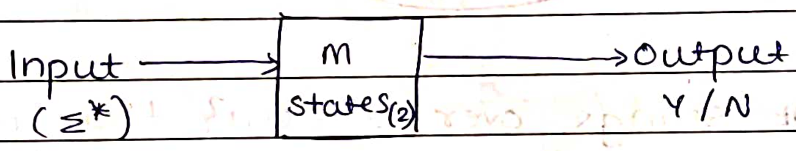
Date: / /

(Σ) Alphabet : finite sets of symbols
(G) Grammar : finite sets of rules
(G) Language : finite / infinite sets of strings over alphabet set Σ

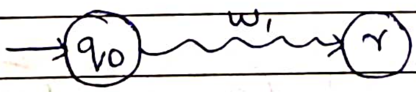
Strings: A sequence of symbols from Σ arranged one after the other.

Ex. L is set of strings over $\Sigma = \{a, b\}$ that contain "aa".
$L = \{ aaa, aaba, baa, \ldots \}$
$\bar{L} / L' = \{ aba, bab, abab, \ldots \}$
$\bar{L} \Rightarrow \Sigma^* - L \Rightarrow$ compliment of L.

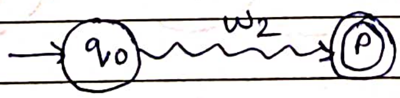
* DFA (Deterministic finite Automation)



$M = (Q, \Sigma, \delta, q_0, F)$
$Q \Rightarrow$ finite set of states
$\Sigma \Rightarrow$ finite set of input symbols
$\delta \Rightarrow$ Transition function $Q \times \Sigma \rightarrow Q$
$q_0 \Rightarrow$ Intial / Start state
$F \Rightarrow$ ~~Final~~ Finite set of final / accepting state



$w_1 \notin L$ reject



$w_2 \in L$ accept

$L(M) = \{ w \mid \delta^*(q_0, w) = P$ where $P \in F \}$

Example | DFA to accept strings over $\Sigma = \{0,1\}$ that ends with "110".

⇒ 5 tuple representation

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

$Q = \{q_s, q_1, q_{11}, q_{110}\}$

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

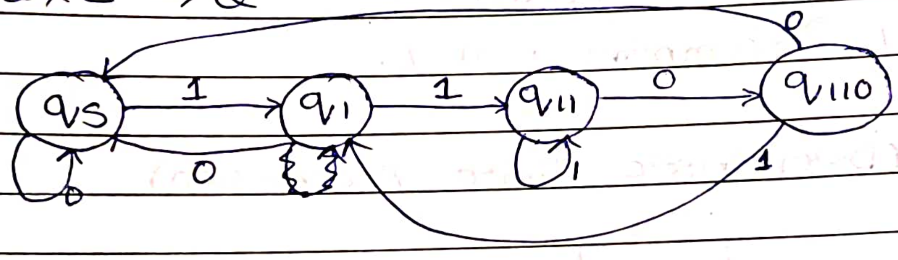
δ = Transition function

$Q \backslash \Sigma$	0	1
q_s	q_s	q_1
q_1	q_s	q_{11}
q_{11}	q_{110}	q_{11}
q_{110}	q_s	q_1

$q_0 = q_s$

$F = \{q_{110}\}$

$\delta = Q \times \Sigma \rightarrow Q$



Example DFA to accept strings over $\Sigma = \{0,1\}$ that contain '1011'.

⇒ 5 tuple representation

$M = (Q, \Sigma, \delta, q_0, \cancel{q}\, F)$

$Q = \{q_s, q_1, q_{10}, q_{101}, q_{1011}\}$

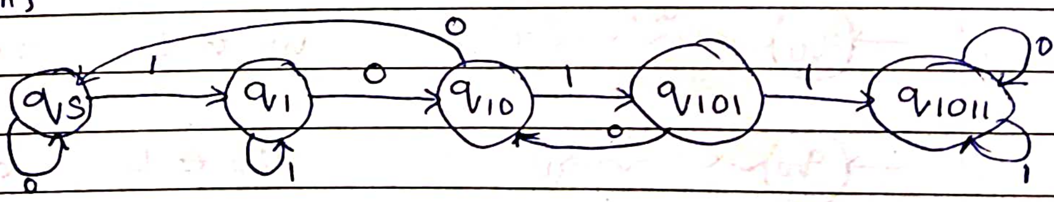
$\Sigma = \{0,1\}$

$\cancel{\delta}\ \delta = Q \times \Sigma \rightarrow Q$

$q_0 = q_s$

$F = \{q_{1011}\}$

$Q \backslash \Sigma$	0	1
q_s	q_s	q_1
q_1	q_{10}	q_s
q_{10}	q_s	q_{101}
q_{101}	q_{10}	q_{1011}
q_{1011}	q_{1011}	q_{1011}



$q_s \rightarrow$ start state

$q_1 \rightarrow$ i/p ends with '1'

$q_{10} \rightarrow$ i/p ends with '10'

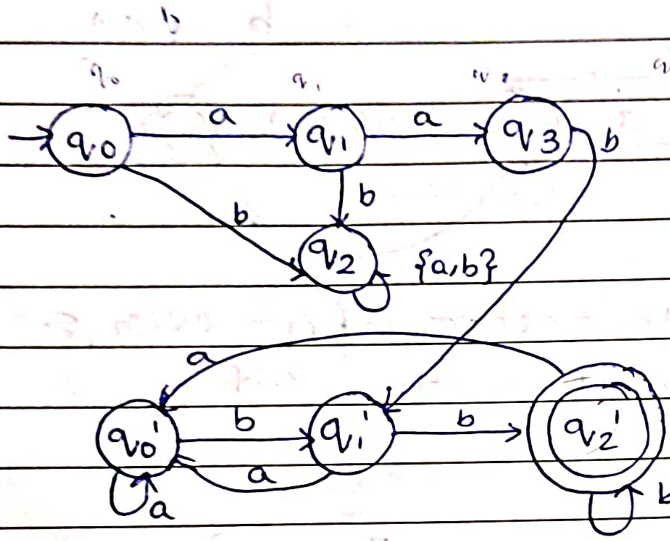
$q_{101} \rightarrow$ i/p ends with '101'

$q_{1011} \rightarrow$ i/p ~~ends~~ contain with '1011' & hence accept

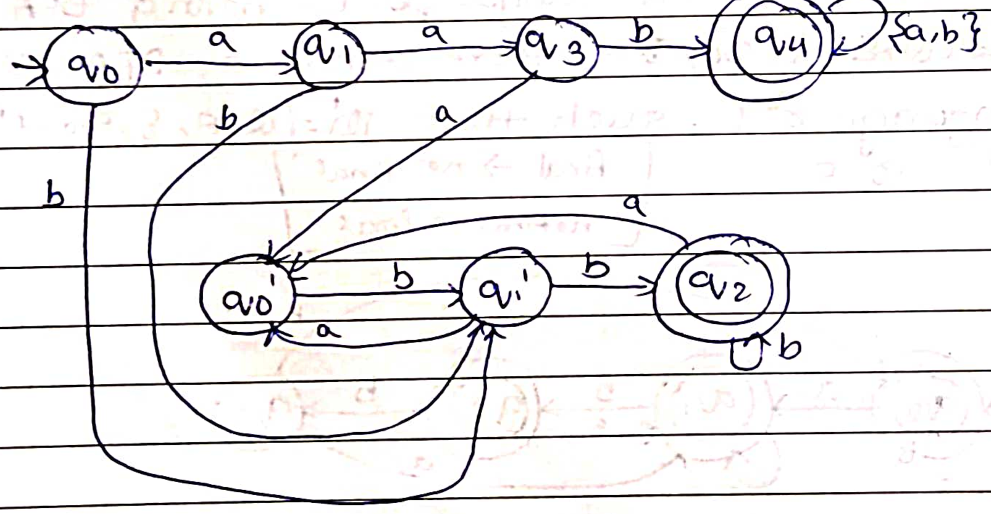
Date: / /

example Design DFA that accepts string over $\Sigma = \{a, b\}$ that ~~cont~~ starts with "aab" and ends with "bb".

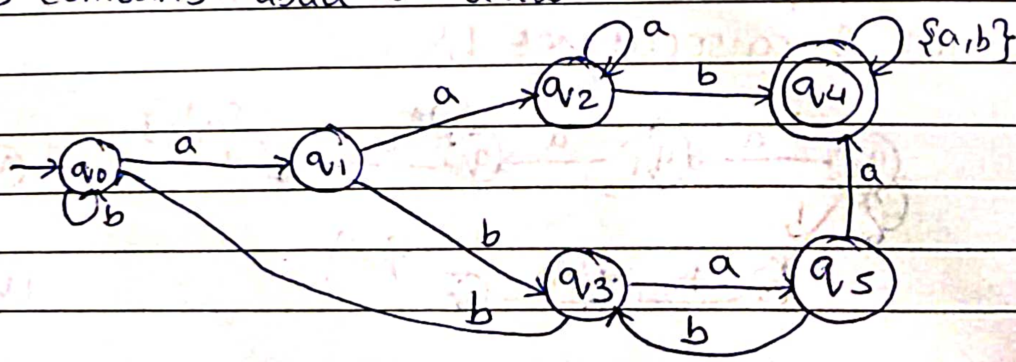
⇒ i/p starts with "aab" and ends with "bb"



⇒ i/p starts "aab" or ends "bb



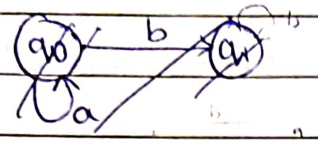
i/p contains "abaa" or "aabb"

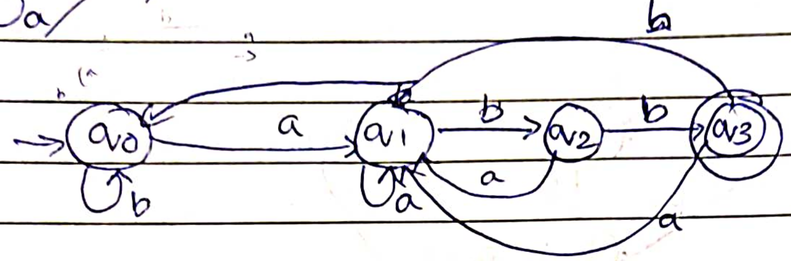


a aaba

Date: / /

Q ~~#~~ ends with "abb"





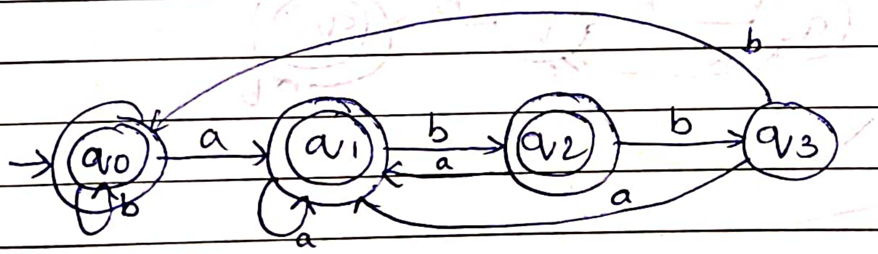
Q Design a DFA to accept input over $\Sigma = \{a,b\}$ that do not end with "abb".

$\Rightarrow$ ~~Comp~~

* Compliment of language ($\bar{L}$ or L')

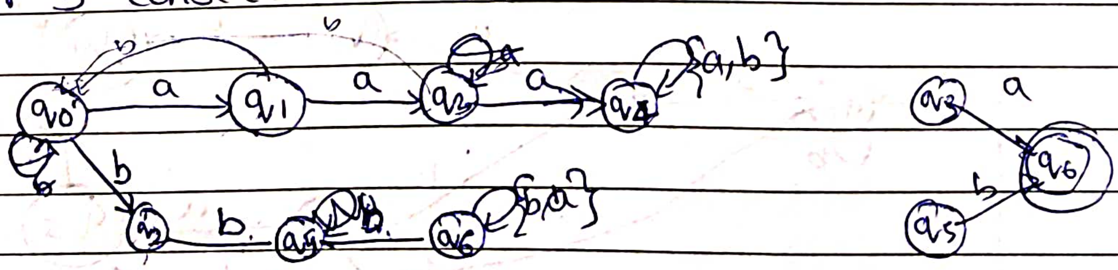
$\Rightarrow$ For every regular language L having DFA $M = (Q, \Sigma, \delta, q_0, F)$ there exist a DFA M' for language $\bar{L}$. such that $M' = (Q, \Sigma, \delta, q_0, F')$ where $F' = Q - F$

final $\rightarrow$ no final
nofinal $\rightarrow$ final

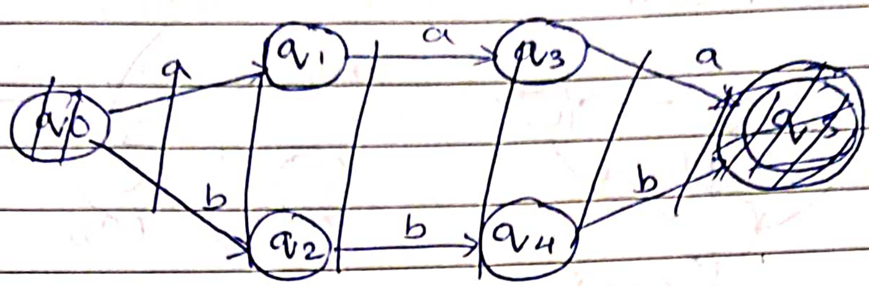


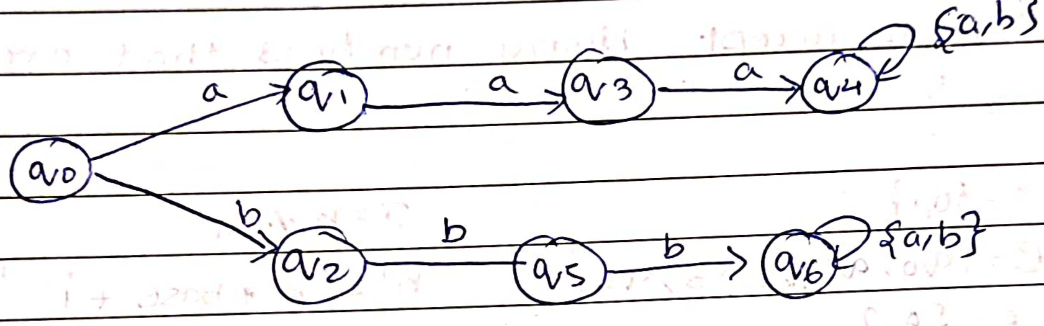
Q. DFA to accept string ~~over $\Sigma = \{$~~ that neither contain 3 conserutive a's nor 3 consecutives b's.

aabba

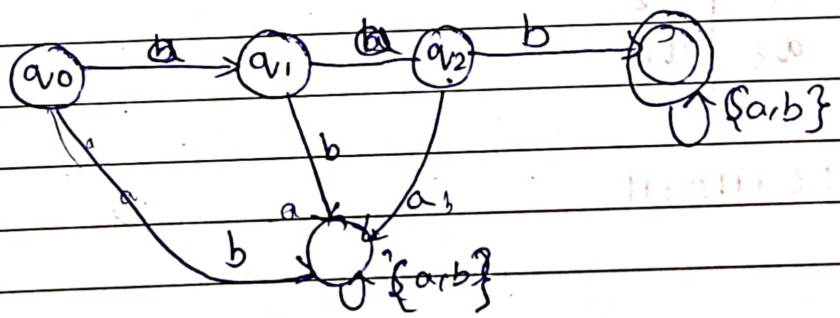


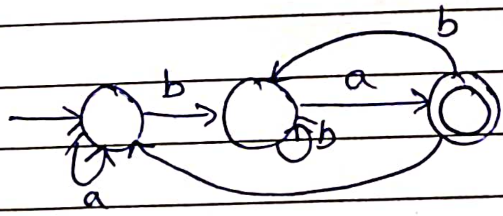
Date: / /





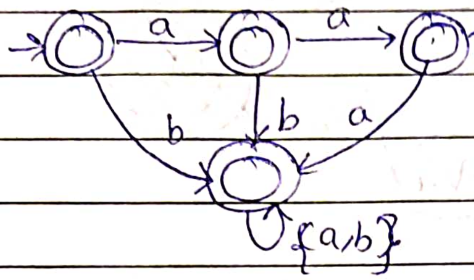
Q DFA to accept string that do not end with "ba" if the i/p starts with "aab" otherwise accepted.

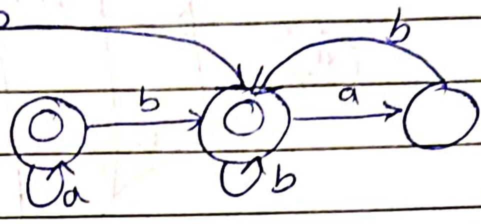




Date: / /

starts with aab





end with ba.

Q. DFA as Modulo machine

Q1. DFA to accept binary numbers that are divisible by 5.

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

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

$F = \{q_0\}$

$\delta = Q \times \Sigma \rightarrow Q$

$r = n \% m$

$n' = n * base + i$ (new input)

$r' = n' \% m$

$r' = (n * base + i) \% m$

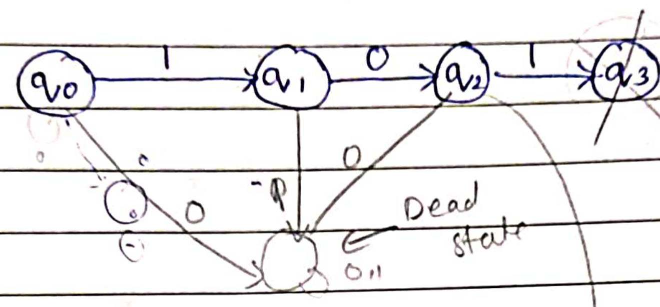
$r' = (r * base + i) \% m$

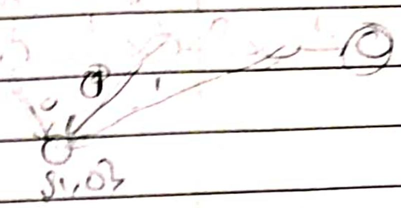
Q\Σ	0	1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_4	q_0
q_3	q_1	q_2
q_4	q_3	q_4

Eg 101110111

Date: 03/08/2023

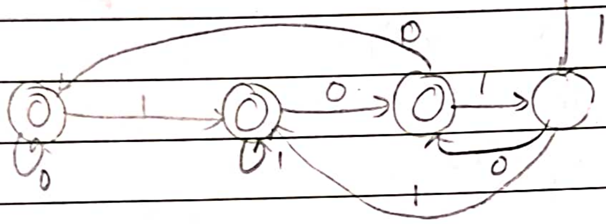
Q Design a DFA that accept string that starts with but not ends with '101'.





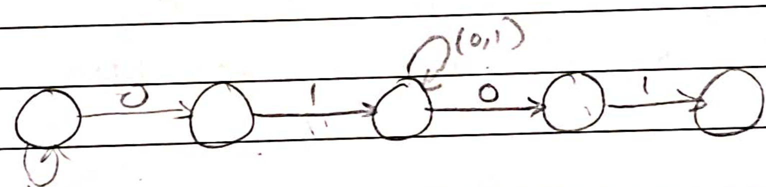
01 1 0 0

1 0



Q Design DFA that accepts ~~str~~ Binary number $\Sigma = \{0,1\}$ that contain '1' at every even position.

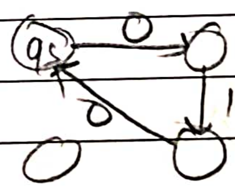
010101

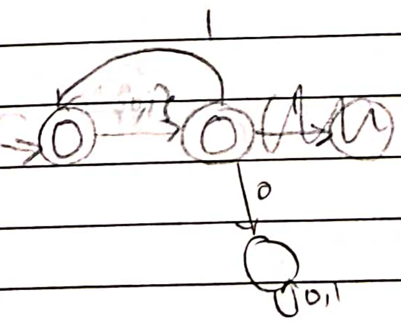


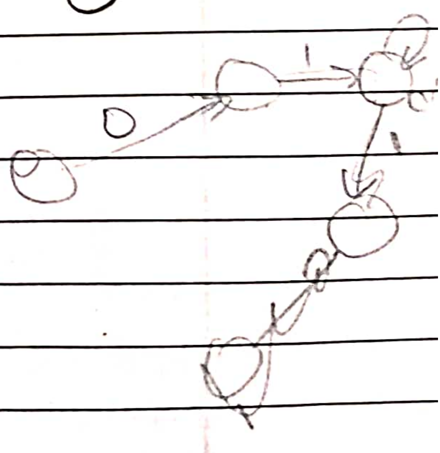
0 1 1 0 1 0

0

	0	1
q_s		
q_0		
q_1		

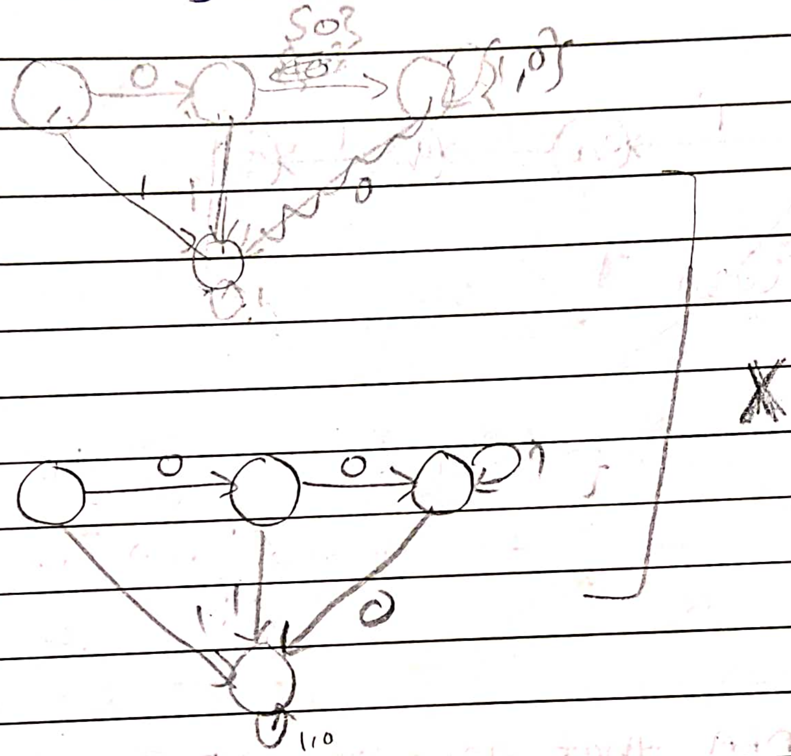


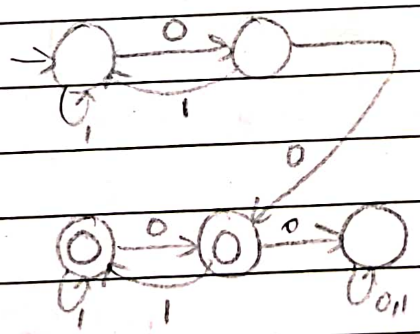




Date: / /

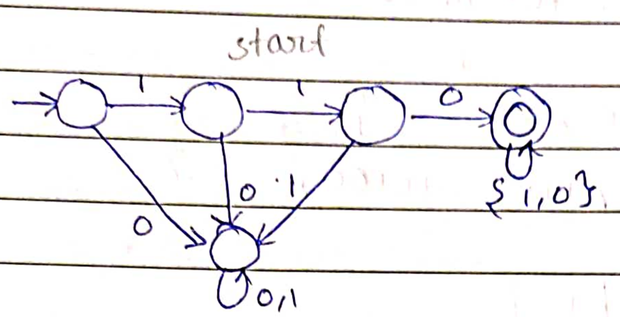
Q Design a DFA to accept string that contain '00' ~~a~~ exactly once.

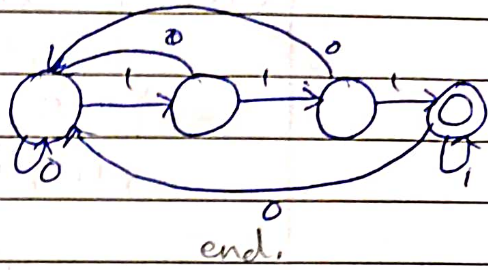


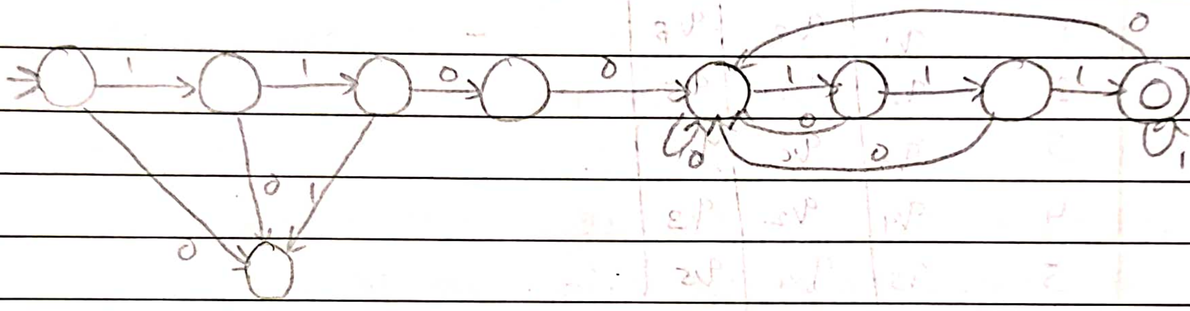


Date: 09/08/23

Q. Design a DFA to accept strings over {0,1} that start with "110" and ends with "111".

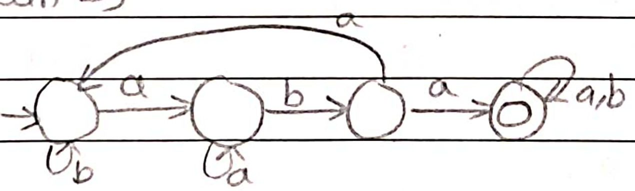




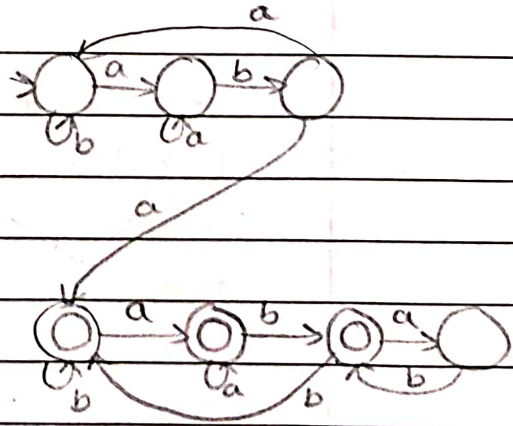


Q. Design a DFA to accept string over {a,b}, that contain but do not end with "aba".

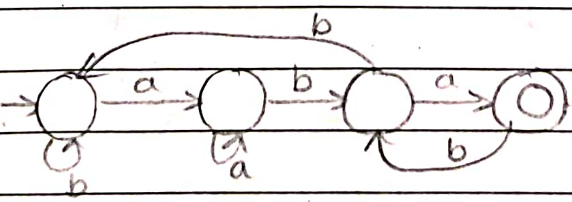
contain =>



Start = dead state



end



⇓

compliment

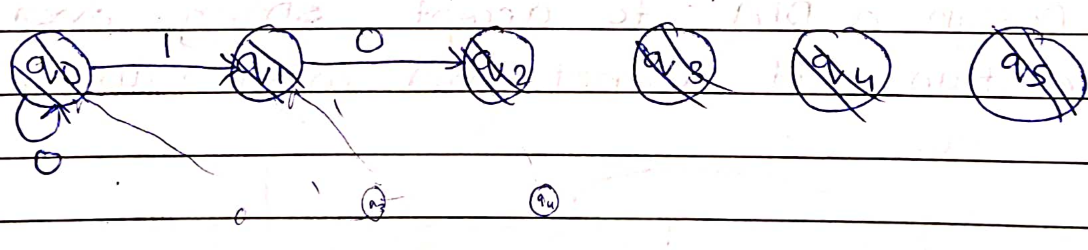
Date: / /

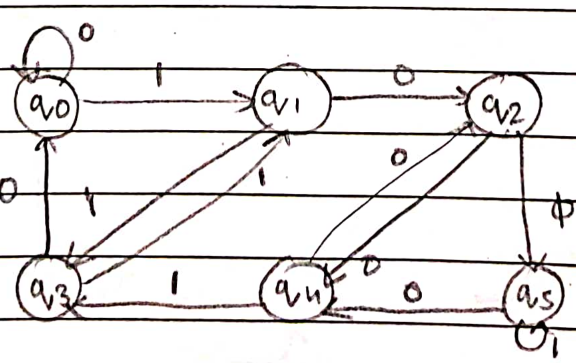
Q Design a DFA that accepts strings over {0,1} that represent set of binary no.s having values 6m+2. for $m \geq 0$

$\{2, 8, 14, 20, 26 \ldots\}$

$\{10, 1000, 1110, 10100, \ldots\}$

rem.		0	1
0	q_0	q_0	q_1
1	q_1	q_2	q_3
2	q_2	q_4	q_5
3	q_3	q_0	q_1
4	q_4	q_2	q_3
5	q_5	q_4	q_5





Date:

Tutorial-1 (next lec, next week)

Q1. Design a DFA for following set of string over set $\Sigma = \{0,1\}$

a) set of strings that contain "11" atmost ~~exactly~~ once

b) set of strings that it starts with "10" then ends with "101". otherwise ends with "00".

c) set of strings that have even ~~represent~~ length and even no. of 1's.

d) set of strings that have even no. of '0's. and no. of '1's. are multiple of 3.

Q2. $\Sigma = \{a, b, c\}$

a) strings that starts & ends with diff. symbols

b) contains atleast one occurence of double letter.

c) starts with but not ends with "bbc".

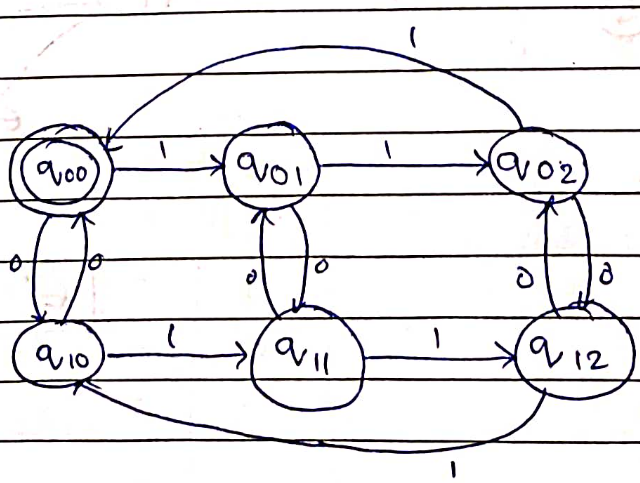
Q3. A DFA to accept octal numbers that are divisible by 5.

1.d) even no. of 0's & no. of 1's multiple of 3.

$\{0, 2, 4, 6, 8 \ldots\}$ | $\{0, 3, 6, 9, 12 \ldots\}$

0, 1 | 0, 1, 2

n%2 | n%3



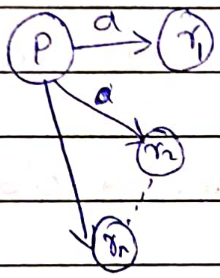
Date: / /

* NFA (Non-Deterministic Finite Automate)

DFA

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

where δ maps $Q \times \Sigma$ to 2^Q



$\delta(p, a) = \{r_1, r_2, r_3, \ldots, r_n\} = R$

$1 \le i \le m$

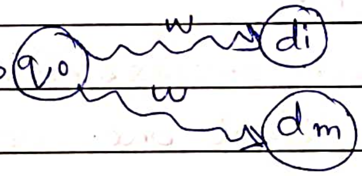
ek bhi accept toh accept

if $\exists\, d_i \in F$ then $w \in L(M)$ i.e. Accept

$\forall$ if $d_i \notin F$ then $w \notin L(M)$ i.e. reject.

sab reject to reject

for any $w \in \Sigma^* \rightarrow$



NFA

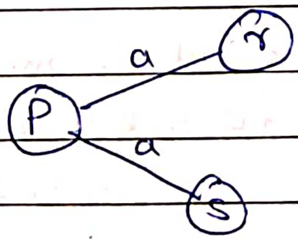
without ϵ

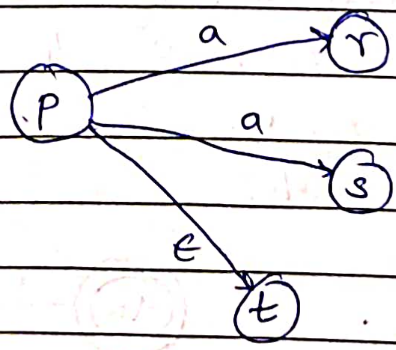
with ϵ

(ϵ stands for null string)

$\delta : Q \times \Sigma \rightarrow 2^Q$

$\delta : Q \times \Sigma \cup \{\epsilon\} \rightarrow 2^Q$

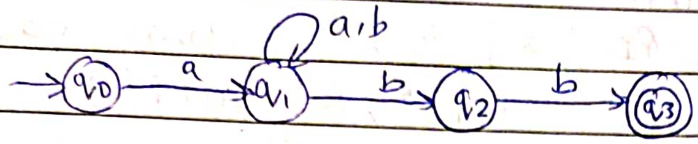




$\boxed{\epsilon^* a \epsilon^* \equiv a}$

Date: 09 / 08 / 2023

Example NFA without ϵ



$Q \backslash \Sigma$	a	b
q_0	q_1	ϕ
q_1	q_1	q_1, q_2
q_2	ϕ	q_3
q_3	ϕ	ϕ

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

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

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

$F = \{q_3\}$

Consider i/p string $w = ababb$

$\delta^*(\{q_0\} ababb)$ $\quad \because \delta(q_0, a) = \{q_1\}$

$\delta^*(\{q_1\} babb)$ $\quad \because \delta(q_1, b) = q_1, q_2$

$\delta^*(\{q_1, q_2\}, abb)$ $\quad \because \delta(q_1, a) = \{q_1\} \cup$

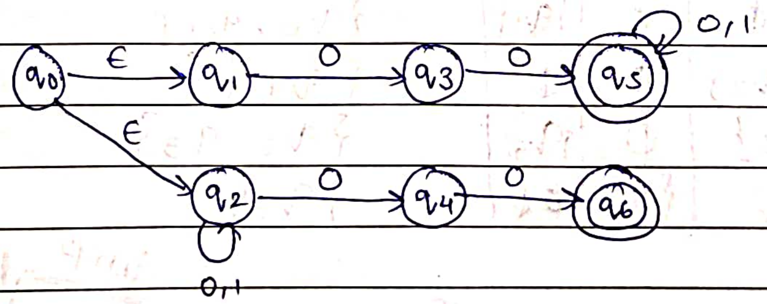
$\delta^*(\{q_1\}, bb)$ $\quad \delta(q_2, a) = \phi$

$\delta^*(\{q_1, q_2\}, b)$ $\quad = \{q_1, q_3\}$

$= q_3 \in F$

$\therefore ababb \in L$

Eg 2. NFA with ϵ



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

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$

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

$F = \{q_5, q_6\}$

$\delta : Q \times \Sigma \cup \{\epsilon\} \mapsto 2^Q$

$Q \backslash \Sigma$	0	1	ϵ
q_0	ϕ	ϕ	$\{q_1, q_2\}$
q_1	$\{q_3\}$	ϕ	ϕ
q_2	$\{q_2, q_4\}$	$\{q_2\}$	ϕ
q_3	$\{q_5\}$	ϕ	ϕ
q_4	$\{q_6\}$	ϕ	ϕ
q_5	$\{q_5\}$	$\{q_5\}$	ϕ
q_6	ϕ	ϕ	ϕ

Date: / /

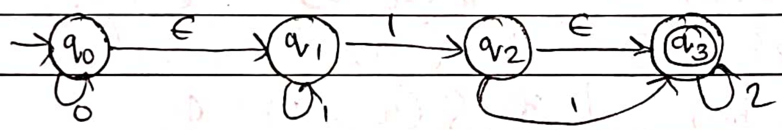
1 For every NFA with ϵ $M = (Q, \Sigma, \delta, q_0, F)$ where δ maps $Q \times \Sigma \cup \{\epsilon\}$ to 2^Q, there exists NFA without ϵ $M' = (Q, \Sigma^*, \delta', q_0, F')$ where δ' maps $Q' \times \Sigma$ to 2^Q such that

$$\boxed{\delta'(p, a) = \epsilon\text{-source}(\delta(\delta^*(p, \epsilon), a))}$$

$\delta^*(P, \epsilon)$ i.e ϵ-closure (P) is set of all states reachable from P over ϵ

and $F' = F \cup \{q_0\}$ if $\delta^*(q_0, \epsilon)$ contains $f \in F$

Example:



$M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2, q_3\}$; $\Sigma = \{0, 1, 2\}$
$F = \{q_3\}$ δ maps $Q \times \Sigma \cup \{\epsilon\}$ to 2^Q

with ϵ

$Q \backslash \Sigma$	0	1	2	ϵ	closure
q_0	$\{q_0\}$	ϕ	ϕ	$\{q_1\}$	$\{q_0, q_1\}$
q_1	ϕ	$\{q_1, q_2\}$	ϕ	ϕ	$\{q_1\}$
q_2	ϕ	$\{q_3\}$	ϕ	$\{q_3\}$	$\{q_2, q_3\}$
q_3	ϕ	ϕ	$\{q_3\}$	ϕ	$\{q_3\}$

Imp

without ϵ

δ' maps $Q \times \Sigma$ to 2^Q

$Q \backslash \Sigma$	0	1	2
q_0	$\{q_0, q_1\}$	$\{q_1, q_2, q_3\}$	ϕ
q_1	ϕ	$\{q_1, q_2, q_3\}$	ϕ
q_2	ϕ	$\{q_3\}$	$\{q_3\}$
q_3	ϕ	ϕ	$\{q_3\}$

$\delta^*(q_0)$ 1) q_0 ka closure $\Rightarrow q_0, q_1$
then q_0 & q_1 ka transition on 0
dono ka union $q_0 \cup \phi \Rightarrow q_0$
then q_0 ka closure $\Rightarrow \{q_0, q_1\}$

q_0

Phele closure + transition on that symbol $\rightarrow$ then union then uska closure.

Date: / /

2. For every NFA w/o ϵ $m = (Q, \Sigma, \delta, q_0, F)$ where δ maps $Q \times \Sigma$ to 2^Q (power set), there exists a DFA.
$m' = (Q', \Sigma, \delta', q_0', F')$ where $Q' = 2^Q$ & δ' maps $Q' \times \Sigma$ to Q' such that $\forall\ p \in 2^Q$ $[P] \in Q'$

$\forall\ p \in Q$ and $a \in \Sigma$ $\delta'([P], a) = [\delta(P, a)]$

$\forall [P, q] \in Q'$ and $a \in \Sigma$ $\delta'([P, q], a) = [\delta(P, a) \cup \delta(q, a)]$

$F' = \{[f] \mid [f]$ contains $m \in F\}$ and $q'_0 = [q_0]$

Eg.

$m = (Q, \Sigma, \delta, q_0, F)$ $Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$ $F = \{q_3\}$
δ maps $Q \times \Sigma$ to 2^Q.

$Q \backslash \Sigma$	a	b
q_0	q_1	ϕ
q_1	$\{q_1, q_2\}$	$\{q_1, q_2\}$
q_2	ϕ	ϕ

$Q \backslash \Sigma$	a	b
q_0	$\{q_1\}$	ϕ
q_1	$\{q_1\}$	$\{q_1, q_2\}$
q_2	ϕ	ϕ

Equivalent DFA
$m' = (Q', \Sigma, \delta', q_0', F')$
$Q' = 2^Q = \{[\phi], [q_0], [q_1], [q_2], [q_0, q_1], [q_0 q_2], [q_1 q_2], [q_0, q_1, q_2]\}$

$q'_0 = [q_0]$

$F' = \{[q_2], [q_0, q_2], [q_1, q_2], [q_0, q_1, q_2]\}$
$\delta' : Q' \times \Sigma \to Q'$

Date: / /

$Q\backslash\Sigma$	a	b
$[q_0]$	ϕ	ϕ
$[q_1]$	q_1	ϕ
$[q_2]$	ϕ	ϕ
$[q_0, q_1]$	q_1	ϕ
$[q_1, q_2]$	q_1	0
$[q_0, q_2]$		
$[q_0, q_1, q_2]$		

$Q\backslash\Sigma$	a	b
$[\phi]$	$[\phi]$	$[\phi]$
$[q_0]$	$[q_1]$	$[\phi]$ ar
$[q_1]$	$[q_1]$	$[q_1, q_2]$
$[q_2]$	$[\phi]$	$[\phi]$
$[q_0, q_1]$	$[q_1]$	$[q_1, q_2]$
$[q_1, q_2]$	$[q_1]$	$[q_1, q_2]$
$[q_0, q_2]$	$[q_1]$	$[\phi]$
$[q_0, q_1, q_2]$	$[q_1]$	$[q_1, q_2]$

abb

23/08/2023

garbage collecting theory

DFA minimisation : to reduce steps

step 1: eliminate ~~all~~ states that are not reachable from start state.

step 2: minimize by classical method./ apprach.

step 3, Equivalence class method.
(based on my hill Nerode theorem)

$p \equiv q$
if $((p \in F \wedge q \in F)$ or $(p \notin F \wedge q \notin F))$
every symbol $\Rightarrow$ and $\forall a \in \Sigma \; \delta(p,a) = \delta(q,a)$

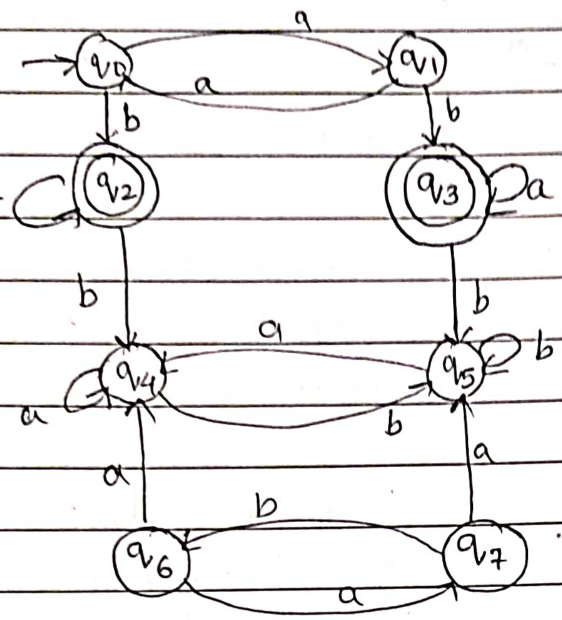
$\Downarrow$ Negation

either
$p \not\equiv q$ if $\wedge(p \in F \wedge q \notin F)$ or $\exists a \in \Sigma \; \delta(p,a) = r$
$\delta(q,a) = s$

~~rs~~ & $\boxed{r \not\equiv s}$

Date: / /

Example



Step 1. Minimalisation

$Q \backslash \Sigma$	a	b
q_0	q_1	q_2
q_1	q_0	q_3
* q_2	q_2	q_4
* q_3	q_3	q_5
q_4	q_4	q_5
q_5	q_4	q_5

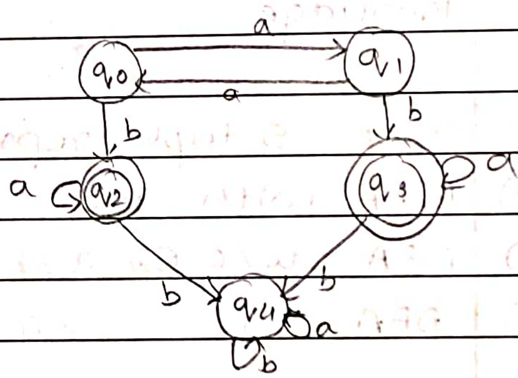
} identical row

Step 2: by classical apprach $q_4 \equiv q_5$

∴ discard q_5 and replace transition to q_5 by q_4.

(higher no. of discard karo)

$Q \backslash \Sigma$	a	b
q_0	q_1	q_2
q_1	q_0	q_3
* q_2	q_2	q_4
* q_3	q_3	q_4 $\cancel{q_5}$
q_4	q_4	q_4 $\cancel{q_5}$



Step 3: $\{q_0, q_1, q_2, q_3, q_4\}$

$\in F$ → $\{q_0, q_1, q_4\}$ $\qquad$ $\notin F$ → $\{q_2, q_3\}$

$\{q_0, q_1\}$ $\quad$ $\{q_4\}$ $\qquad$ $\{q_2, q_3\}$

$\{q_0, q_1\}$ $\quad$ $\{q_4\}$ $\qquad$ $\{q_2, q_3\}$

$\equiv$ $\qquad$ $\equiv$

$\left.\begin{matrix} \delta(q_0, a) = q_1 \\ \delta(q_1, a) = q_0 \end{matrix}\right\} =$

$\left.\begin{matrix} \delta(q_0, b) = q_2 \\ \delta(q_1, b) = q_3 \end{matrix}\right\} =$

$\left.\begin{matrix} \delta(q_0, a) = q_1 \\ \delta(q_4, a) = q_4 \end{matrix}\right\} =$

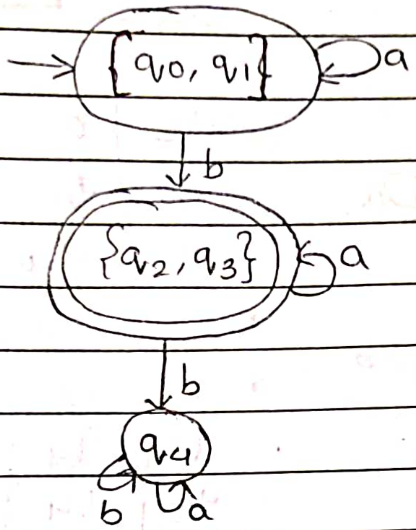
$\left.\begin{matrix} \delta(q_0, b) = q_2 \\ \delta(q_4, b) = q_4 \end{matrix}\right\} \neq$

$\{q_4\}$ alga

$\left.\begin{matrix} \delta(q_2, a) = q_2 \\ \delta(q_3, a) = q_3 \end{matrix}\right\} =$

$\left.\begin{matrix} \delta(q_2, b) = q_4 \\ \delta(q_3, b) = q_4 \end{matrix}\right\} \equiv$

Date: / /



40 Mks MSE

16 mks ✱ Assigment 1

Q1. Define following
(a) alphabet
(b) string
(c) grammar
(d) language

Q2. Give 5 tuple representation of
(a) NFA with ϵ.
(b) NFA w/o ϵ.
(c) DFA

Q3. Prove equivalence of NFA with ϵ. & NFA w/o ϵ.
(ϵ is null string).

without ϵ. default.

Q4. Prove equivilene of NFA & DFA.

Q5. Explain steps of minimization of DFA.

Date: / /

Tutorial-2

NFA and its equivalence

1) Construct NFA for following languages over the set $\{0,1\}$

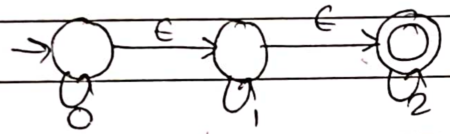
a) set of string that start with 00 and contain "110" to be accept.

b) set of string that contain 000 and 111

2) Construct NFA to accept string over $\Sigma = \{a,b,c\}$ that start and ends with diff. symbol.

3) Convert following NFA with ϵ to NFA w/o E.

a)



b) $m = \{\{p,q,r,s\}, \{0,1\}, \delta, p, \{r,s\}\}$

δ:

	0	1	ϵ
p	$\{r\}$	ϕ	$\{q\}$
q	$\{q\}$	$\{s\}$	$\{p\}$
r	$\{r\}$	$\{r\}$	$\{\phi\}$
s	$\{q\}$	$\{s\}$	$\{q\}$

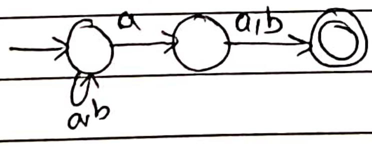
m: {

Tutorial-3

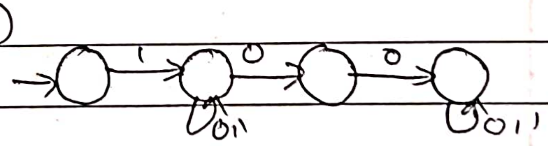
Equivalence of NFA & DFA & DFA minimization

Q1) Convert the following NFA's to DFA's.

(i)

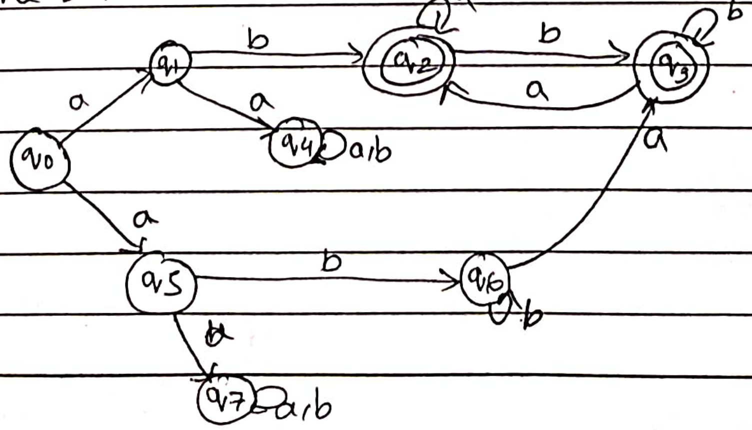


(ii)



Q2) Minimize the DFA.

(i)



(ii)	Q\Σ	0	1
	a	a	b
	b	a	c
	c	d	c
*	d	a	b
	e	b	g
	f	e	f
*	g	a	b

Final state