21/08/23

Assignment No. 2

Q.1) Explain the concept of DFA along with it transition table & representation transition diagram.

→ DFA - (Deterministic Finite Autometa)

Any DFA is tuple of five factors. so formally DFA is defined as -

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

where, δ = Transition function

Q = set of all states

Σ = I/P of alphabete

q_0 = Initial set

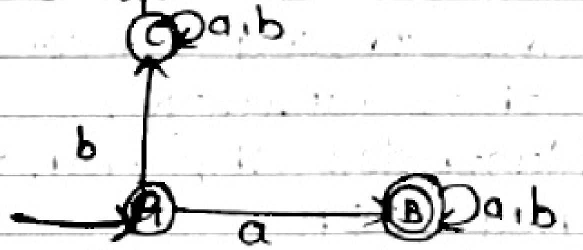
F = set of final state

$\delta : Q \times \Sigma \longrightarrow Q$

Note:-

In DFA every state must show transition of all input character exactly once.

Sample DFA for demonstration.



with respect to this machine, we can define the following elements.

$Q = \{A, B, C\}$ $q_0 = \{A\}$

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

To define delta (δ) at each state, we can have following entries

$\delta(A, a) = B$ $\delta(B, b) = B$

$\delta(A, b) = C$ $\delta(C, a) = C$

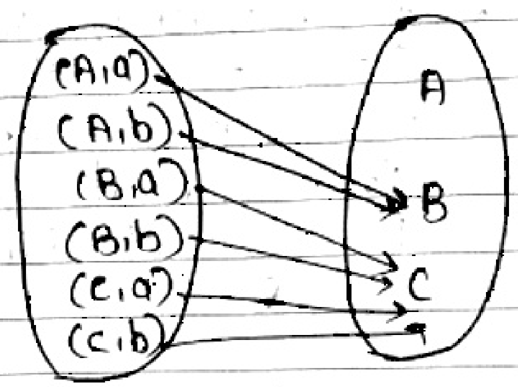
$\delta(B, a) = B$ $\delta(C, b) = C$

Transition of character 'a' at state A machine goes to state B.

Above entries can be simulated by defining the transition function δ(delta) at machine level as follows.

$\{A, B, C\} \times \{a, b\}$

$Q \times \Sigma$ Q



The DFA can be represented in any of the 2 ways.

1: Transition graph - It is a state transition diagram traced according to symantic of the language.

The fig (a) illustrates the transition graph of concerned DFA.

2. Transition table - It is simply a matrix of finite no. of rows & columns. All the I/P characters placed on column whereas all the states form the rows of the table.

Following diagram illustrate the transition table of DFA.

Transition table

	a	b
A	B	C
B*	B	B
C	C	C

→ : initial state
* : final state.

In the above DFA each state shows transition of each I/P character exactly once That is, state A shows the transition of character a & b exactly once.

IIly, state B also shows transition of character a & b exactly once in the form of loop.

IIly, state C also shows the transition of character a & b exactly Once in the form of loop.

∴ Above machine is perfectly a DFA.

Q.2) Explain the process of string acceptance & language acceptance.

→ Let ω is any string over Σ^* (universal language), if we start the processing of ω for initial State q_0 of DFA upon processing the string completed if machine rechese at final state or accepting state then a censered string ω is accepted by that DFA.

Formally string acceptance can be expressed as

$$\delta(q_0, \omega) = P \in F$$

For example, consider the I/P string ABA
when, we start processing of ABA
from initial state ABA the state A
will process 1st will reach to next
state B. Now, it is the responsibility
of state B to process remaining part
of the I/P. According to state B will
process 'b' from the I/P string it will
move on the loop & remains in the
same state.

Now, Again state 'B' will process the
last character 'a' from the I/P string
in same manner. i.e. 2nd time it will
move on the loop & processes character
'a', input string is completely over &
machine reaches at final state B. There-
fore, the string 'aba' has been accepted.

consider, Another string bab, upon proces-
sing b from initial state A machine rea-
ches next state C. Now, it is responsi-
bility of state c to process remaining
part of the input.

The state c will process charactor 'a'
by moving on the loop & remains in the
same state.

IIly, The state c will be in the last
character 'b' in the same manners &
remains in state c itself.

So, upon processesing entire string
machine reaches at state c which is
actually non-Final. Infact it is a dead

state.
Hence the string bab is rejected by the machine.

Language acceptance by DFA -
If L is any formal language then if all the strings of the language accepted by the DFA M then & then only language L is accepted by DFA M.
In that case, we call L as regular language.
Formally, we can be expressed as

$$L(M) = \{ \omega / \omega \in \Sigma^* ; \delta(q_0, \omega) = P \in F \}$$

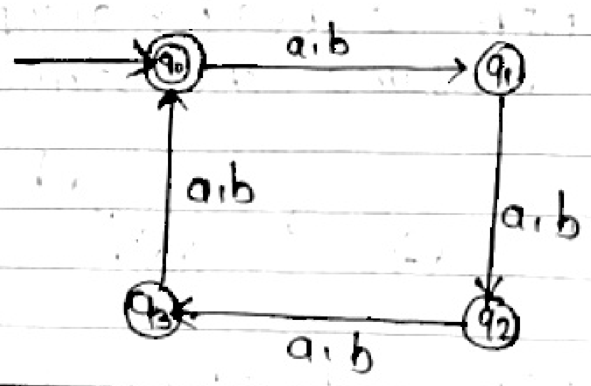
---

3) construct the DFA over {a,b} to accept the string containing all the string exactly divisible by 4.

→ Since, this language containing all the string of length exactly divisibility by 4.

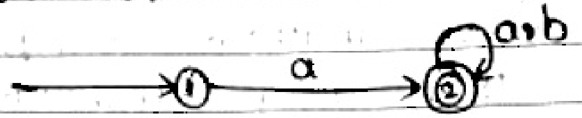
$$L = \{ \epsilon ; aaaa, aaab, bbbb \ldots \}$$

The machine can be drawn as,

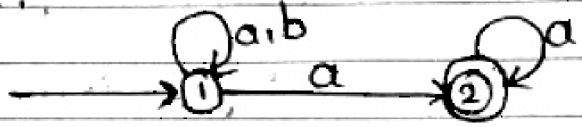


Q.4) construct the NFA for all the already drawn the DFA :

1) construct the NFA over {a,b} accepting the language containing all the string strat with a.

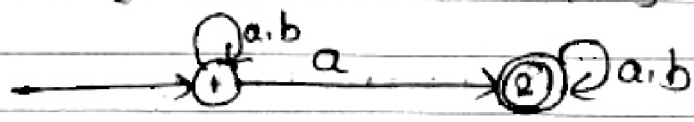


2) construct the NFA over {a,b} accepting the language containing all the string ending with a.

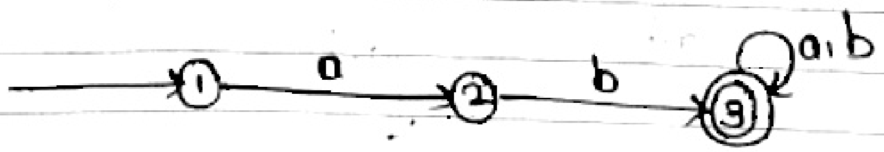


3) construct the NFA over {a,b} which accept the language containing the string having a as substring.

L = {a, baa, aaa ....}



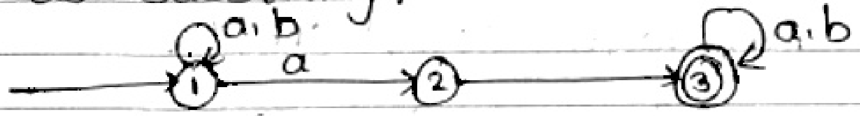
4) construct the NFA over {a,b} accepting the language that contains the string starting with ab.



5) Construct the NFA over {a,b} accepting the language that containing the string having

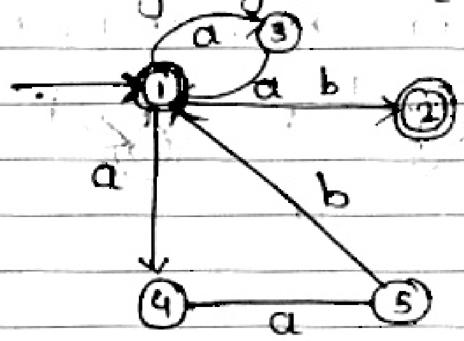
classmate
Date
Page

ab as substring.

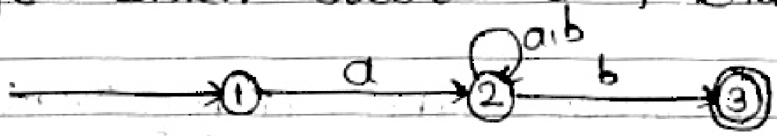


6) Construct the NFA over{a,b} accepting the language that containing the string having end with ab.

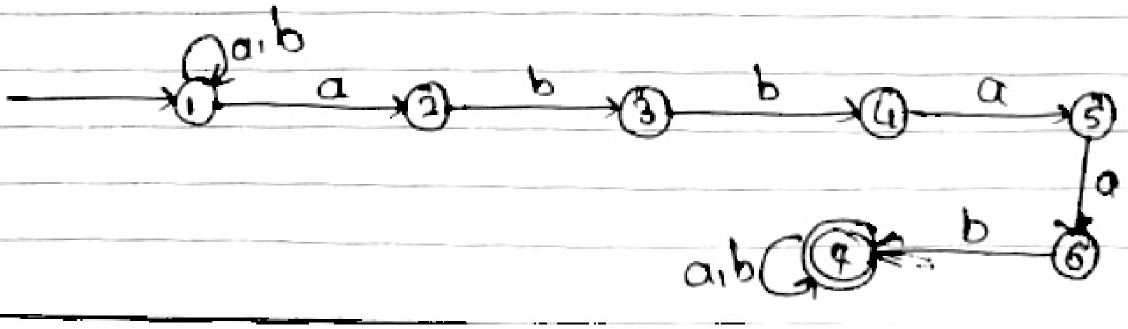
7) construct the NFA over{a,b} accepting the language L = {aa, aab}* {b}



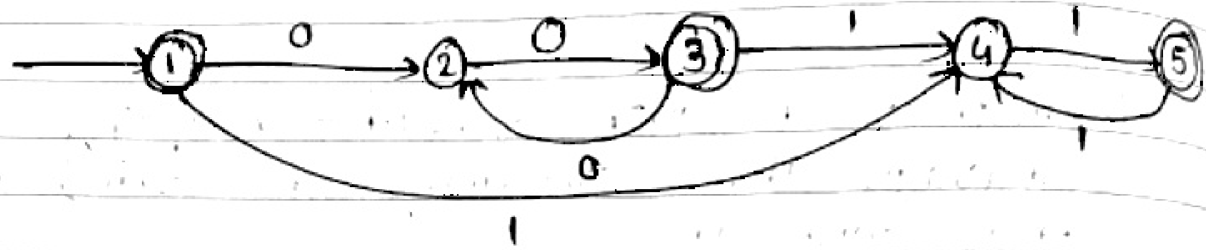
8) Construct the NFA over{a,b} accepting the language which start 'a' & end with 'b'.



9) Construct the NFA over{a,b} accepting the language abbaab as substring.

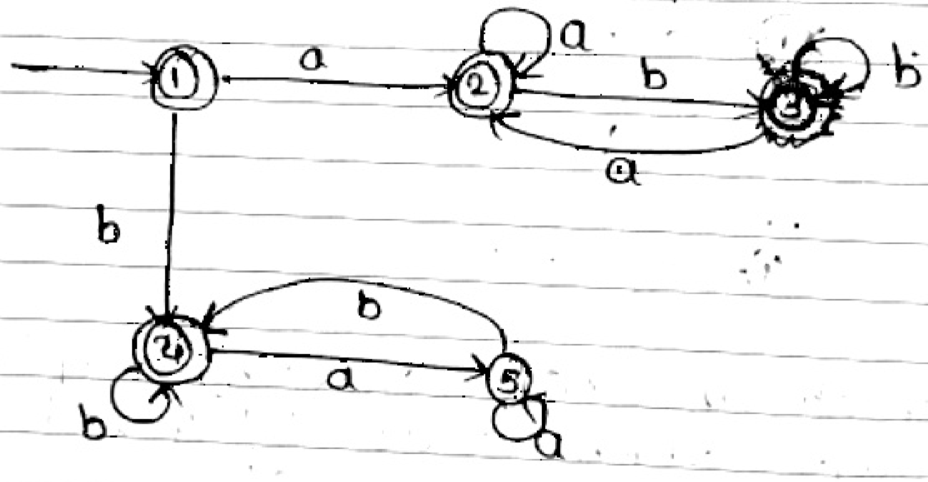


10) construct the NFA over{0,1} accepting the language $(00)^*(11)^*$.



11) construct the NFA over{a,b} that accepting the language that containing the string having starting & ending with same symbol.

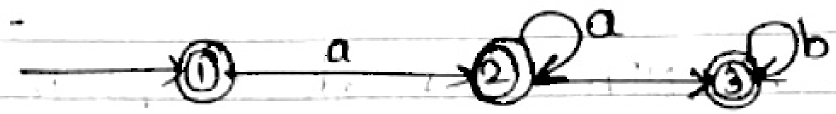
L= {ε, a, b, aa, bb, aba, bab ...}



12) construct the NFA over{a,b} accepting the language containing all the string ending with 'b' but not ending aa.

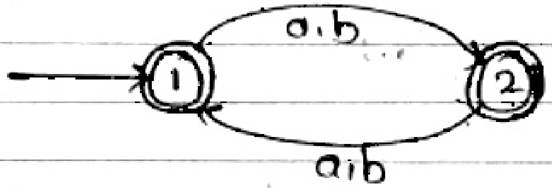
classmate

13) construct the NFA over {a,b} to accept the language $L = \{a^n b^m / n,m \geq 0\}$



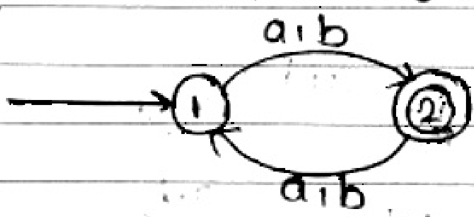
① construct NFA containing language of string even length.

→ L = {aa, bb, ε, aabb, aaab, baba ---- }



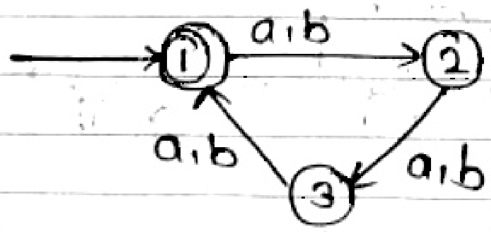
② construct NFA over {a,b} of all odd length string.

L = {a, b, aab, bab, baa ------ }



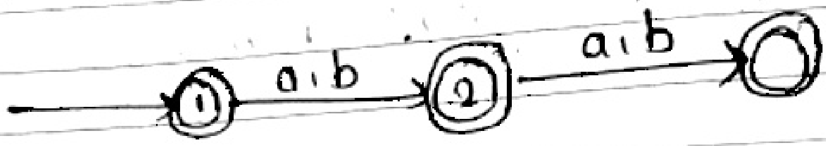
③ Draw NFA to accept language containing all string exactly divisible by 3.

→ L = {ε, aba, baaabaabb ----- }



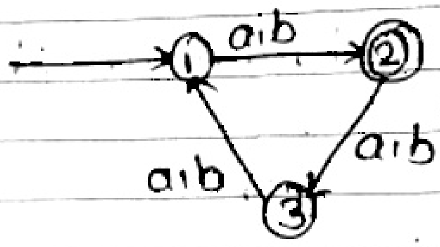
④ construct NFA to accept language containing string of length atmost 2.

L = {ε, a, b, aa, ba, bb, ab}



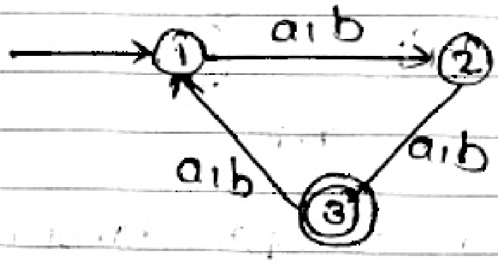
5) construct NFA over a,b accept the string 1 mod (3)

L = {a, b, aaaa, abab, aabbaa ----}

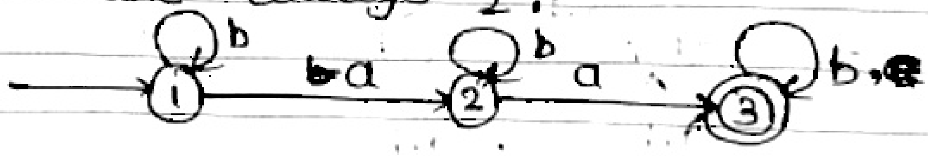


6) construct NFA over a, b where length is congurent to 2 mod(3).

L = {aa, bb, ab, ba, aaaa, aaaab ---}

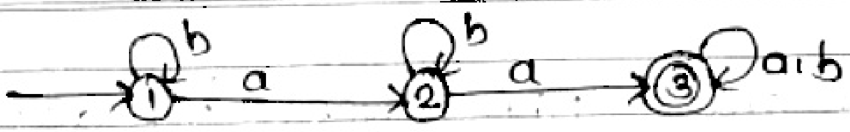


7) construct NFA over a, b where number of 'a' are always 2.



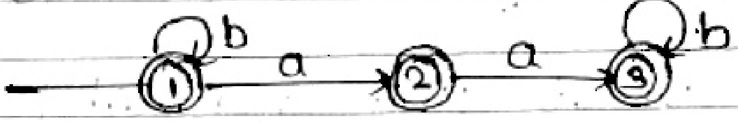
8) construct NFA over a, b where number of 'a' are atleast 2.

L = {aa, aba, aaab, aabaab ---}

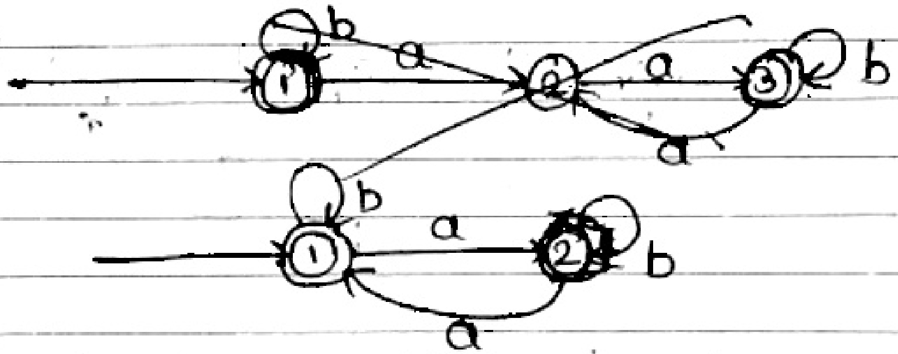


9) construct NFA over {a, b} to accept all the string containing atmost two number of a.

L = {ϵ, b, aa, a, ab, ba, bb, abab ...}

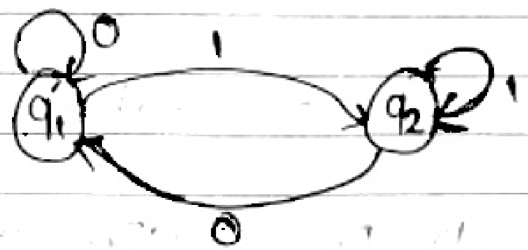


10) construct NFA over {a, b} accepting language containing all the strings having even of no. of 'a'.



11) construct NFA over (0, 1) which when interrupted as binary number gets exactly divisible by 2.

	0	1
q_1	q_1	q_2
q_2	q_1	q_2

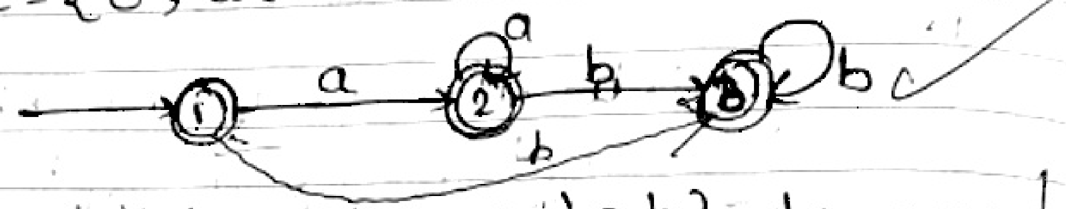


12) construct NFA over a, b which start a

a a b b

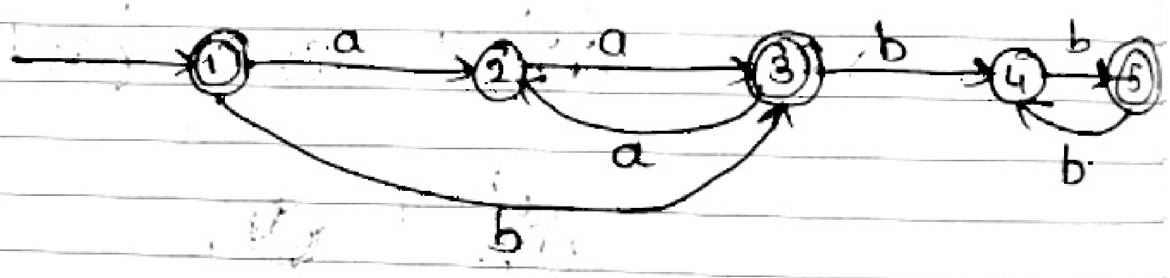
* 13) construct NFA over {a, b} accept language $L = \{a^n b^m \mid n, m \geq 0\}$

$L = \{\epsilon, ab, aabb, aaabbb \dots\}$



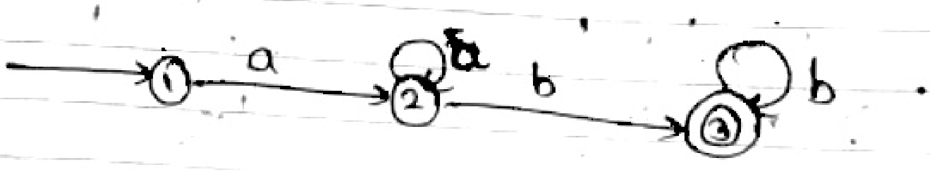
14) construct NFA over {a, b} to accept the language $\{aa\}^* \{bb\}^*$

$L = \{\epsilon, aa, bb, aabb, bbbb \dots\}$



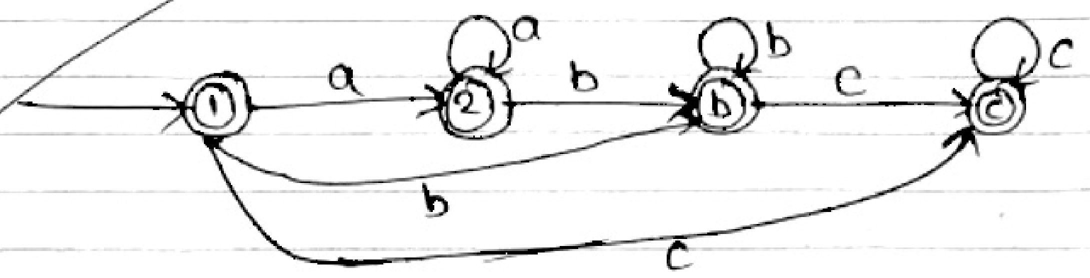
15) construct NFA over {a, b} to accept the language $L = \{a^m b^n\}$ $m \geq 1, n \geq 1$

$L = \{ab, aab, aabb \dots\}$



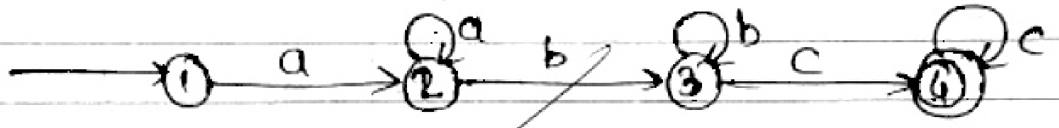
16) Draw NFA to accept the language $L = \{a^m, b^n, c^k \mid m, n, k \geq 0\}$

$L = \{\epsilon, a, b, c, ab, bc, ac, abc, aab \dots\}$



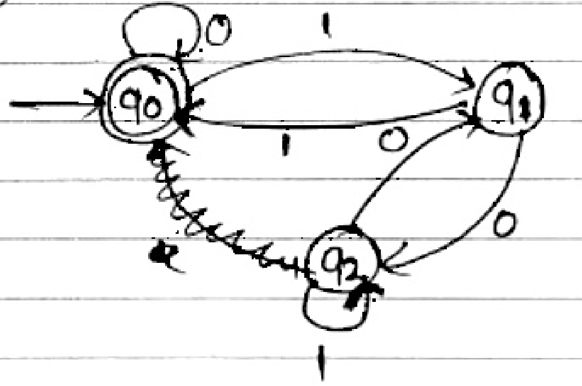
*17) Draw NFA to accept the language $L = \{a^m b^n c^k \mid m, n, k \geq 1\}$

L = {abc, aabbcc, abbcc.- - - -}



18) construct NFA over {0,1} when enterpred as binary number gets divisible by 3

	0	1
q_0	q_0	q_1
q_1	q_2	q_0
q_2	q_1	q_2



28/8
good

Q.2) Explain the algorithm to eliminate unit production.

Q.3) Explain the concept of useful and useless symbols in grammer.

Q.4) union, concatenation, Kleens closure of CFG.

IMP. 15. please carefully see the example of CFG for

L = { a^ib^jc^K | j <> i+K }