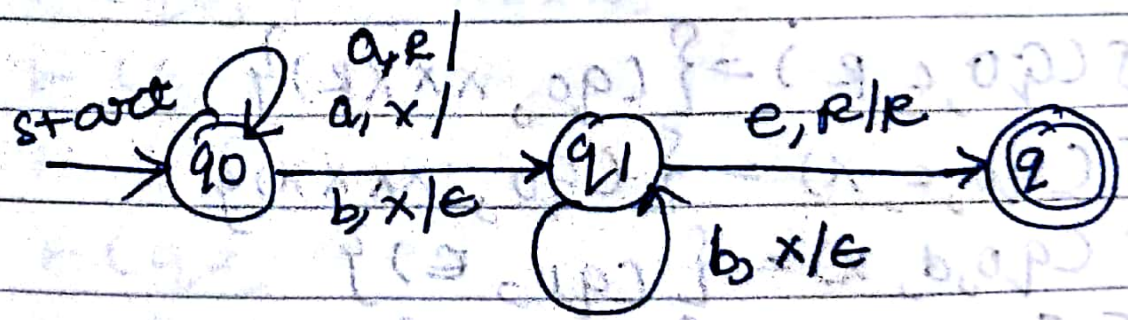
$\delta(q_0, 0, R) = \{(q_0, xxR)\}$

$\delta(q_0, 0, x) = \{(q_0, xxx)\}$

$\delta(q_0, 1, x) = \{(q_1, \epsilon)\}$

$\delta(q_1, 1, x) = \{(q_1, \epsilon)\}$

$\delta(q_1, \epsilon, R) = \{(q_f, R)$



$(q_0, 000111111, R)$

$\vdash (q_0, 00111111, XXR)$

$\vdash (q_0, 0111111, XXXXR)$

$\vdash (q_0, 111111, XXXXXXR)$

$\vdash (q_1, 11111, XXXXXR)$

$\vdash (q_1, 1111, XXXXR)$

$\vdash (q_2, 111, XXXR)$

$\vdash (q_1, 11, XXR)$

$\vdash (q_1, 1, XR)$

$\vdash (q_1, 1, XR)$

$\vdash (q_1, 1, XR)$

$\vdash (q_1, \epsilon, R)$

$\vdash (q_f, f, R)$ accept

(P) $L = \{c^n j^{3n} \mid n \geq 1\}$

Solution   logic:-

δ:

$\delta(q_0, c, R) = \{(q_0, XXXR)\}$

$\delta(q_0, c, X) = \{(q_0, XXXX)\}$

$\delta(q_0, d, X) = \{(q_1, \epsilon)\}$

$\delta(q_1, d, X) = \{(q_1, \epsilon)\}$

$\delta(q_1, \epsilon, R) = \{(q_f, R)\}$

$L = \{ (ab)^n c^n \mid n \geq 1 \}$ (2)

ababab ccc

For each 'ab' push 1 x

For each 'c' pop 1 x

$\delta(q0, a, R) = \{(q1, R)\}$
$\delta(q1, b, R) = \{(q0, xR)\}$
$\delta(q0, a, x) = \{(q1, x)\}$
$\delta(q1, b, x) = \{(q0, xx)\}$
$\delta(q0, c, x) = \{(q2, \epsilon)\}$
$\delta(q2, c, x) = \{(q2, \epsilon)\}$
$\delta(q2, \epsilon, R) = \{(q_f, R)\}$

$(q0, \text{abababccc}\ R)$
$\vdash (q1, \text{bababccc}, R)$
$\vdash (q0, \text{ababccc}, xR)$
$\vdash (q1, \text{babccc}, xR)$
$\vdash (q0, \text{abccc}, xxR)$
$\vdash (q1, \text{bccc}, xxR)$
$\vdash (q0, \text{ccc}, xxxR)$
$\vdash (q2, \text{cc}, xxR)$
$\vdash (q2, \text{c}, xR)$
$\vdash (q2, \epsilon, R)$

$L = \{ (213)^n 4^n \mid n \geq 1 \}$

$\delta(q_0, 2, R) = \{(q_1, R)\}$
$\delta(q_1, 1, R) = \{(q_2, R)\}$
$\delta(q_2, 3, R) = \{(q_0, xR)\}$

$\delta(q_0, 2, x) = \{(q_1, x)\}$
$\delta(q_1, 1, x) = \{(q_2, x)\}$
$\delta(q_2, 3, x) = \{q_0, xx)\}$
$\delta(q_0, 4, x) = \{(q_3, \epsilon)\}$
$\delta(q_3, 4, x) = \{(q_3, \epsilon)\}$
$\delta(q_3, \epsilon, R) = \{(q_f, R)\}$

$L = \{ (bdac)^n e^n \mid n \geq 1 \}$

$\delta(q0, a, R) = \{(q0, xR)\}$

$\delta(q0, a, x) = \{q0, xx)\}$

$\delta(q0, b, x) = \{(q1, x)\}$

$\delta(q1, c, x) = \{(q2, x)\}$

$\delta(q2, d, x) = \{(q3, \epsilon)\}$

$\delta(q3, b, x) = \{(q1, x)\}$

$\delta(q3, \epsilon, R) = \{(qf, R)\}$

$L = \{0^n (2143)^n \mid n \geq 1\}$

$\delta(q0, 0, R) = \{(q0, xR)\}$

$\delta(q0, 0, x) = \{(q0, xx)\}$

$\delta(q0, 2, x) = \{(q1, x)\}$

$\delta(q1, 1, x) = \{(q2, x)\}$

$\delta(q2, 4, x) = \{(q3, x)\}$

$\delta(q3, 3, x) = \{(q4, \epsilon)\}$

$\delta(q4, 2, x) = \{(q1, x)\}$

$\delta(q4, \epsilon, R) = \{(qf, R)\}$

3) $L = \{ (ab)^n (cdef)^n \mid n \geq 1 \}$

$\delta(q_0, a, R) = \{(q_1, R)\}$
$\delta(q_1, b, R) = \{q_0, xR\}$

$\delta(q_0, a, x) = \{(q_1, x)\}$
$\delta(q_1, b, x) = \{q_0, xx\}$

$\delta(q_0, c, x) = \{(q_2, x)\}$
$\delta(q_2, d, x) = \{(q_3, x)\}$
$\delta(q_3, e, x) = \{(q_4, x)\}$
$\delta(q_4, f, x) = \{(q_5, \epsilon)\}$
$\delta(q_5, c, x) = \{(q_2, x)\}$
$\delta(q_5, \epsilon, R) = \{(q_f, R)\}$

PDA design methods.

1) PDA by final state method

$(q_0, w, Z) \vdash^* (q, \epsilon, \alpha) \quad q \in F, \ \alpha \in \Gamma^*$

2) PDA by Null / Empty state method

$(q_0, w, Z) \vdash^* (q, \epsilon, \epsilon) \quad q \in Q \quad F = \{$

1. $L = \{ a^n b^{n+1} / n \geq 1 \}$

For each 'a' push 1x
Bypass 1st b
For each 'b' pop 1x

$\delta(q_0, a, R) = \{(q_0, xR)\}$
$\delta(q_0, a, x) = \{(q_0, xx)\}$
$\delta(q_0, b, x) = \{(q_1, x)\}$
$\delta(q_1, b, x) = \{(q_1, \epsilon)\}$
$\delta(q_1, \epsilon, R) = \{(q_f, R)\}$ = FSM
$= \{(q_1, \epsilon)\}$ = NSM

2. $L = \{ a^n b^m a^n / m, n \geq 1 \}$

aaa bbbbbbbbaaa

For each 'a' push 1x Bypass all b's
for each 'a' pop 1x

$\delta(q_0, a, R) = \{(q_0, xR)\}$
$\delta(q_0, a, x) = \{(q_0, xx)\}$
$\delta(q_0, b, x) = \{(q_1, x)\}$
$\delta(q_1, b, x) = \{(q_1, x)\}$
$\delta(q_1, a, x) = \{(q_2, \epsilon)\}$
$\delta(q_2, a, x) = \{(q_2, \epsilon)\}$
$\delta(q_2, \epsilon, R) = \{(q_2, \epsilon)\}$

(3) $L = \{ a^n b^m c^m d^n \mid m, n \geq 1 \}$

logic :- $a \rightarrow$ push 1x, $d \rightarrow$ Pop 1x | $b \rightarrow$ pu | $c \rightarrow$ Po

Solution

$\delta(q0, a, R) = \{q0, xR\}$
$\delta(q0, a, x) = \{q0, xx\}$

$\delta(q0, b, \cancel{X}) = \{q1, \cancel{y}x\}$
$\delta(q1, b, Y) = \{q1, YY\}$

$\delta(q1, c, y)\} = \{q2, \epsilon\}$
$\delta(q2, c, y)\} = \{(q2, \epsilon)\}$

$\delta(q2, d, x)\} = (q3, \epsilon)$
$\delta(q3, d, x) = (q3, \epsilon)$
$\delta(q3, \epsilon, x) = \{(q3, \epsilon)\}$

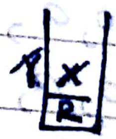
(4) $L = \{ x \mid na(x) = nb(x) \}$

Logic

ab

ba

abba



CONCLUSION

a push

$\delta(q_0, a, R) = \{(q_0, xR)\}$

$\delta(q_0, b, R) = \{(q_0, YR)\}$

$\delta(q_0, a, x) = \{(q_0, xx)\}$

$\delta(q_0, b, y) = \{(q_0, yy)\}$

$\delta(q_0, a, y) = \{(q_0, \epsilon)\}$

$\delta(q_0, b, x) = \{(q_0, \epsilon)\}$

$\delta(q_0, \epsilon, R) = \{(q_0, \epsilon)\}$

Design PDA ~~to~~ check for well formedness of parenthesis

(( ) (( )))

For each '(' push 1 x

for each ')' pop 1 x

δ:-

$\delta(q_0, (, R) = \{(q_0, xR)\}$

$\delta(q_0, (, x) = \{(q_0, xx)\}$

$\delta(q_0,), x) = \{(q_0, \epsilon)\}$

$\delta(q_0, \epsilon, R) = \{(q_0, \epsilon)\}$

★ $L = \{ wcw^R \mid w \in \{a,b\}^*,\ w^R\text{ - Reverse of } w \}$ odd palin

abaabcbaaba

Scanned by CamScanner

$\delta(q_1, \epsilon, R) = \{(q_1, \epsilon)\}$

$L = \{ ww^R \mid w \in (a,b)^*, \; w^R \text{ reverse of } w \}$:-

This is a case of NPDA design where NPDA stands for non deterministic PDA

whenever there is a double letter two cases are possible

The middle of the string is not reached & hence we continue to push the respective symbols on to the stack middle of the string is reached & hence we start to pop the respective symbols from the stack.

$\delta(q0, \epsilon, R) = \{(q0, \epsilon)\}$

$\delta(q0, a, R) = \{(q0, xR)\}$

$\delta(q0, b, R) = \{(q0, YR)\}$

$\delta(q0, a, x) = \{(q0, xx), (q1, \epsilon)\}$

$\delta(q0, b, x) = \{(q0, Yx)\}$

$\delta(q0, a, Y) = \{(q0, xY)\}$

$\delta(q0, b, Y) = \{(q0, Y\cancel{x}), (q1, \epsilon)\}$

$\delta(q1, a, x) = \{(q1, \epsilon)\}$

$\delta(q2, b, Y) = \{(q1, \epsilon)\}$

$\delta(q1, \epsilon, R) = \{(q1, \epsilon)\}$

$(q0, abba, R)$

$\vdash (q0, bba, xR)$

Define PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ for recognizing CFL generated by CFG G

S1 Express CFG in GNF $(A \to a\alpha)$

S2 PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

1. $Q = \{q_0\}$
2. $\Sigma = T'$
3. $T = V'$
4. $q_0 = q_0$
5. $z_0 = S$
6. $F = \{ \}$
7. δ :-

* $A \to aBCD$

$\delta(q_0, a, A) = \{(q_0, BCD)\}$

S1 $S \rightarrow aSc_1 \mid bSc_2 \mid c$

$c_1 \rightarrow a$

$c_2 \rightarrow b$

S.R selective replacement

$m : (Q, \Sigma, \Gamma, q, Z, \delta)$

S2 PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

$Q = \{q_0\}$

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

$\Gamma = \{S, c_1, c_2\}$

$q_0 = q_0$

$z_0 = S$

$F = \{\}$

s i/p ST

$S \rightarrow aSc_1$	$\delta(q_0, a, S) = \{(q_0, Sc_1)\}$
$S \rightarrow bSc_2$	$\delta(q_0, b, S) = \{(q_0, Sc_2)\}$
$S \rightarrow c$	$\delta(q_0, c, S) = \{(q_0, \epsilon)\}$
$c_1 \rightarrow a$	$\delta(q_0, a, c_1) = \{(q_0, \epsilon)\}$
$c_2 \rightarrow b$	$\delta(q_0, b, c_2) = \{(q_0, \epsilon)\}$

Design PDA for

$S \rightarrow 0BB$

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

and test whether 010^4 belongs to the language

on S1 :- (done)

S2 :- PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

$Q = \{q_0\}$

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

$\Gamma = \{S, B\}$

$q_0 = q_0$

$z_0 = S$

$F = \{\}$

δ :-

$S \rightarrow 0BB$	$\delta(q_0, 0, S) = \{(q_0, BB)\}$
$B \rightarrow 0S$	$\delta(q_0, 0, B) = \{(q_0, S)\}$
$B \rightarrow 1S$	$\delta((q_0, 1, B) = \{(q_0, S)\}$
$B \rightarrow 0$	$\delta(q_0, 0, B) = \{(q_0, \epsilon)\}$

$(q_0, 010000, S)$

$\vdash (q_0, 10000, BB)$

$\vdash (q_0, 0000, SB)$

$\vdash (q_0, 000, BBB)$

$\vdash (q_0, 00, BB)$ $\vdash (q_0, 00, SBB)$

$\vdash (q_0, 0, B)$

* $\delta(q, a, A) = \{(p, BC)\}$

$[q, A, r] \rightarrow a\,[p, B, s]\,[s, C, r]$, $r, s \in Q$

PDA

$m = \{ (q_0, q_1), \{a, b\}, \{z_0, z\}, \delta, q_0, z_0, \phi)$

A

$\delta(q_0, a, z_0) = \{ (q_0, z z_0) \}$

$\delta(q_0, a, z) = \{ (q_0, zz) \}$

$\delta(q_0, b, z) = \{ (q_1, z) \}$

$\delta(q_1, b, z) = \{ (q_1, z) \}$

$\delta(q_1, a, z) = \{ (q_1, \epsilon) \}$

$\delta(q_1, \epsilon, z_0) = \{ (q_1, \epsilon) \}$

Solution

$G = (V, T, P, S)$

$V = \{ S, \overset{A}{[q_0, z_0, q_0]}, \overset{B}{[q_0, z_0, q_1]},$

$C\ [q_1, z_0, q_0], [q_1, z_0, q_1]\ B$

$E\ [q_0, z, q_0], [q_0, z, q_1]\ F$

$G\ [q_1, z, q_0], [q_1, z, q_1]\ H \}$

$T = \{a, b\}$

$S = S$

$\delta(q_0, a, z_0) = \{(q_0, zz_0)\}$

$[q_0, z_0, q_0] \rightarrow a\,[q_0, z, q_0][q_0, z_0, q_0]$

$[q_0, z_0, q_0] \rightarrow a\,[q_0, z, q_1][q_1, z_0, q_0]$

$[q_0, z_0, q_1] \rightarrow a\,[q_0, z, q_0][q_0, z_0, q_1]$

$[q_0, z_0, q_1] \rightarrow a\,[q_0, z, q_1][q_1, z_0, q_1]$

$\delta(q_0, a, z) = \{(q_0, zz)\}$

$[q_0, z, q_0] \rightarrow a\,[q_0, z, q_0][q_0, z, q_0]$

$[q_0, z, q_0] \rightarrow a\,[q_0, z, q_1][q_1, z, q_0]$

$[q_0, z, q_1] \rightarrow a\,[q_0, z, q_0][q_0, z, q_1]$

$[q_0, z, q_1] \rightarrow a\,[q_0, z, q_1][q_1, z, q_1]$

$\delta(q_0, b, z) = \{(q_1, z)\}$

$[q_0, z, q_0] \rightarrow b\,[q_1, z, q_0]$

$[q_0, z, q_1] \rightarrow b\,[q_1, z, q_1]$

$\delta(q_1, b, z) = \{(q_1, z)\}$

Turing machine is considered to be a simple model of a computer & is the most powerful machine

Tm can perform the following
language recognition
language generation
computation of some functions

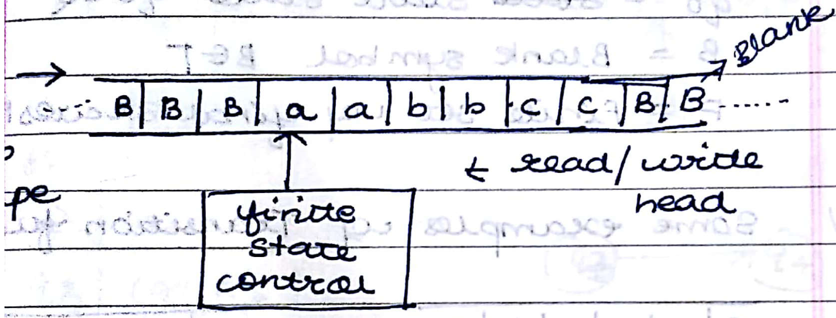


Fig model of Tm

Components of Tm
Tm consists of finite set of states, I/P, O/P tape & read/write head.

working of Tm
Depending on the state of tape symbol:-
Tm can change the tape symbol or keep it the same
T.m can move the ~~bend~~ head either to {L,R,S}
stay

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

where

Q = Finite set of states

Σ = I/P alphabet

Γ = tape alphabet

δ = transition function

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$$

q_0 = ~~State~~ start state $q_0 \in Q$

B = Blank symbol $B \in \Gamma$

F = Finite set of final states $F \subseteq Q$

Some examples of Transition function

(1)

B	a	a	b	b	B

↑ $\boxed{q_0}$ (under the first a)

B	x	a	b	b	B

↑ $\boxed{q_1}$ (under the second a)

state $Q \backslash \Gamma$	a (tape symbol)
q_0	(q_1, x, R)

$q_0 \xrightarrow{a/xR} q_1$ make a x & move right

(2)

B	a	b	b	b	B

↑ $\boxed{q_2}$ (under the first b)

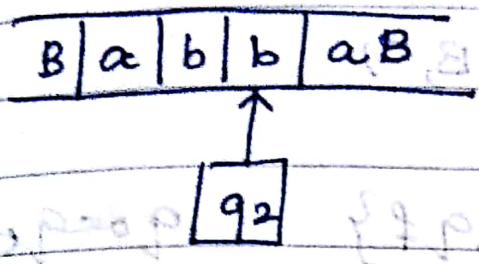
B	a	b	b	b	B

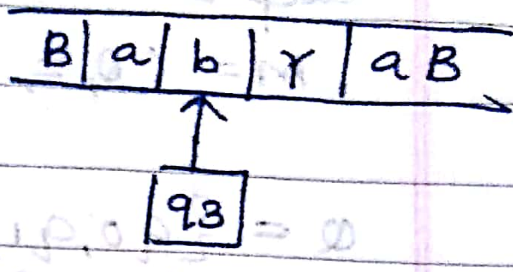
↑ $\boxed{q_2}$ (under the second b)

bypass

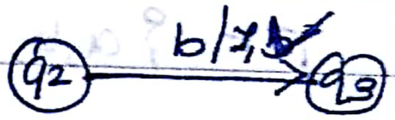
$Q \backslash \Sigma$	b
q_2	(q_2, b, R)

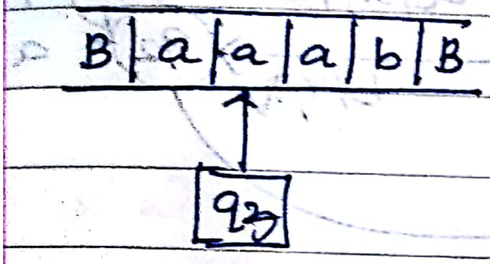
$q_2 \xrightarrow{b/b,R} q_2$

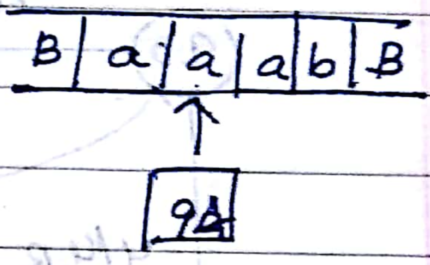




Q \ Γ	b
q2	(q3, Y, ~~b~~)

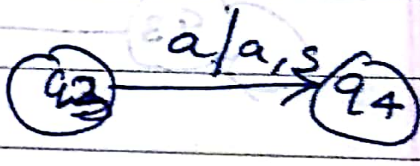






δ:-

Q \	a
q3	(q4, a, S)



Design TM for recognizing
$L = \{ a^n b^n \mid n \geq 1 \}$

on **Step 1** Theory

Step 2 Logic

q0 – a make it X

q1 – b make it Y

q2 – comeback

q3 – check

q4 – final state

Scanned by CamScanner

'a' make 'x' 'b' make it 'y' come back.

~~a~~ x a a a ~~b~~ y b b b B → final state

Step 3

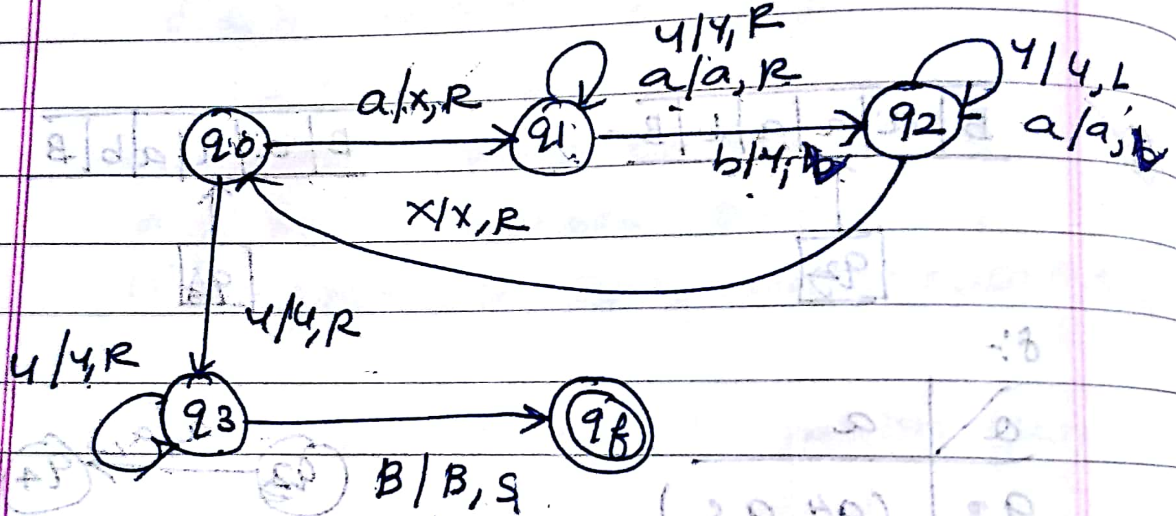
$m = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$

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

$\quad B = B$

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

$\Gamma = \{a, b, X, Y, B\}$ $\quad F = \{q_f\}$



Q \ Γ	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)	(qf, B, S)
* qf					

Step 1

①

B	a	a	a	b	b	b	B

↑ q_0

②

B	X	a	a	b	b	b	B

↑ ↑ ↑ q_1

③

B	X	a	a	Y	b	b	B

↑ ↑ ↑ q_2

④

B	X	a	a	Y	b	b	B

↑ q_0

⑤

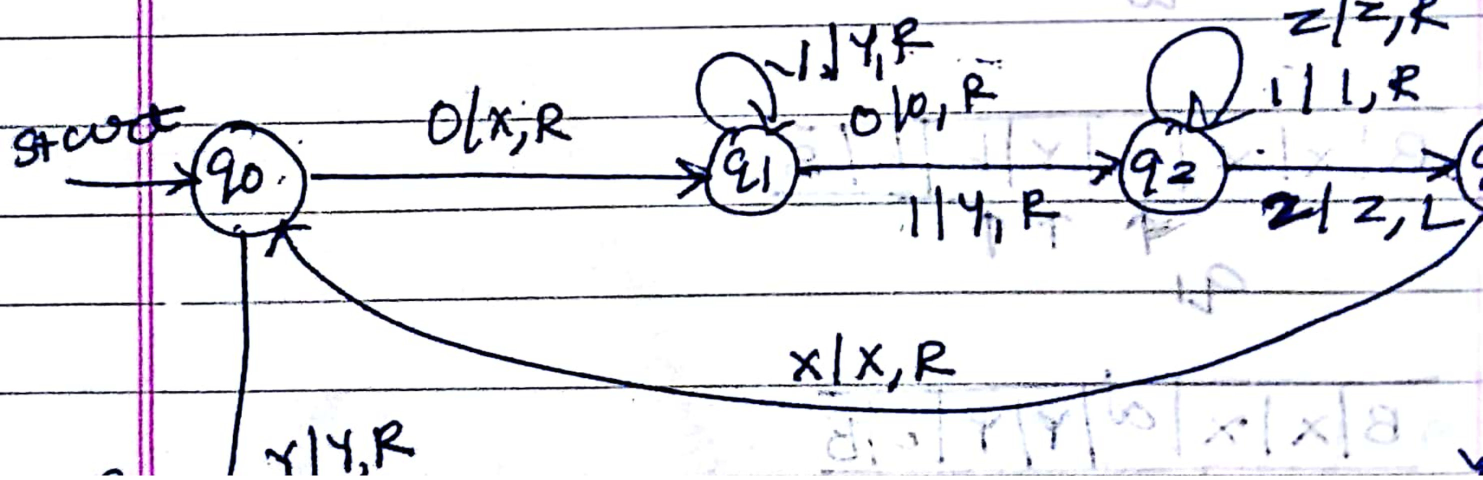
B	X	X	a	Y	b	b	B

↑ ↑ ↑ q_1

Q. Design T.M for recognizing
$L = \{ 0^n 1^n 2^n \mid n \geq 1 \}$

B   XXX YYY ZZZ B
↑ qf

q_0 – 0 make it X
q_1 – 1 make it Y
q_2 – 2 make it Z
q_3 – comeback
q_4 – check
q_f – final state



Design TM for recognizing
$L = \{ x \mid n_a(x) = n_b(x) \}$

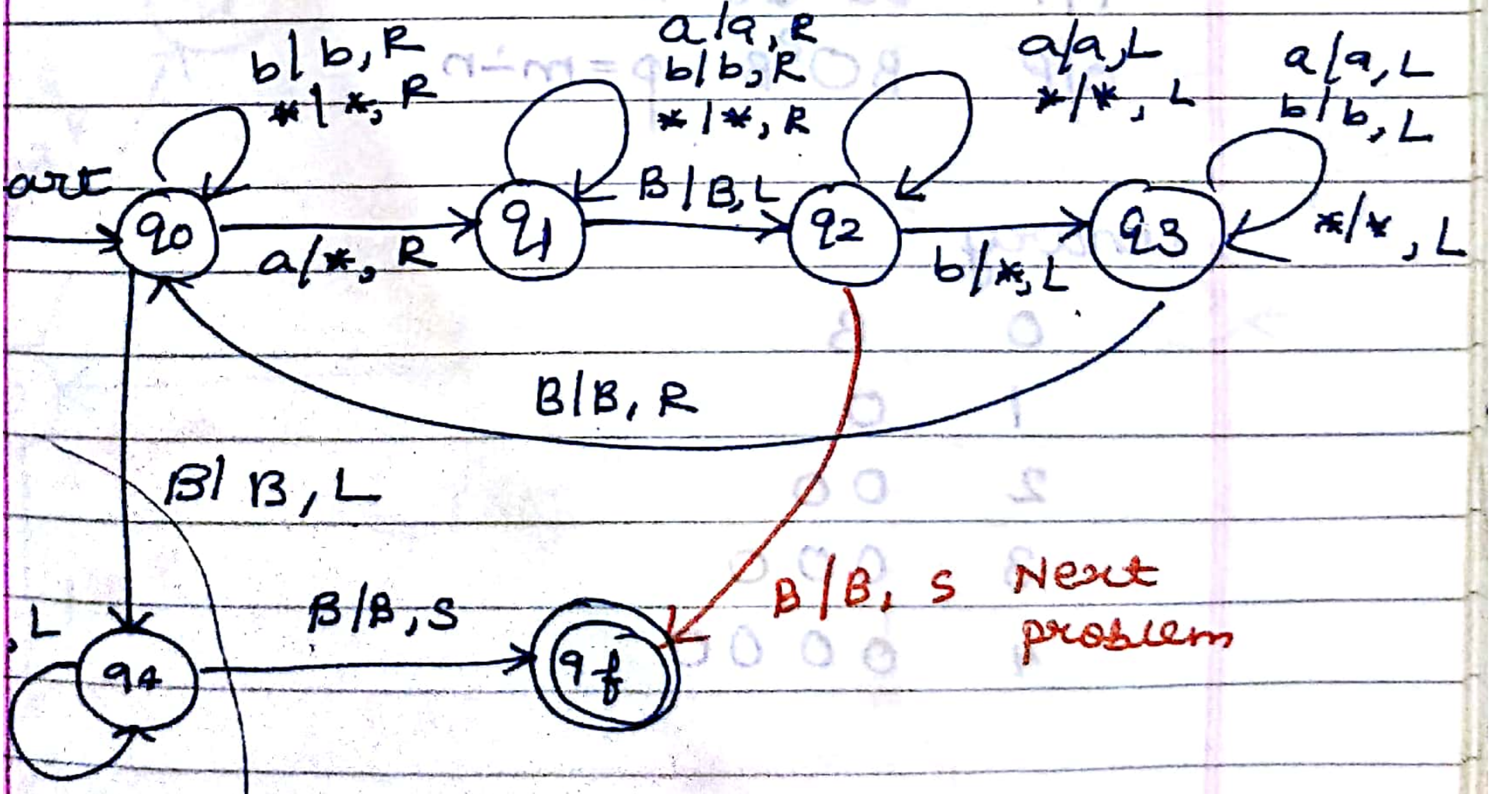
B b b a a b a B

B b b * a * a B

B b * * * * a B

B * * * * * * B

q_f q_0



q_0 - find 'a' make it *
q_1 - goto right

Regular Grammar

Defination - Grammar is said to be regular if all the productions are of the form

A → any number of T's and atmost

1V

LLG ← → RLG

RLG to F.A

Step 1 Draw the T.D with the number of states = number of V + 1 (final state)

Step 2

Production	Transition
$A \to aB$	$\delta(A, a) = B$
$A \to a$	$\delta(A, a) =$ final state

Step 3 If L(G) contains ∈, then make the start state also final.

Step 4 NFA → DFA → min DFA.

① Draw FA for the following

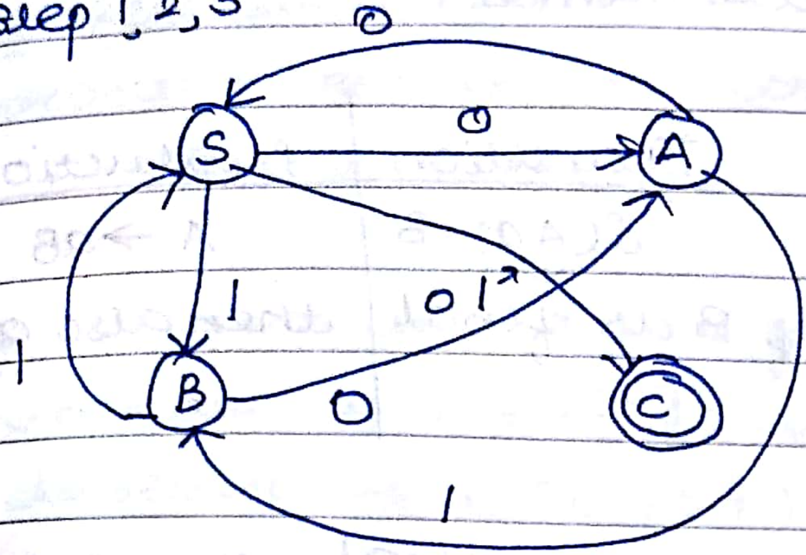
$S \to 0A \mid 1B \mid 0 \mid 1$

$A \to 0S \mid 1B$

$B \to 0A \mid 1S$

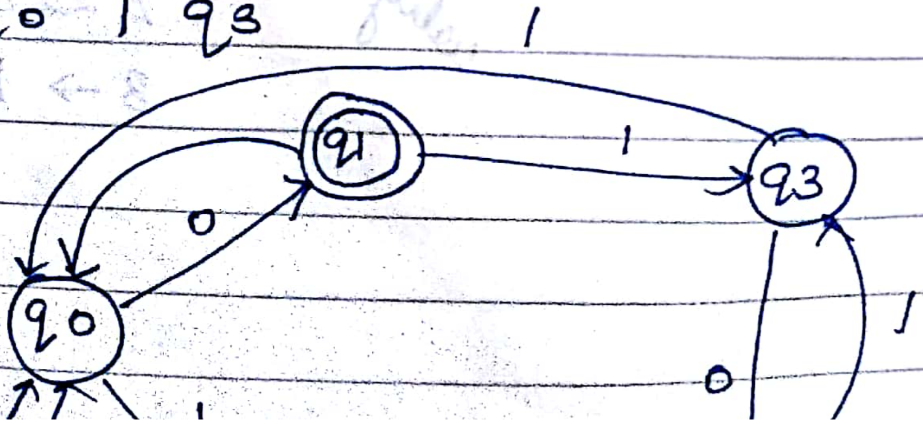
Solution

Step 1, 2, 3



x	y = Eclosure(x)		δ(y,0)	δ(y,1)
{S}	{S}	q0	{A,C} q1	{B,C} q2
{A,C}	{A,C}	q1*	{S} q0	{B} q3
{B,C}	{B,C}	q2*	{A} q4	{S} q0
{B}	{B}	q3	{A} q4	{S} q0
{A}	{A}	q4	{S} q0	{B} q3

Q \ Σ	0	1
→q0	q1	q2
q1*	q0	q3
q2*	q4	q0
q3	q4	q0
q4	q0	q3



PAGE NO.
DATE: / /

FA to RLG

Step1 Associate variables like S, A, B with state names

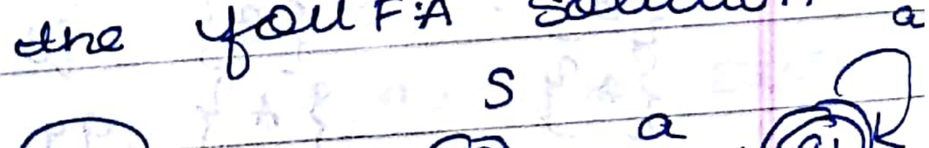
Step2

Transition	Production
$\delta(A, a) = B$	$A \rightarrow aB$
if B is final	then also add $S \rightarrow \epsilon$

Step3 If start is final, then add $S \rightarrow \epsilon$

Step4 Perform elimination of unit & useless productions

Find RLG for the foll F.A Solution



S1 Rewrite the RLG so that all the productions are of the form:-

$A \rightarrow aB$ OR $A \rightarrow a$

S2 Draw the T.D with vertices labelled as $(V \cup \{\epsilon\})$ and transition labelled as $(T \cup \{\epsilon\})$

S3 Interchange the position of start & final state.

S4 Reverse the direction of all transitions

S5 Rewrite the grammar in left linear fashion.

(1) Find LG for the foll RLG

$S \rightarrow abA$

$A \rightarrow aB$

$B \rightarrow bB / ca$

Step1 S1

$S \rightarrow aC_1$

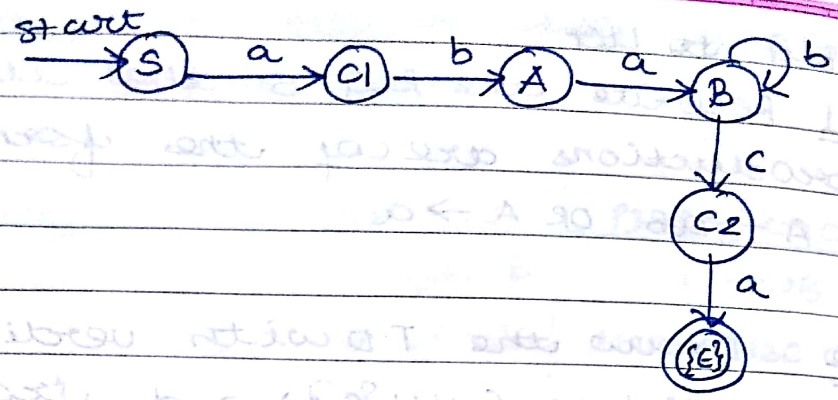
$C_1 \rightarrow bA$

$A \rightarrow aB$

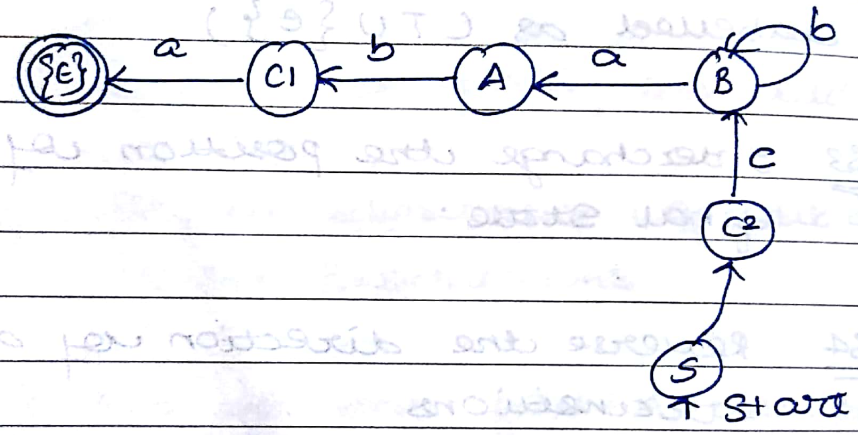
$B \rightarrow bB$

$B \rightarrow ca$

S2



S3 & S4



S5

$S \to C_2 a$

$C_2 \to Bc$

$B \to Bb / Aa$

$A \to C_1 b$

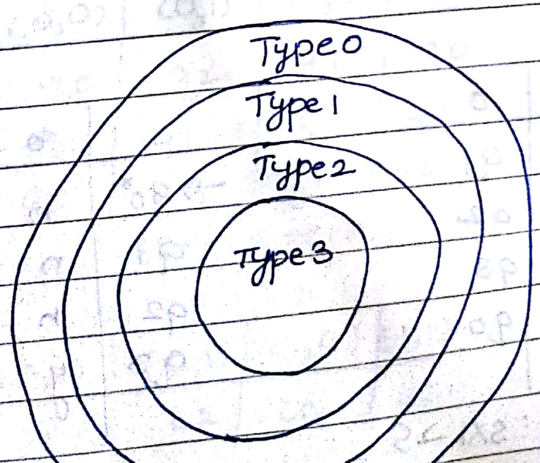
$C_1 \to a$

Type 0
Unrestricted Grammar
Recursively enumerable language
TM

Type 1
Context sensitive Grammar
Context sensitive language
LBA

Type 2
Context Free Grammar
Context free language
DPDA & NPDA

Type 3
Regular Grammar
Regular language
DFA & NFA



Solution to design

S1 Theory

S2 Logic

S3 Implementation

S4 Example

1) Design a FSM to check whether the given unary number is divisible by 4

Solution: $I = \{0\}$

$O = \{y, n\}$

$S = \{q_s, q_0, q_1, q_2, q_3\}$

	0
→ q_3*	q_1
q_0*	q_1
q_1	q_2
q_2	q_3
q_3	q_0

q_0

$0/4 = 1$

$0000/4 = 1$

$000000/4 = 2$

$000/4 = 3$

$0000/4 = 0$

	0
q_0*	q_1
q_1	q_2
q_2	q_3
q_3	q_0

	0
→ q_0*	n
q_1	n
q_2	n
q_3	y

STF : $S \times I \to S$ MAF $S \times I \to O$

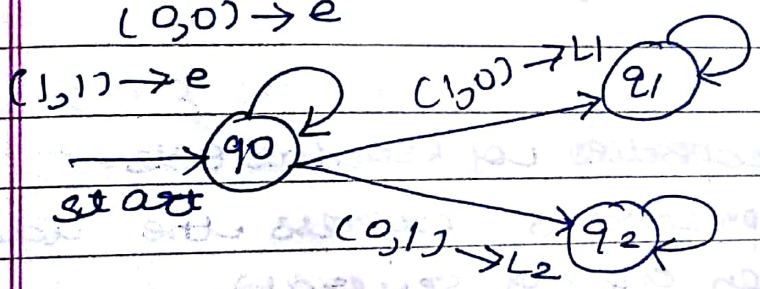
Design FSM to compare two binary no's of same length to determine whether they are equal or which is larger.

Solution $I = \{(0,0), (0,1), (1,0), (1,1)\}$

$O = \{e, L1, L2\}$

$S = \{q_0, q_1, q_2\}$

eg1	eg2	eg3
1 0 1 1	1 1 0 1	1 0 1 1
1 0 1 1	1 0 1 1	1 1 0 1
e e e e	e L1 L1 L1	e L2 L2 L2



(0,0) → L1
(0,1) → L1
(1,0) → L1
(1,1) → L1

(0,0) → L2
(0,1) → L2
(1,0) → L2
(1,1) → L2

	(0,0)	(0,1)	(1,0)	(1,1)
→ q_0	q_0	q_2	q_1	q_0

PAGE NO.
DATE: / /

Regular expressions

Commutative law of union	$L+M=M+L$	T
associative law of union	$(L+M)+N=L+(M+N)$	T
Commutative law of concatenation	$L\cdot M=M\cdot L$	F
Associative law of concatenation	$L(MN)=(LM)N$	T
ϕ is the identity for union	$\phi+L=L+\phi=L$	T
ϵ is the identity for union	$\epsilon+L=L+\epsilon=L$	T/F
ϕ is the annihilator for concatenation	$\phi L=L\phi=\phi$	T
ϵ is the annihilator for concatenation	$\epsilon L=L\epsilon=\epsilon$	F
left distributive laws	$L(M+N)=LM+LN$	T
Right distributive laws	$(M+N)L=ML+NL$	T
Law of idempotence for union	$L+L=L$	T
Law of idempotence for concatenation	$L\cdot L=L$	F

Theory

Closure Properties of Regular Sets

Closure properties express the idea that when one (or several) languages are regular, then certain related languages are also regular

Pg 3-10

The following summarizes the principal closure properties for regular languages:

The union, concatenation, complement, intersection & complement of two regular languages is regular.

The reversal, closure, homomorphism & inverse homomorphism of regular language is regular.

m+L $\boxed{T}$
m+N) $\boxed{T}$
= $\boxed{F}$
=(LM)N $\boxed{T}$
=L $\boxed{T}$
L $\boxed{T/F}$
$L\phi=\phi$ $\boxed{T}$
$L\epsilon=\epsilon$ $\boxed{F}$
$\boxed{T}$
$\boxed{T}$
$\boxed{T}$
=L $\boxed{F}$
idea
tain
lar
m &
er

Dec 2009 / Dec 2010 - 10m

Decision properties of regular language

Theory

Testing emptiness of regular language.
Testing membership in a regular language $\boxed{\text{Pg 3-12}}$
Testing equivalence of states
Testing equivalence of regular languages.

Write RE for the following language

1) $L=\{a^m \mid m \geq 2\}$
Sol $r = aa^+$ Sol 2 $r = aaa^*$

2) $L = \{b^n \mid n = 2\}$
Soln 1 $r = bb$ Soln 2 $r = b^2$

3) $L = \{c^m \mid m \leq 2\}$
$r = c^0 + c^1 + c^2$ $r = \epsilon + c + cc + cc$

4) $L = \{a^m b^n \mid m \geq 4, n \leq 3\}$
$r = aaaa^+(\epsilon + b + bb + bbb)$

5) $L = \{a^m b^n \mid (m+n) \text{ is even}\}$
$r = (aa)^*(bb)^* + a(aa)^* b(bb)^*$

6) $L = \{a^i b \mid i >= 1\}$
$r = a^+ b$ $r = aa^* b$

$L = \{ a^i b^i \mid i >= 1 \}$

we cannot write a regular expression for the above language because it is not a regular language which can be proved by using pumping lemma for regular languages

Stater

Note for non regular languages. FA cannot be constructed

Steps to prove that the language is not regular.

assume
Let L is a regular language. **S1**

Let ~~z~~ assume a string z of that language **S2**

Obtain the equation using pumming lemma for regular language. **S3**

PAGE NO.
DATE: / /

Pg 8-13

Pumping lemma for regular language.

ment - [Let L be a regular language & let z be a word of L such that $|z| \geq n$ where n is the min number of DFA states required for recognizing L,

then as per Pumping lemma, we can write $z = uvw$ where $|uv| \geq n$ and $1 \leq |v| \leq n$,
such that all the strings of the form uv^iw
where $i \geq 0$ would belong to L.]

Prove using Pumping lemma.
$L = \{a^i b^i \mid i \geq 1\}$ is not regular
Let L be a regular language

Let $z = a^n b^n$, such that $|z| \geq n$, where n is pumping lemma constant.

then as per pumping lemma, we can write $z = uvw$
and $1 \leq |v| \leq n$, such that

Let $i = 2$ $\quad |uv^2w| = |uvw| + |v|$

$$1 \le |v| \le n$$

Let us add $|uvw|$ on all sides

$$1 + 2n \le |uv^2w| \le n + 2n$$

$$2n < |uv^2w| < 3n+1 \quad n \ge 1$$

<u>S4</u> $i=1$ $\quad |ab| = 2$

$i=2$ $\quad |aabb| = 4$

$i=3$ $\quad |aaabbb| = 6$

$$2 < |uv^2w| < 4$$

Since L cannot contain any strings of length 3, L is not regular.

Q2) Prove using pumping lemma $L = \{a^i b^{2i} \mid i \ge 1\}$ is not regular

<u>S1</u> Let L be a regular language.

<u>S2</u> Let $z = a^n b^{2n}$, such that $|z| >= n$, where n is pumping lemma constant

<u>S3</u> then as per pumping lemma, we can write $z = uvw$ where $|uv| \ge n$ and $1 \le |v| \le n$, such that all the strings of the form uv^iw where $i > 0$ would belong to L.

Let $i = 2$ $\quad |uv^2w| = |uvw| + |v|$

$$1 \le |v| \le n$$

Let us add $|uvw|$ on all sides.

PAGE NO.
DATE: / /

$1+3n \le |uv^2w| \le n+3n$

$3n < |uv^2w| < 4n+1 \quad n \ge 1$

$i=1 \quad |abb|=3$

$i=2 \quad |aabbbb|=6$

$i=3 \quad |aaabbbbbbb|=9$

$3 < |uv^2w| \le 5$

since L cannot contain any strings of length 4, L is not regular.

$L=\{0^{i^2} \mid i \ge 1\}$

Let L be a regular language

$z=0^{n^2}$, such that $|z| >= n$, where n is a pumping lemma constant.

then as per pumping lemma, we can write $z=uvw$ where $|uv| \ge n$ and $1 \le |v| \le n$, such that all the strings of the form uv^iw where $i>0$ would belong to L.

Let $i=2$ $\quad |uv^2w| = |uvw| + |v|$

$1 \le |v| \le n$

Let us add $|uvw|$ on all sides

$1+n^2 \le |uv^2w| \le n+n^2$

$n^2 < |uv^2w| < n^2+n+1 \quad n \ge 1$

$i=1 \quad |0|=1$

$i=2 \quad |0000|=4$

$i=3$ $\quad |000000000| = 9$

$1 < |uv^2w| < 3$

$4 < |uv^2w| < 7$

Since L cannot contain any strings whose length is not a perfect square L is not regular.

Chap 4 DFA NFA

Dec 2009/10m

1) DFA for strings ending with 011 or 111

$r = (0+1)^* (011+111)$

2) DFA for strings containing 010 as substring

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

June 2010/5m

1) DFA for atleast 3 consecutive zeros

$r = (0+1)^* 000 (0+1)^*$

Dec 2010/10m

1) DFA for odd no of 1's followed by even no of 0's

$r = 1(11)^* (00)^*$

2) DFA for strings which begin & end with diff letters $\Sigma = \{x, y, z\}$