THEORY OF Computation

Language: A finite non-empty set Σ of symbols called the alphabets. From the individual symbols we contruct strings which are finite sequences of symbols from the alphabates. We write $w = abaaaa$ to indicate that the string name w has the specific value abaaa.

⊞ The contradination of two strings w and v is the string obtained by appending the symbols of v to the end of w.

e.g.,

$w = abaaa$

$v = bb$

$wv = abaaabb$

$w^R = aaaba$ $\quad |w| = 5$

Ⓐ Length of a string 'w' is denoted by $|w|$.

$|\lambda| = 0 \rightarrow$ empty string.

$$w = \lambda w = w\lambda.$$

$$|wv| = |w| + |v|$$

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

W^n is the string that we obtained after repeating w 'n' times.

e.g → $w^1 = abaaa$; $w^2 = ww = abaaaabaaa$.

$w^0 = \lambda$

Grammer:

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

G is defined by a quadruple. V, T, S, P

- V → finite set of objects called variable
- T → " " " " " terminal symbols.
- $S \in V$ is a special symbol called the start variable.
- P → finite set of productions.

P is given by:

$S \to aSb$

$S \to \lambda$

$G = (\{S\}, \{a, b\}, S, P)$

$S \Rightarrow aSb$
$\Rightarrow a\lambda b$
$\Rightarrow ab$

$S \Rightarrow \lambda$
$S \Rightarrow aSb$
$\Rightarrow aaSbb$
$\Rightarrow aa\lambda bb$
$\Rightarrow aabb$

$S \Rightarrow aSb \Rightarrow a\lambda b \Rightarrow ab$

$S \overset{n}{\Rightarrow} ab$ after 'n' no of steps

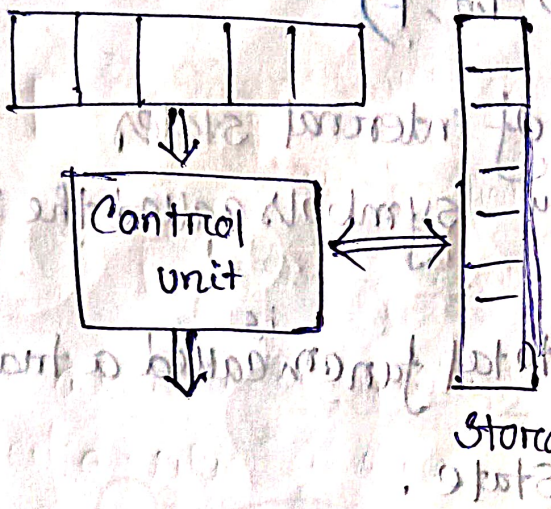
$S \overset{2}{\Rightarrow} ab$
$S \overset{4}{\Rightarrow} aabb$
$S \overset{6}{\Rightarrow} aaabbb$.

GOOD LUCK™

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Automata



* The internal state of the control unit and the next time state is determined by the next state or the transition funcn. the term configeration is used to refer a particular state of control unit, input file and temporary storage. The transisation of the automata from one configuration to the next is called a MOVE. An automaton whose o/p response is limited to sample yes/no is called acceptor. Presented with an o/p string an acceptor either accepts the string or rejects it.

A finite acceptors is an automaton with no temporary storage. Thus it can't remember things during computation. A deterministic automaton is one in which each move is uniquely determined by the current configuration. * for a non-deterministic automaton, it may have several moves so we can only predict a set of possible actions.

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

A deterministic finite acceptor or DFA is defined by the quintuple (5)

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

where,

- Q = is a finite set of internal states
- Σ = the symbols called the i/p alphabet.
- $\delta : Q \times \Sigma \rightarrow Q$ is a total function called a transition function.
- $q_0 \in Q$ is the initial state.
- $F \subseteq Q$ is a set of final state.

At the initial time it is assume that the DFA is in the initial state q_0, with it's i/p mechanism on the left most symbol of the i/p string. During each move of the automaton, the i/p mechanism advances one posn to the right, so each move consume one i/p symbol. When the end of the string is reached the string is accepted if the automaton is it one of it's final states. Otherwise the string is rejected.

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

If the DFA is in state q_0 and the current i/p symbol

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

is a the DFA will go the state q_1.

set of all states, alphabet.

e.g. $M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$

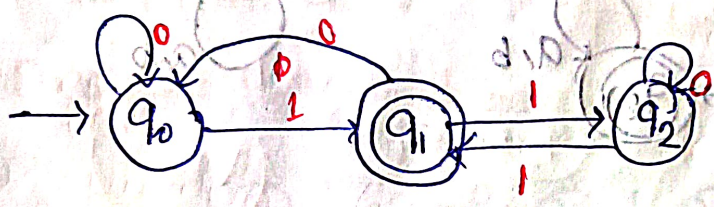
Q δ is given by.

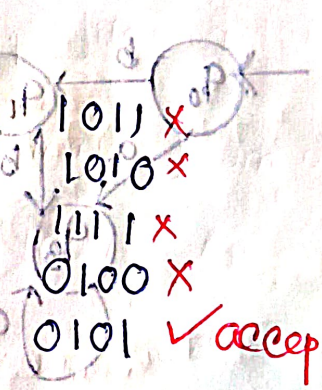
$\delta(q_0, 0) = q_0$ $\quad \delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_0$ $\quad \delta(q_1, 1) = q_2$

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

Ans

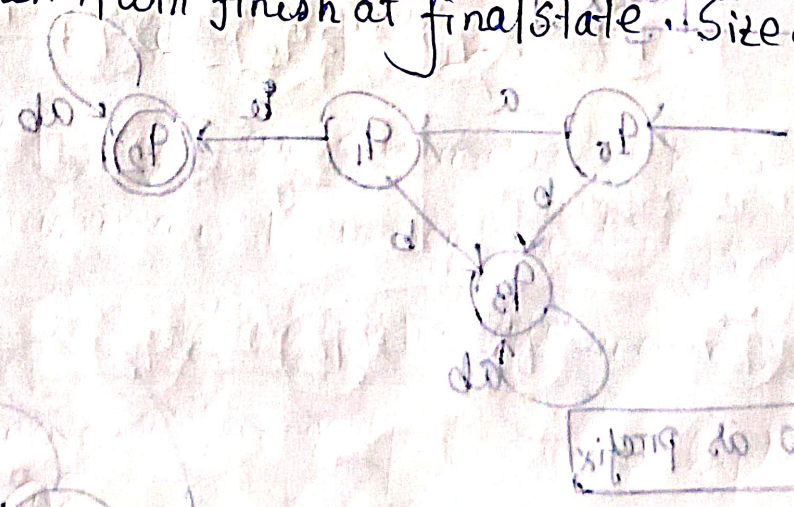


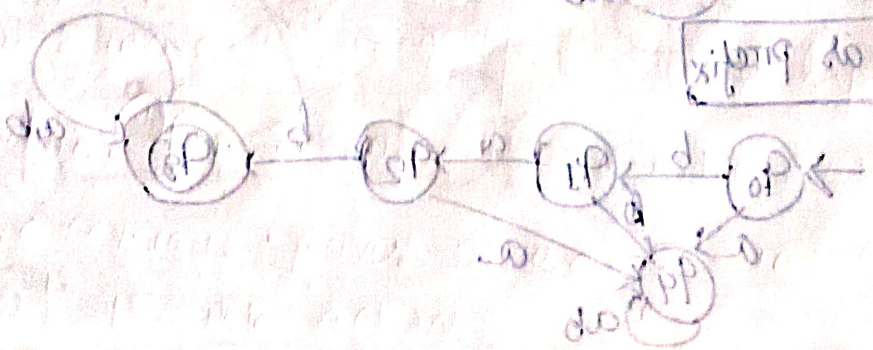


(0) → finalstate ; min: 1 ; maxm: ∞

0 → initial state ; min: 1 ; maxm: 1.

→ accepted when it will finish at finalstate. Size doesn't matter.





GOOD LUCK™

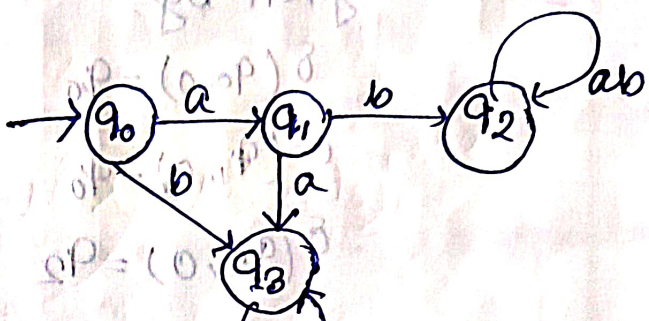
TOPIC NAME : _____ DAY : _____

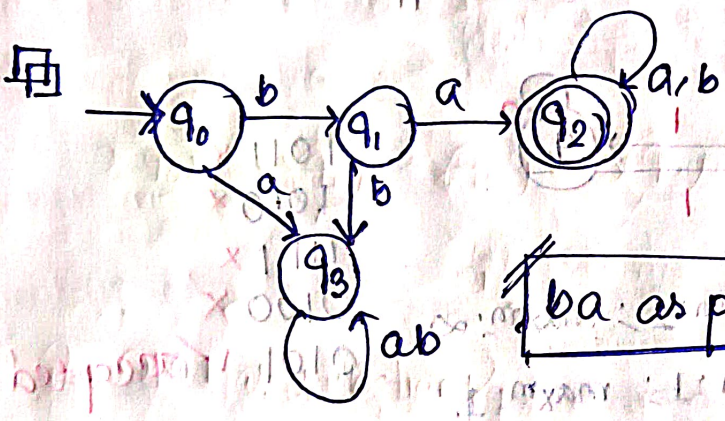
TIME : DATE : / /

Q. Find a redundestic finite acceptors that recognizes set of all strings on $\Sigma = \{a, b\}$ starting with the prefix ab.

ab ✓
aba ✓
ababa ✓
abaaa ✓

aabb ×
aaba ×
baba ×
bbba ×

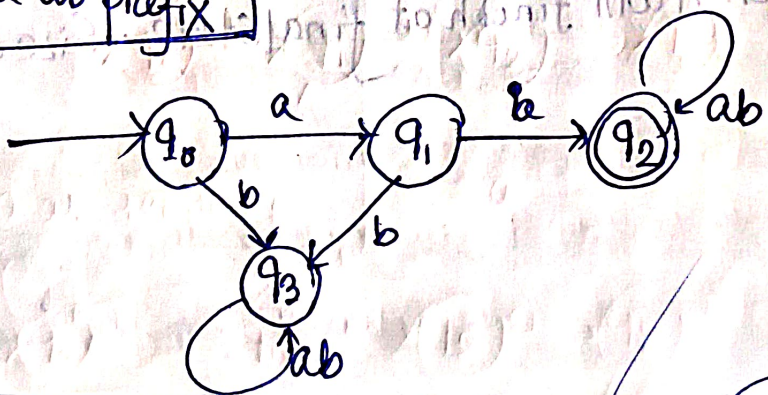




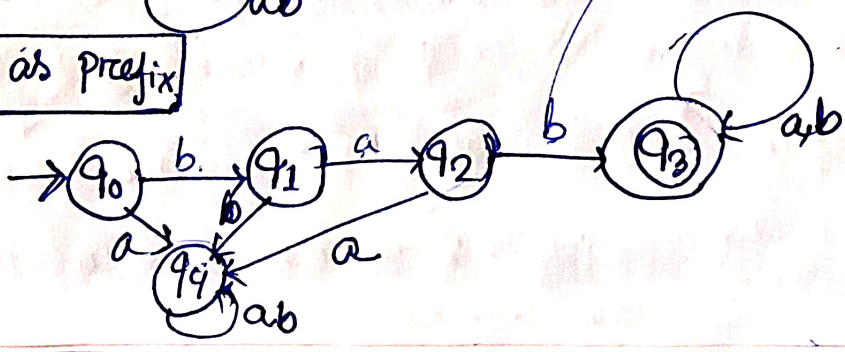
ba as prefix.

DFA is not unique

aa as prefix.



bab as prefix



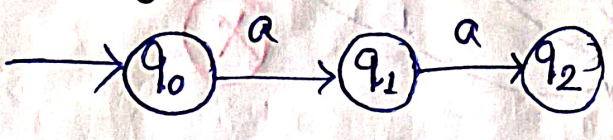
GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

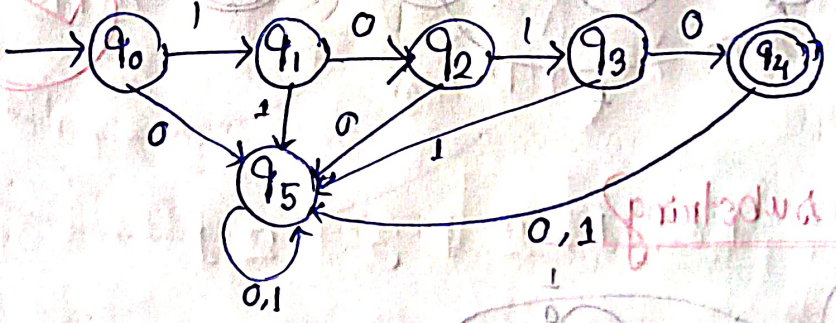
Assg: (i) bbb as prefix (ii) ab as suffix (iii) aa as suffix

Q. aa as suffix

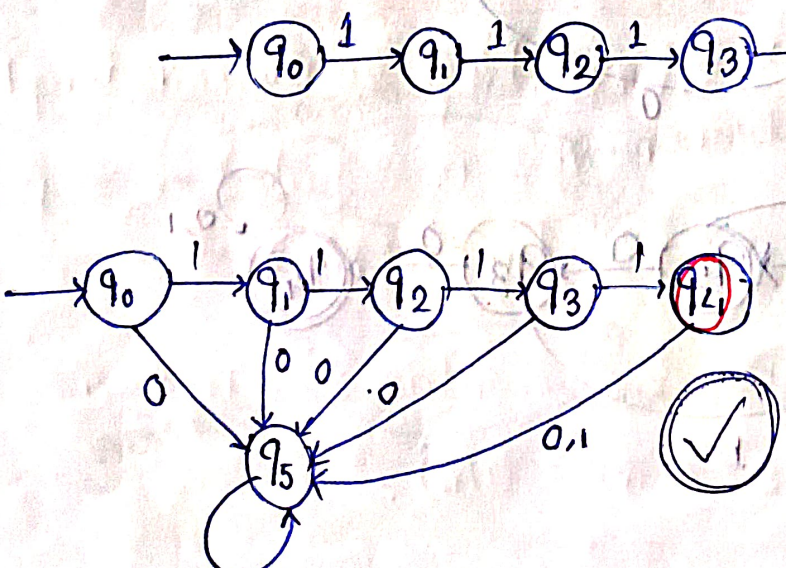


Regular Language: A lang L is called regular if and only if there exists DFA(M) such that $M = L(n)$, L as a function of n.

Q. Design a DFA which accepts only 1010 only.



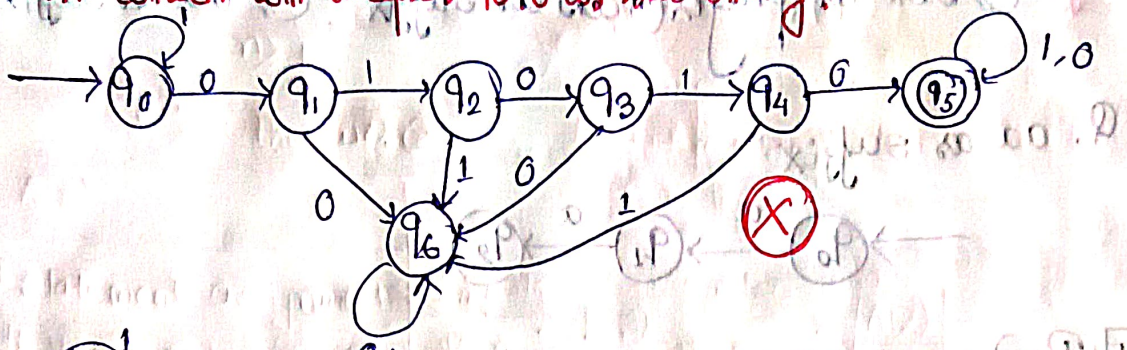
Q. 111 only

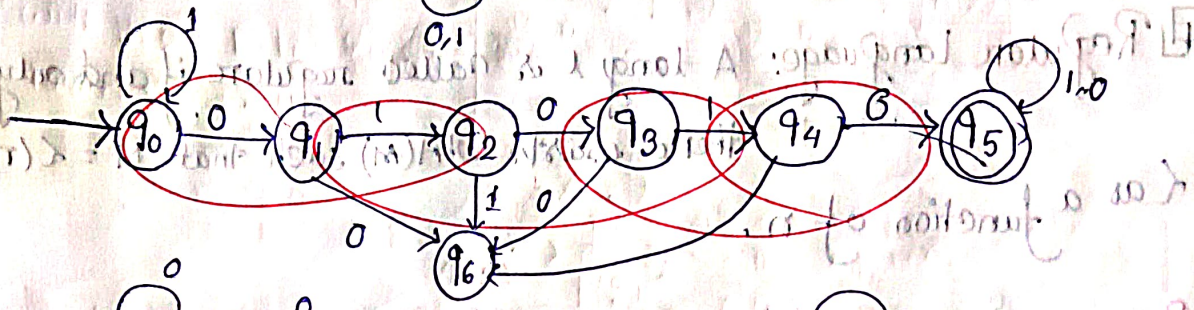


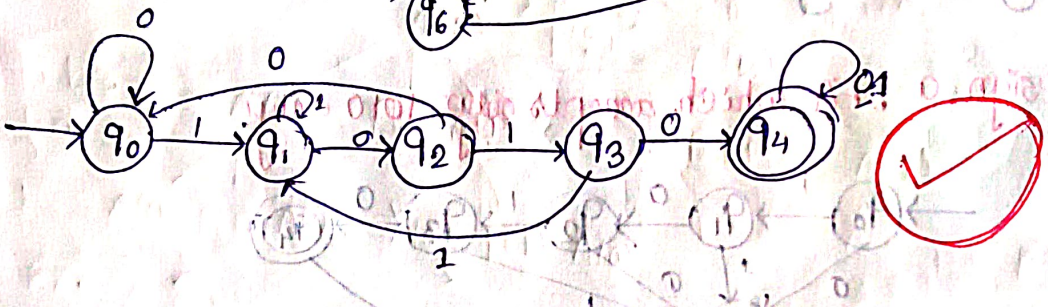
TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

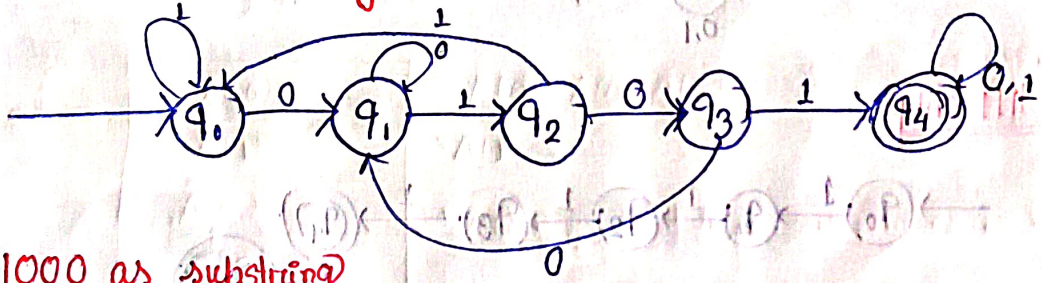
Q. DFA which will accepts 1010 as sub-string.



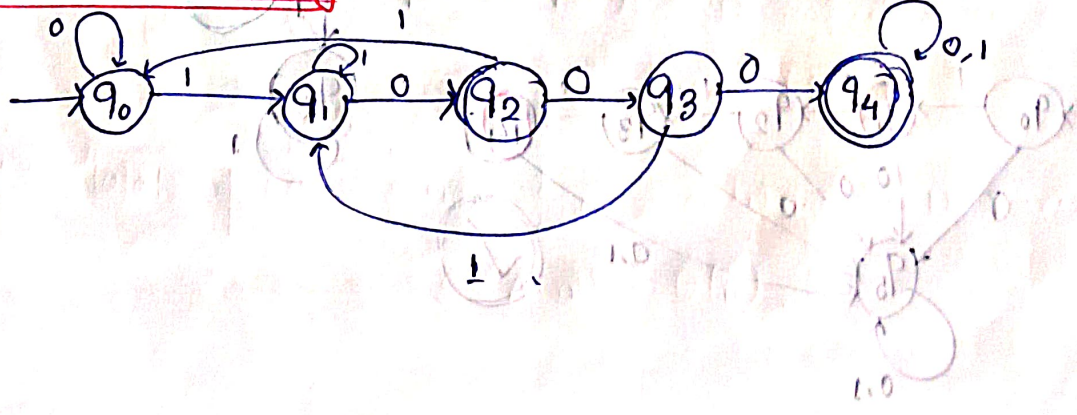




Q. 0101 as substring



Q. 1000 as substring

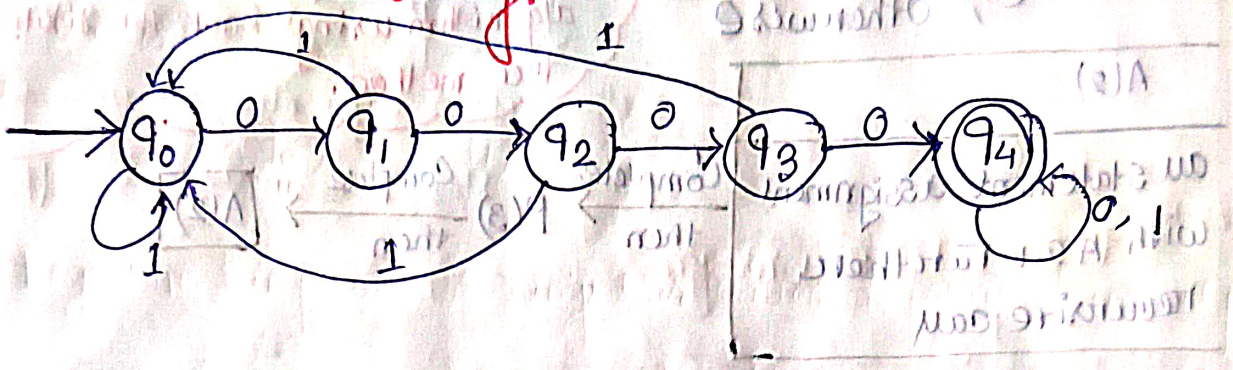


GOOD LUCK™

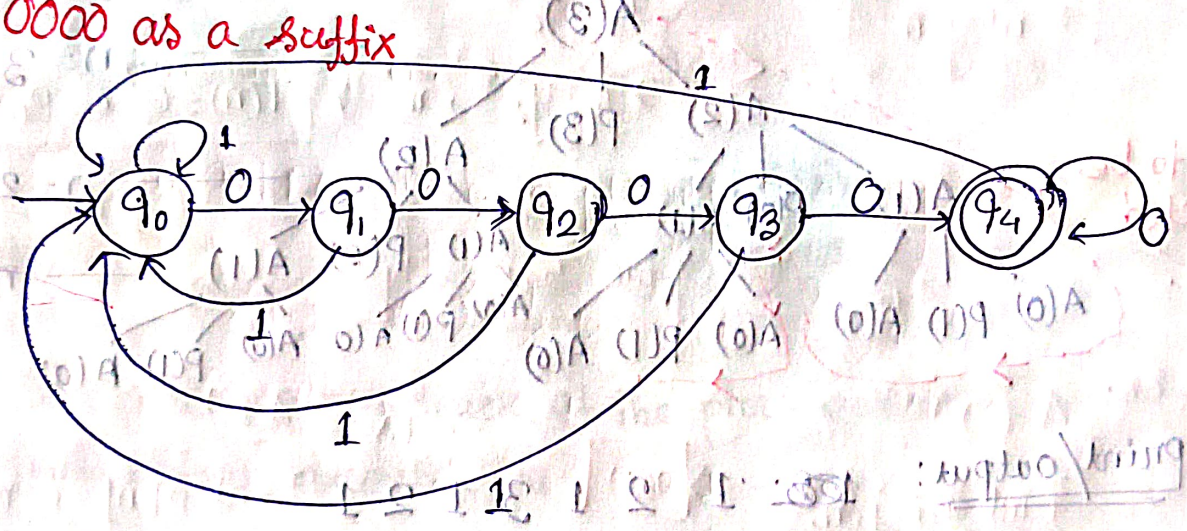
TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

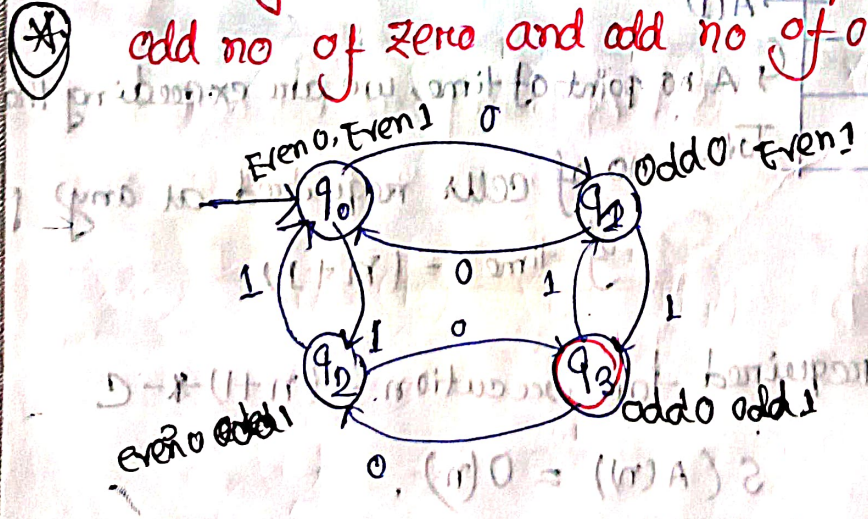
Q. Construct a DFA which will accept all the string which has 0000 as substring.



Q. 0000 as a suffix



Q. Design a DFA which accepts all the strings containing odd no of zero and odd no of one



Q. odd zero even one.

Q. even zero even one.

Q. even zero odd one.

GOOD LUCK

TOPIC NAME : _____ TIME : DATE : / /

TOC

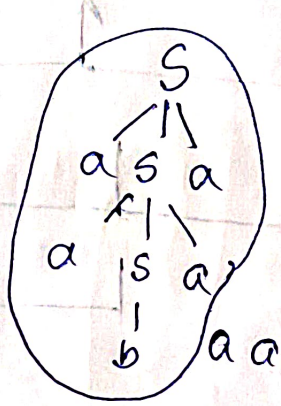
Context - Free Grammer -

A language is said to be CFG iff there a grammar G such that where the form $A \to x$, $A \to b$ and $x \in (V \cup T)$

Q. Construct a CFG to generate set of pallindrome over alphabet a, b.

$S \to ABa$
$S \to a$
$B \to b$
$S \to \epsilon$
$ab \leftarrow aba$
$S \to ABa$
$\to aBa$
$\to aba$
$S \to \epsilon$

$S \to ASB$
$S \to \epsilon$
$A \to aB$
$B = bB$
$B \to \epsilon$



aabaa

$S \to asa/bsb$
$S \to a/b/\epsilon$

*. Production Tree

Q. Same no of a & same no of b.

Q. Even palindrome and odd palindrome.

5 marks

$S \to aSa \mid bsb$
$S \to a \mid b$
↳ Odd

abba → even ; aba → odd.

$S \to aSa \mid bsb$
$S \to \epsilon$
↳ even

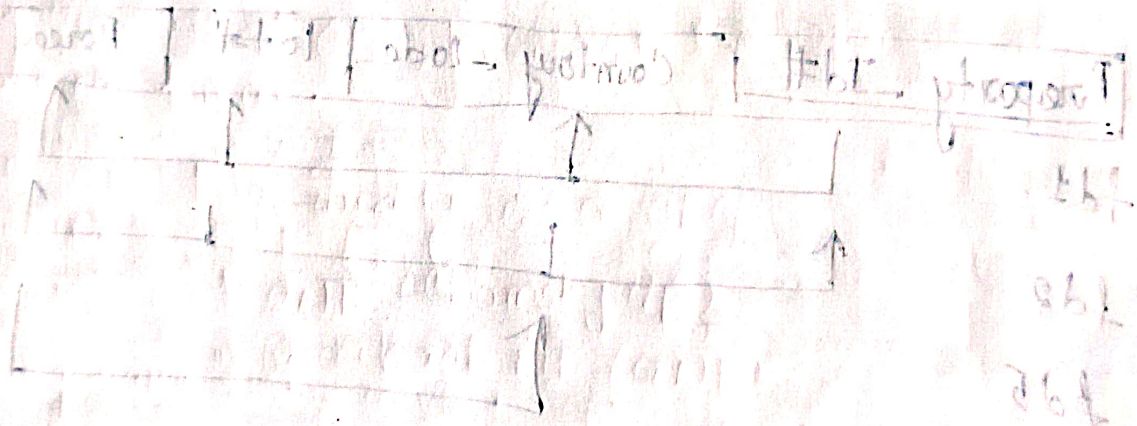
GOOD LUCK

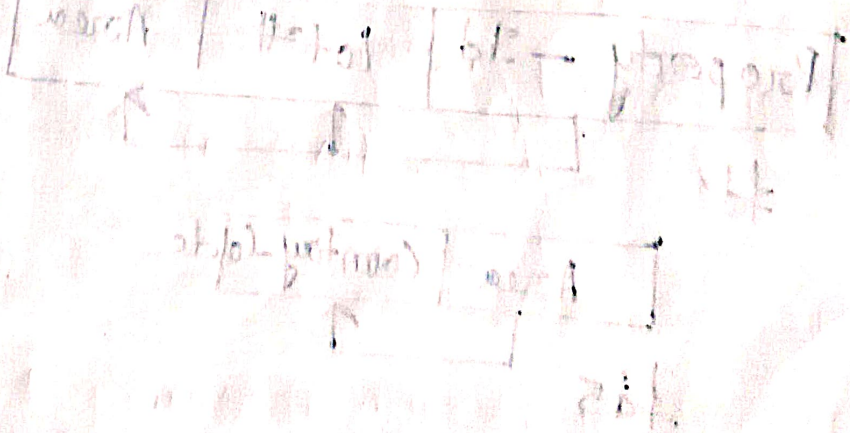
TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

Q. ab where a is followed b and no. b is greater than a.

Q. for a language $L = 0^i, 1^j, 0^k$, where $j > i+k$.





GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

TOC

Normal Forms

1) In a context free grammer in the RHS of the production can be any string of variables and terminals. When production in G satisfied certain restriction then G is said to be in normal form.

There are two widely used CFG.

1) CNF (Chomsky NF)
2) Greibach NF (GNF)

a context free gramer G is in CNF if any production is in the form $A \to BC$ or $A \to a$, when A, B, C are non-cardinals and a is terminals. For every CFG there are equivalent grammer in CNF.

For example-

Q. Find an equivalent grammer in CNF for the grammer

(5 marks)

$S \to aAD$

$A \to aB / bAB$

$B \to b$

$D \to d$

$\Rightarrow$

$S \to C_aD$

$C_a \to a$

$D_1 \to AD$

$A \to C_aB$

$A \to C_bD_2$

$C_b \to b$

$D_2 \to AB$

GOOD LUCK

TOPIC NAME : _____ DAY : _____

TIME : DATE : / /

TOC

I) NFA → DFA conversion

II) Simplify a CFG

III) To convert a CFG to CNF

IV) Short note of Chomsky NF. → 5 marks (1 page)

V) Closure property of RE

VI) what is Left most & Right most derivation.

VII) what is Ambigeous Grammer, Example.

VIII) Divisible by 3, 7. mod 2, mod 3 types Question.

IX) Even odd, (4 ta stepwala Question)

X) Construct a PDA to accept a CFG

XI) How to draw a perse tree from CFG.

XII) Regular Grammer.

Starting with aa_ bb_ —

corcresponding DFA.

13) palindrome. (CFG)

14) What is Tunning machine.

GOOD LUCK