1 2 3

Formal Languages and Automata Theory

aa	bb	abb	abb	aa	bb	aa
aab	ba	b	b	aab	ba	aab

1	2	1	3
aa	bb	aa	abb
aab	ba	aab	b

(undecidable)

Matrix mortality problem-
{M1, M2, M3 . . . }
product is zero.
undecided.

Finite automata - regular languages
- lambda calculus was used by church to prove that Entscheidungs problem is undecidable.

compiler - ① Lexical analyser - checks symbols
finite automata can detect lexical problems. (regular language)
② syntax analysis - can be detected by push down automata.
(context free languages)

Turing machine can recognise recursively enumerable languages.

Alphabet

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

Strings

$w = 0110$

$w = 10110$

Language - set of ~~valid~~ strings all of which are chosen from Σ where Σ is the alphabet of language.

$L = \{w \mid w \text{ begins with } 0\}$

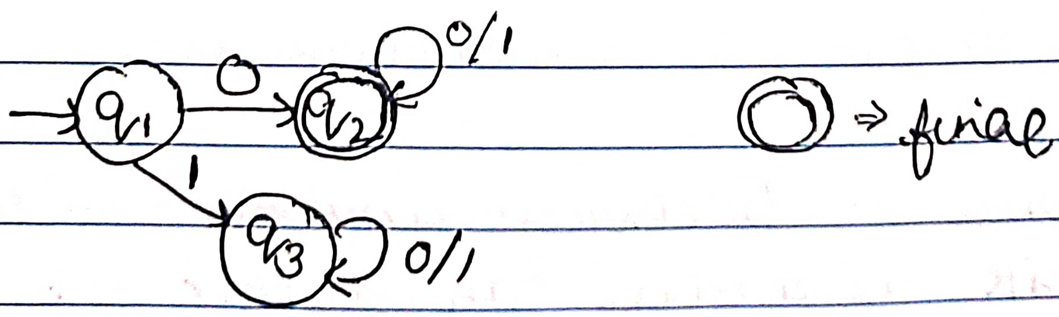
p vs np

↙ ↘

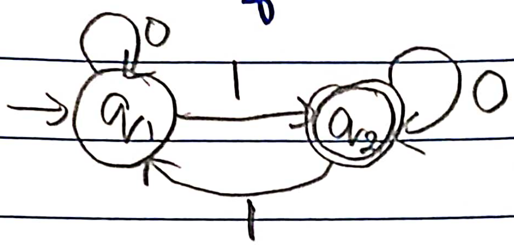
polynomial time

non deterministic polynomial time

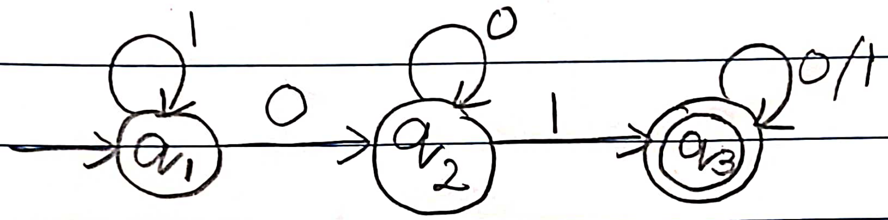
L :



Q. Design a FA that detects an Odd no. of ones.



Q. DFA that accepts strings that have '01' somewhere.



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

Q → set of states

Σ → alphabet

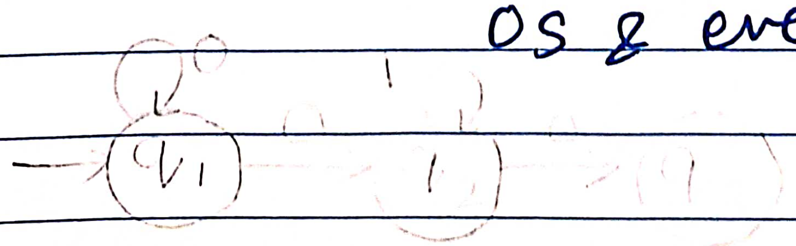
δ → set of transitions

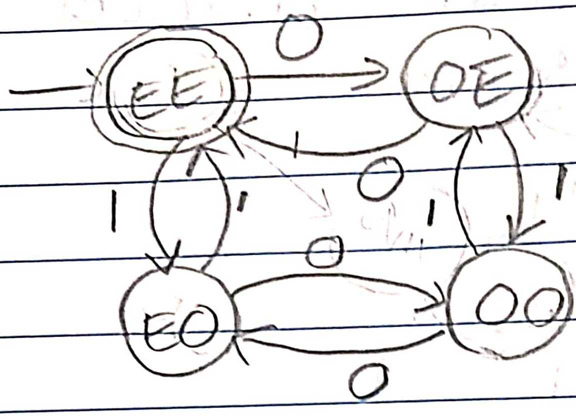
q_0 → initial state

F → set of final states

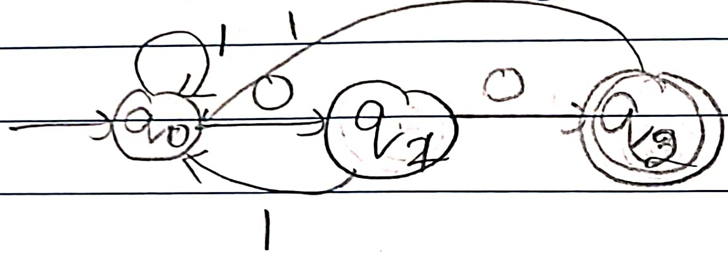
$\delta(q_1, 0) = q_1$

Q. DFA that accepts $L = \{w$ has even no. of 0s & even no. of 1s$\}$

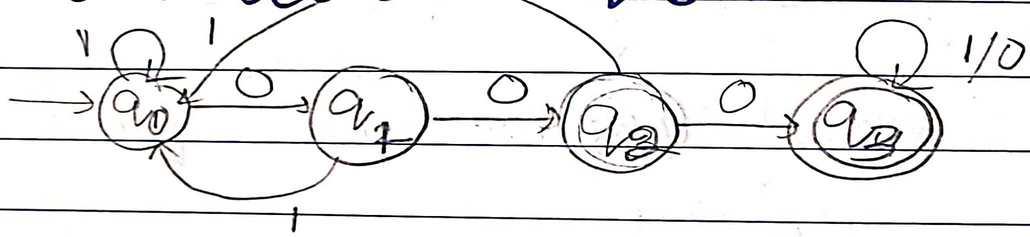




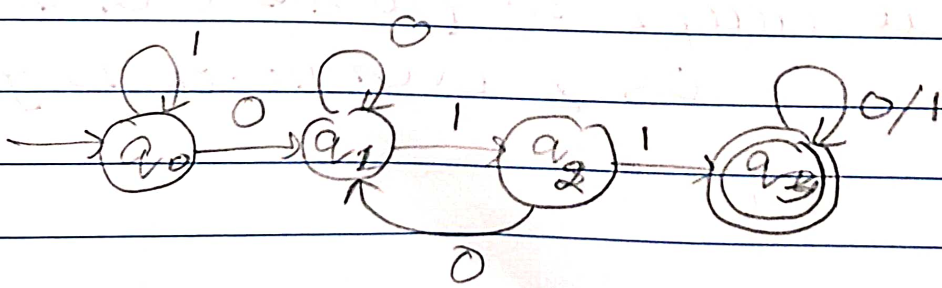
Q/ strings ending in '00'



Q/ 3 consecutive zeroes.

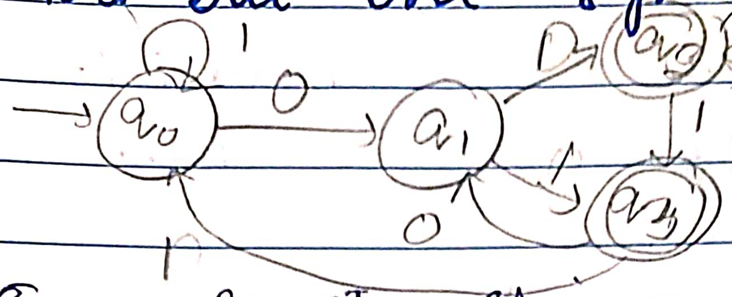


Q/ 011 substring



$\hat{\delta}(q_0, 0111) = \hat{\delta}(\delta(q_0, 0), 111) = \hat{\delta}(\delta(q_1, 1), 11)$
$= \hat{\delta}(\delta(q_3, 1), 1)$

Q/ last but one symbol is zero.



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

24/1

ϵ - empty string

$|w|$ - length of string

Σ^1 - set of all strings of length 1 using alphabet Σ.

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

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

$\Sigma^2 = \{01, 10, 00, 11\}$

$\Sigma^* =$ set of all strings using Σ.

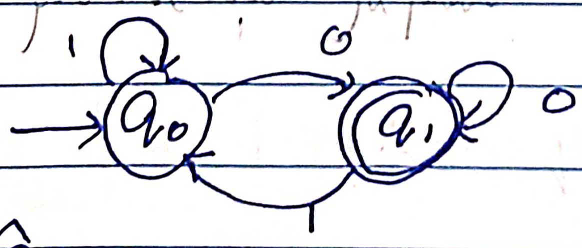
$\Sigma^+ \rightarrow$ excludes ϵ

$\Sigma^+ = \Sigma^* - \epsilon$

ϵ : identity (concatenation with ϵ gives original)

ϕ - empty language

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



$\hat{\delta}(q_0, 100) = q_1$

	0	1
$\rightarrow q_0$	q_1	q_0
$* q_1$	q_1	q_0

Transition table

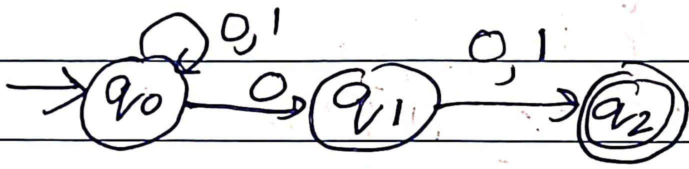
Q. $A = (Q, \Sigma, \delta, q_0, F)$

$L(A) = \{ w \mid \hat{\delta}(q_0, w) \text{ is in } F \}$

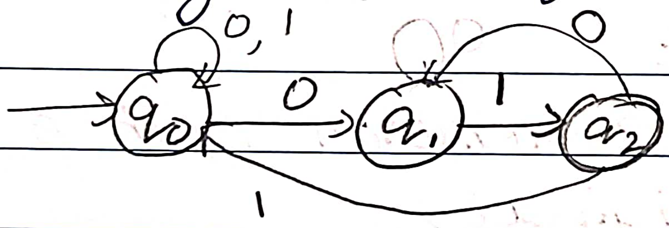
25/1

NFA →

last but one is 0



Q. strings that end with 01



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

DFA vs NFA → different transition functions

one transition destination

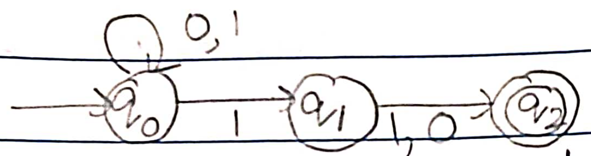
~~set of~~ multiple

NFA →

$$L(A) = \{ w \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset \}$$

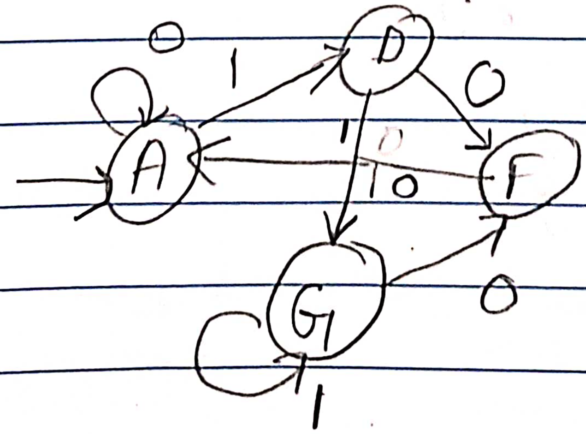
NFA - Subset construction

Q. 1 at second last position



	0	1
ϕ	ϕ	ϕ
$\rightarrow \{q_0\}$ A	$\{q_0\}$ A	$\{q_0, q_1\}$ D
$\{q_1\}$ B	$\{q_2\}$	$\{q_2\}$
* $\{q_2\}$ C	ϕ	ϕ
$\{q_0, q_1\}$ D	$\{q_0, q_2\}$ F	$\{q_0, q_1, q_2\}$ G
* $\{q_1, q_2\}$ E	$\{q_2, \phi\}$ C	$\{q_2, \phi\}$ C
* $\{q_0, q_2\}$ F	$\{q_0, \phi\}$ A	$\{q_0, q_1, \phi\}$ D
* $\{q_0, q_1, q_2\}$ G	$\{q_0, q_2, \phi\}$ F	$\{q_0, q_1, q_2, \phi\}$ G

	0	1
$\rightarrow$ A	A	D
D	F	G
* F	A	D
* G	F	G

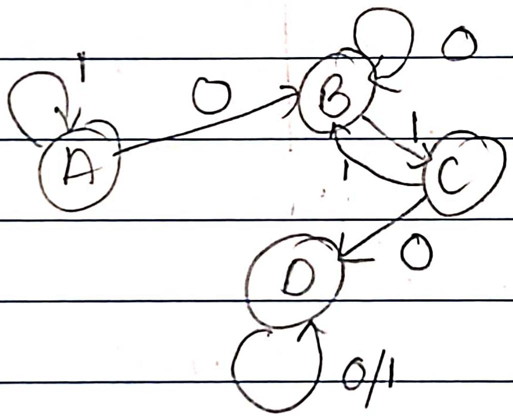


Q

δ	0	1
→ p	{p,q}	{p}
q	ϕ	{r}
* r	{p,r}	{q}

Find DFA by subset construction.

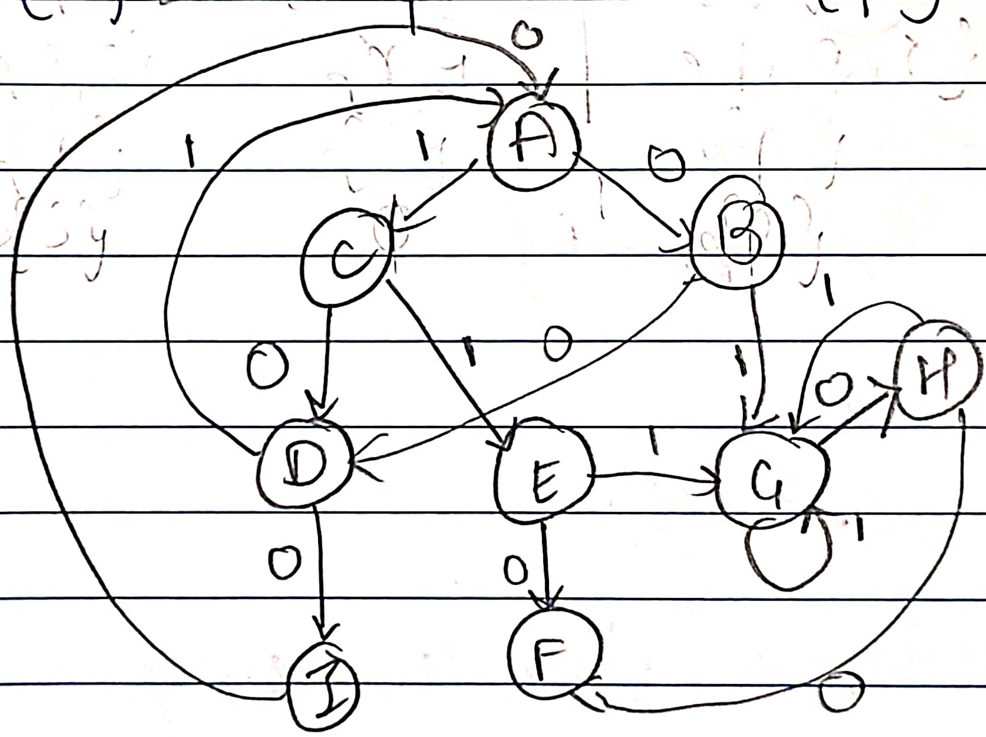
δ	0	1
→ p A	{p, q} B	{p} A
{p, q} B	{p, q} B	{p, r} C
* {p, r} C	{p, q, r} D	{p, q} B
* {p, q, r} D	{p, q, r} D	{p, q, r} D



Q

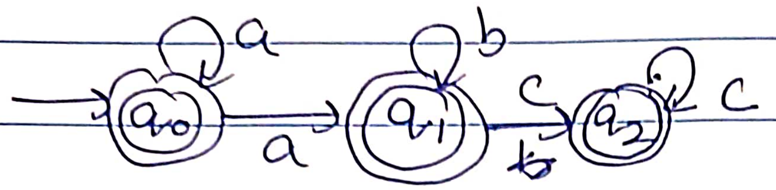
	0	1
→ p	{q, s}	{q}
* q	{r}	{q, r}
r	{s}	{p}
* s	ϕ	{p}

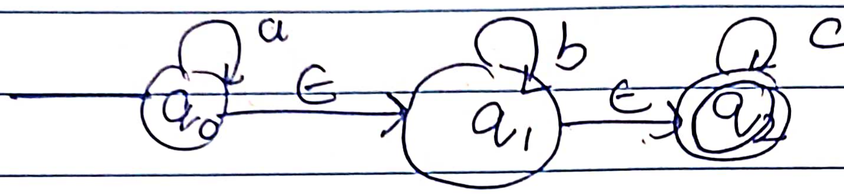
	0	1
→p A	{q, s} B	{q} C
*{q, s} B	{r} D	{p, q, r} G
*{q} C	{r} D	{q, r} E
{r} D	{s} I	{p} A
*{r, q} E	{r, s} F	{p, q, r} G
*{r, s} F	{s} I	{p} A
*{p, q, r} G	{q, r, s} H	{p, q, r} G
~~{s}~~	∅	~~∅r~~
*{q, r, s} H	{r, s} F	{p, q, r} G
* {s} I	∅	{p} A



Q. Draw NFA ~~where~~ for lang

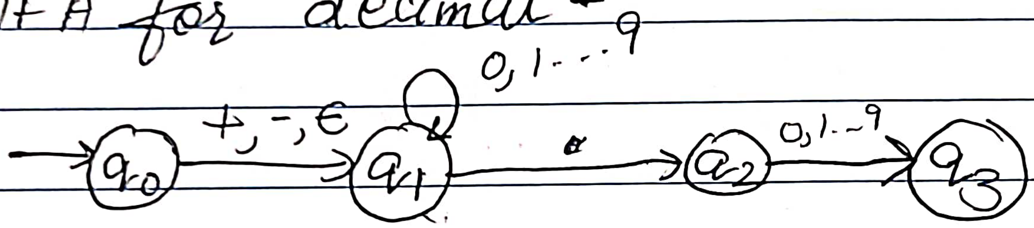
≥ 0 a's + ≥ 0 b's + ≥ 0 c's





Epsilon NFA

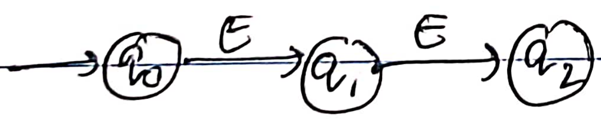
NFA for decimal -



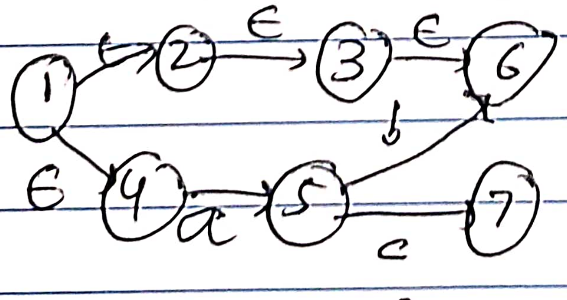
Epsilon NFA -

$A = \{Q, \Sigma, \delta, q_0, F\}$

Epsilon closure -



$ECLOSE(q_0) = \{q_0, q_1, q_2\}$



$ECLOSE(1) = \{1, 2, 3, 6\}, \{4\}$

Q

NFA

	ϵ	a	b	c
$\rightarrow p$	ϕ	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	ϕ
$* r$	$\{q\}$	$\{r\}$	ϕ	$\{p\}$

ECLOSE$(p) = \{p\}$

ECLOSE$(q) = \{q, p\}$

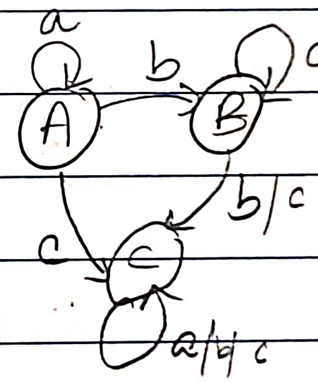
ECLOSE$(r) = \{r, q, p\}$

$F_D = \{S : S$ is in Q_D and $S \cap F_E \neq \phi\}$

q_0 of DFA = ECLOSE(p) = $\{p\}$

ϵ

	a	b	c
$\rightarrow \{p\}^A$	$\{p\}^A$	ECLOSE(q) $\{p, q\}^B$	$\{p, q, r\}^C$
$\{p, q\}^B$	$\{p, q\}^B$	$\{p, q, r\}^C$	$\{p, q, r\}^C$
$* \{p, q, r\}^C$	$\{p, q, r\}^C$	$\{p, q, r\}^C$	$\{p, q, r\}^C$



Q $\delta_D(S, a)$

a) $S = \{P_1, P_2 \cdots P_k\} = -$

b) compile $\bigcup_{i=1}^{k} \delta_E(p_i, a)$

Q	ϵ	a	b	c
$\rightarrow$ p	{q,r}	ϕ	{q}	{r}
q	ϕ	{p}	{r}	{p,q}
* r	ϕ	ϕ	ϕ	ϕ

	a	b	c
$\rightarrow$ * {p,q,r}	{p,q,r}	{q,r}	{p,q,r}
* {q,r}	{p,q,r}	{r}	{p,q,r}
* {r}	ϕ	ϕ	ϕ
ϕ	ϕ	ϕ	ϕ

$\rightarrow$ $L = \{01, 10\}$

$L^* = \{01, 10, \epsilon, 0110, 0101, 1010, 1001, 0101 \ldots\}$

Kleene closure

$\rightarrow$ $L_1 = \{01, 10\}$ $L_2 = \{0\}$

$L_1 . L_2 = \{010, 100\}$ (concatenation)

$L_1 \cup L_2 = \{01, 10, 0\}$ (union)

* closure

. concatenation

\+ union

$L = \{ w \mid w$ has alternating 0s and 1s $\}$

$$(01)^* + (10)^* + 0(10)^* + 1(01)^*$$

$L = \{ w \mid w$ ends with 1 $\}$

$$(0^* 1^*)^* 1 \quad \text{or} \quad (1+0)^* 1$$

$L = \{ w \mid w$ starts with 110 $\}$

$$110(0+1)^*$$

$L = \{ w \mid w$ has equal 0's and 1's and they alternate $\}$

$$(01)^* + (10)^*$$

$L = \{ w \mid$ no. of 0's are even but contain 00s $)$

$$\cancel{\{} 1^* (00)^* 1^* \cancel{\}^*}$$

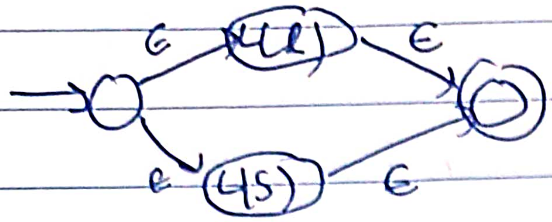
$L = \{ w \mid$ 2nd last is zero $\}$

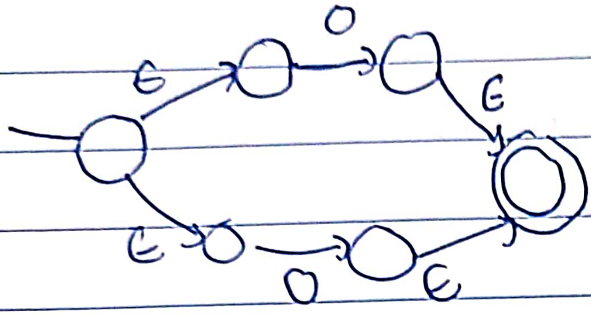
$$(1+0)^* \, 0 \, (0+1)$$

$$
\begin{tikzcd}
\text{NFA} \arrow[dr, "\text{subset construction}"] & & \\
\text{E.NFA} \arrow[r] & \text{DFA} \arrow[r, "\text{minimisation}"] \arrow[dl, bend left, "\text{Arden's theorem}"] & \text{minimised DFA} \\
\text{RE} \arrow[u] & &
\end{tikzcd}
$$

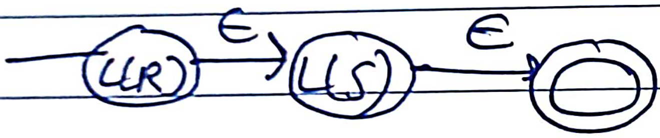
<u>Thompson's construction</u>

① $L(R) \cup L(S)$

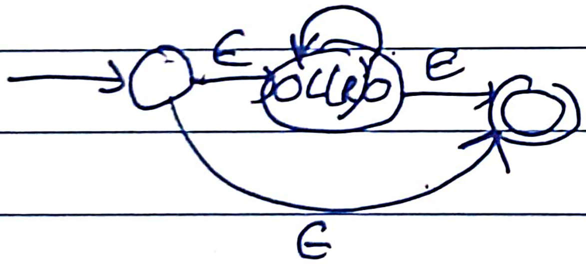




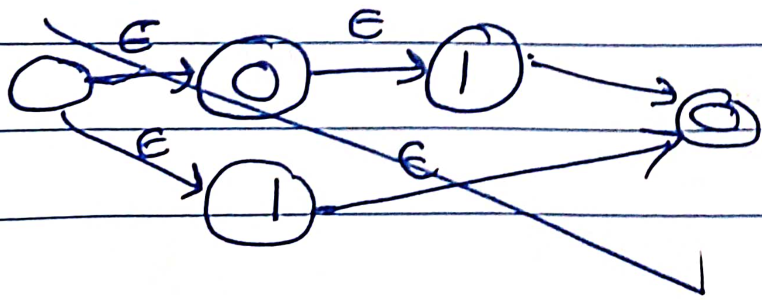
2) $L(R) \cdot L(S)$

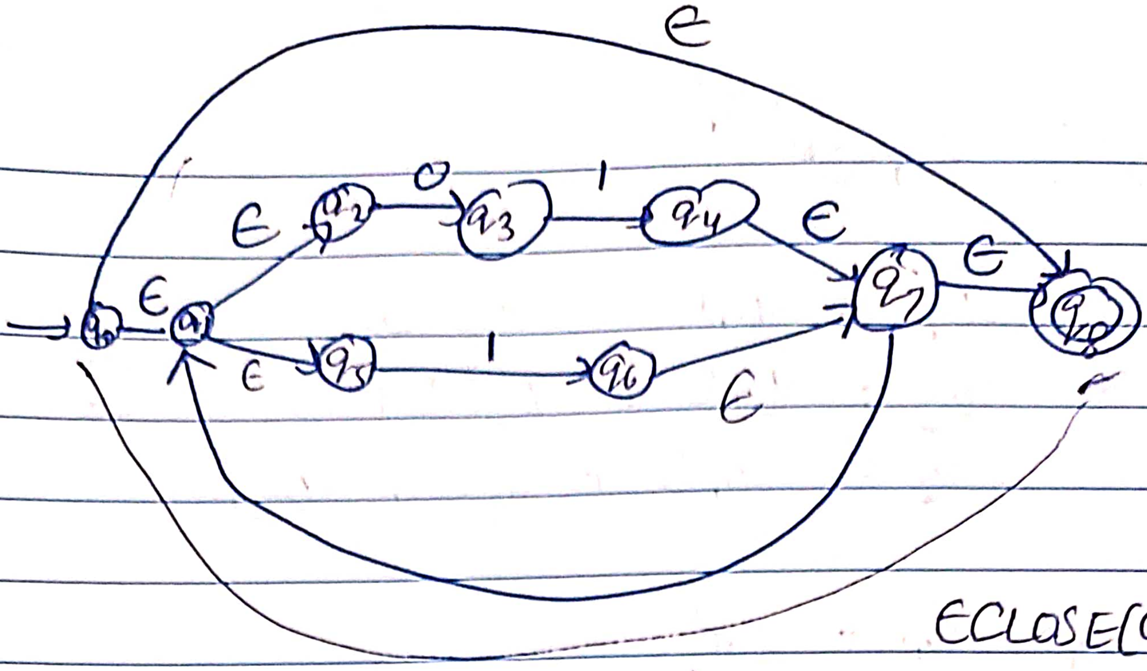


3) $(L(R))^*$



4) $(01+1)^*$





ECLOSE(q_0)

	0	1
$\{q_0, q_1, q_2, q_5, q_8\}$	$\{q_3\}$	$\{q_6, q_7, q_8, q_1, q_2, q_5\}$
$\{q_3\}$	ϕ	$\{q_4, q_7, q_8, q_1, q_2, q_5\}$

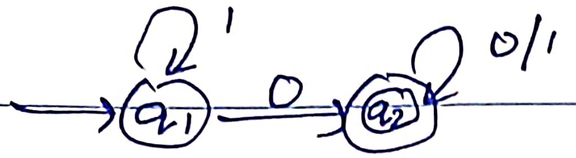
Arden's Theorem - If P and Q are 2 regular expressions over Σ and if P does not contain ϵ, then the following equation in R

$R = Q + RP$ has unique soln.

i.e., $R = QP^*$

$R = Q + (QP^*)P$

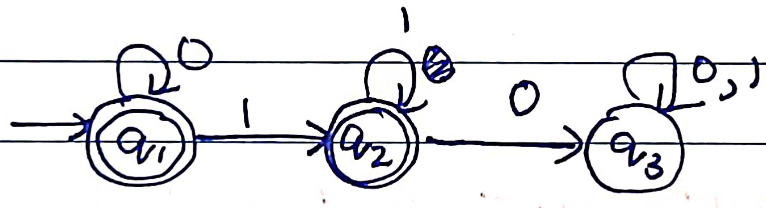
$= Q(\epsilon + P^*P) = QP^*$



$$q_1 = q_1 1 \quad — (1)$$
$$q_2 = q_1 0 + q_2(0+1) \quad — (2)$$
$$q_2 = q_1 0 + q_2 0 + q_2 1$$

$$q_1 = q_1 1 + \epsilon$$
$$R = RP + Q$$
$$R = QP^* = 1^*$$
$$q_2 = 1^*0 + q_2(0+1)$$
$$R \qquad Q + RP$$
$$q_2 = 1^*0(0+1)^*$$

Q//



$$q_1 = q_1 0$$
$$q_2 = q_1 1 + q_2 1$$
$$q_3 = q_2 0 + q_3(0+1)$$

$$q_1 = q_1 0 + \epsilon$$
$$R = RP + Q$$
$$R = QP^* = 0^*$$

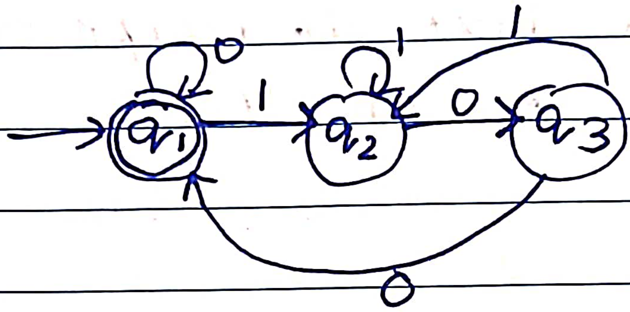
$$q_2 = 0^* 1 + q_2 1$$
$$R = Q + RP$$
$$R = QP^* = 0^* 1 1^*$$

$$q_3 = 0^*11^*0 + q_3(0+1)$$
$$R = Q + R\,P$$
$$R = QP^* = 0^*11^*0(0+1)^*$$

RE = union of final states

$$= 0^* + 0^*11^*$$

Q//



$$q_1 = q_1 0 + q_3 0$$
$$q_2 = q_1 1 + q_2 1 + q_3 1$$
$$q_3 = q_2 0$$

$$q_2 = q_1 1 + q_2 1 + q_2 01$$
$$R = Q + RP \qquad q_2(\underbrace{1+01}_{P})$$
$$q_2 = R = q_1 1(1+01)^*$$

$q_3 = q_2 0$, $q_1 = q_3 0 = q_2 00$

$$q_1 = q_1 0 + q_1 1(1+01)^* 00$$
$$= q_1(0 + 1(1+01)^* 00) + \epsilon$$
$$R = RP + Q$$

$$(0 + 1(1+01)^* 00)^*$$

Q. $(01+1)^*$

$R = QP^*$

$= \epsilon(01+1)^*$

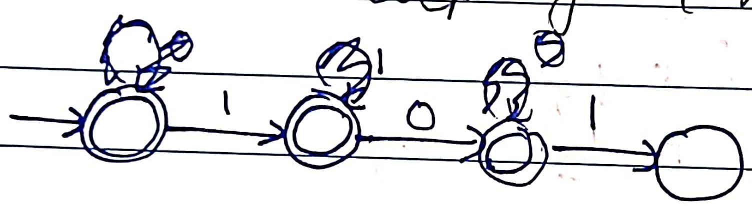
$R = RP + Q$

$= R(01+1) + \epsilon$

Q. Design DFA that ~~do~~ accepts strings which do not contain 101.

draw DFA which accepts 101

Invert accepting & non accepting

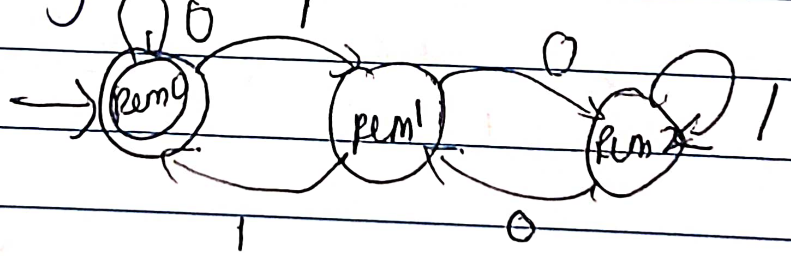


Q. DFA which accept finite nos. divisible by 3.

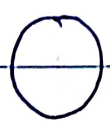
011

110

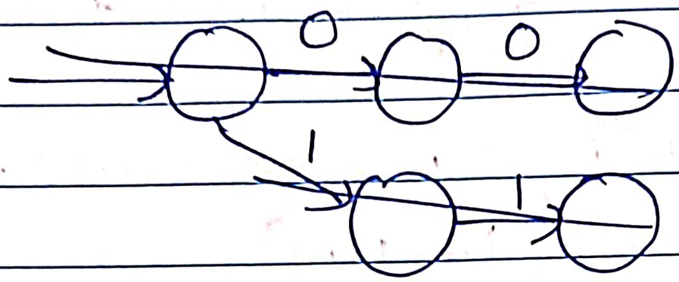
001

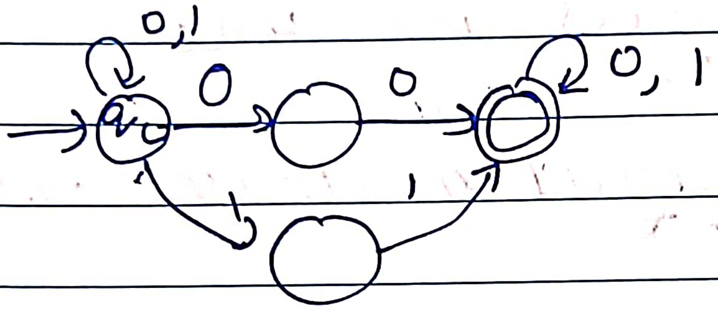


Q. $L(M) = \{0^m 1^n \mid m\ n \geq 0\}$

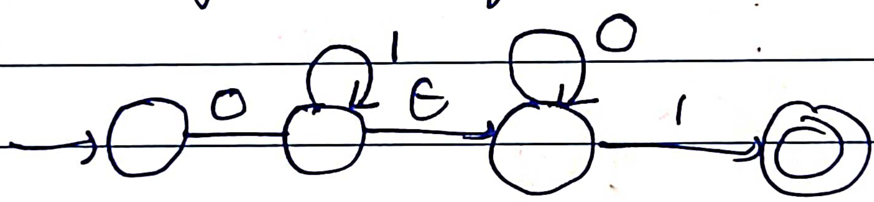


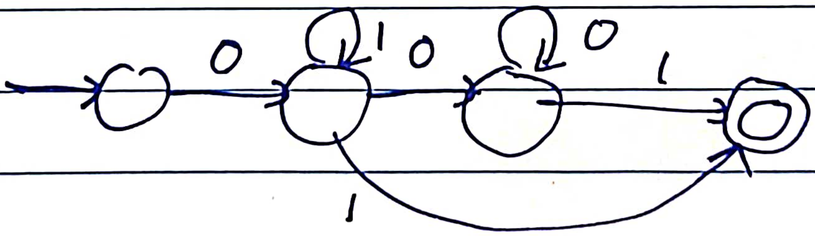
Q. NFA for strings having 2 consecutive 0's or 1's.



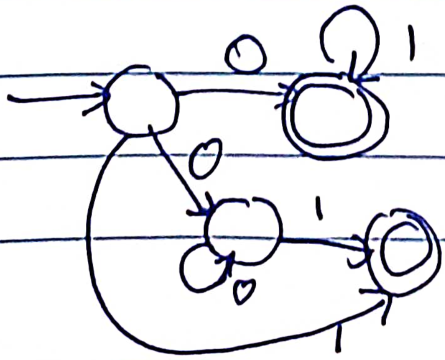


Q. Design NFA for 01^*0^*1





Q. $01^* + 0^*1$



Regular Exp

Q// all string that begin with zero and end with 1. $0(0+1)^*1$

Q// strings having odd no. of ones
$\{1\}$ $1(11)^*$

Q beginning with 0, no 2 consecutive zeroes
$(0+1)(1+10)^*$ (10)

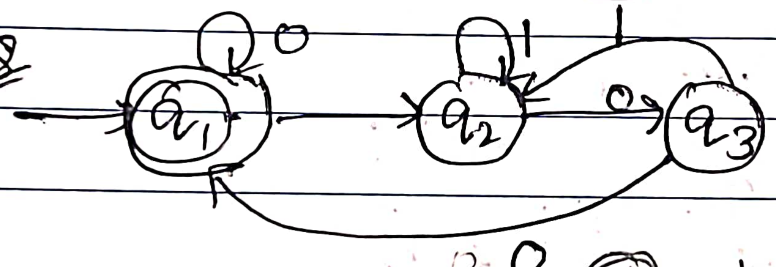
Q last two symbols are same.
$(0+1)^*(00+11)$

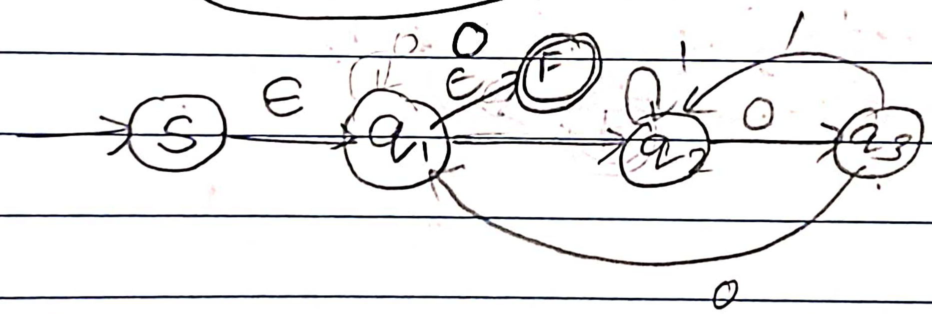
Q every 0 is followed by atleast 2 ones.
$(1+011)^*$

Q Strings have utmost one pair of consecutive 1s. $(01+10)^*(11+\epsilon)(0+01)$ ~~*~~

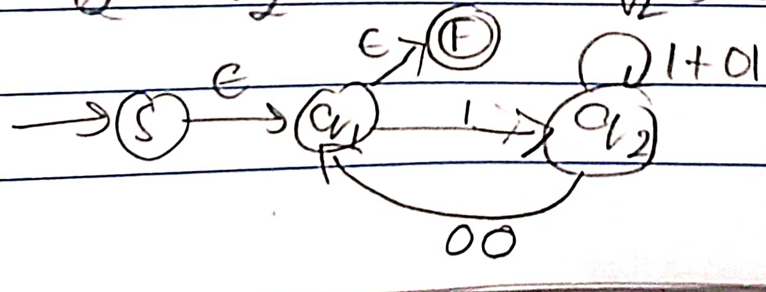
State elimination

Q

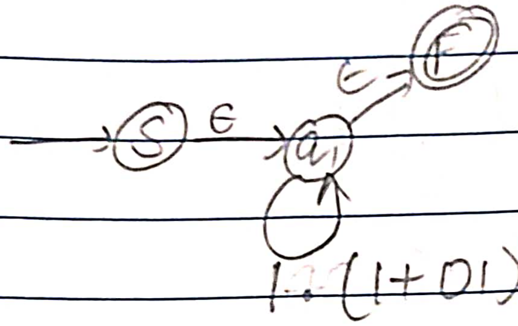




If we want to remove q_3
$q_2 \to q_2 = 1$ $q_2 \to q_2 = 01$ $q_2 \to q_1 \to 00$

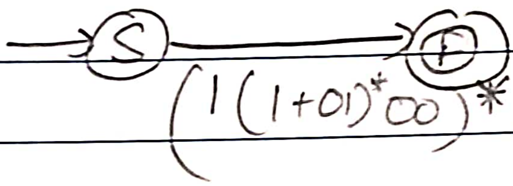


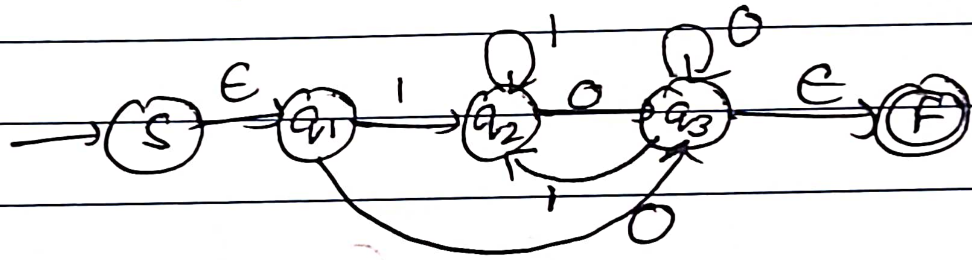
Eliminate q_2

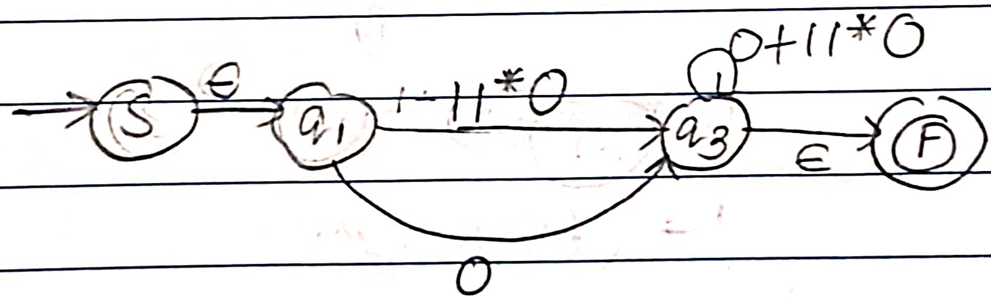


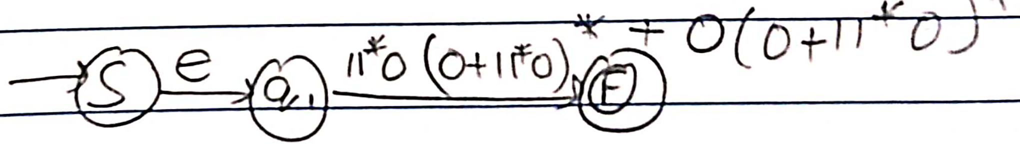
to remove q_2 loop – make closure

remove q_1





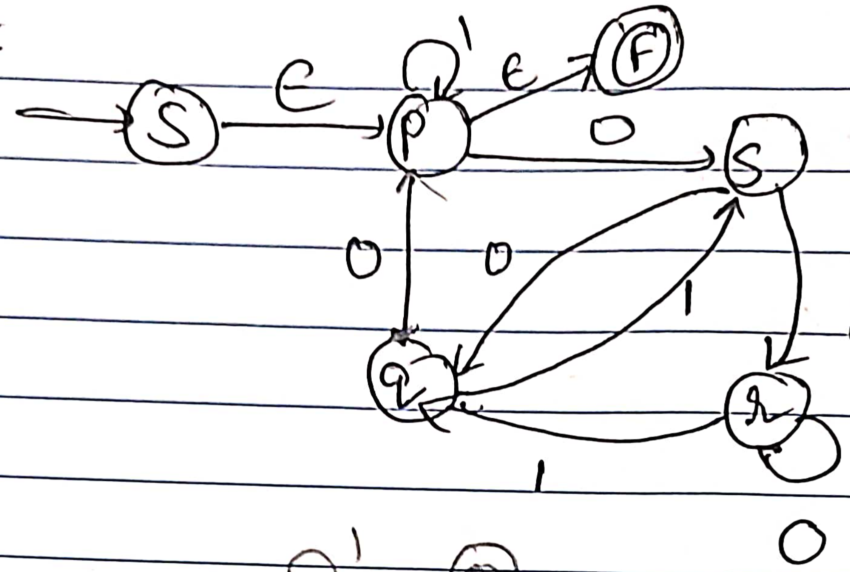


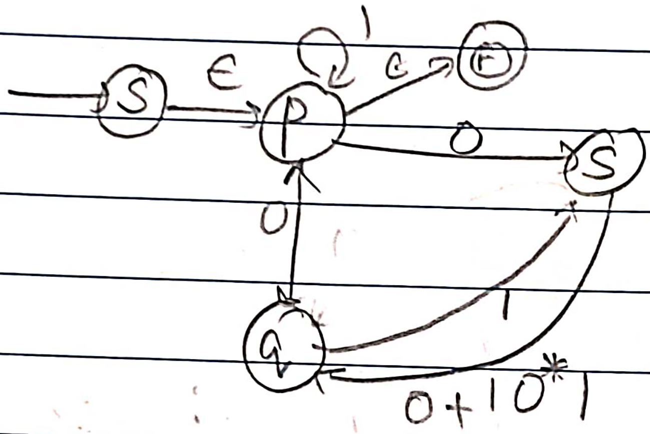


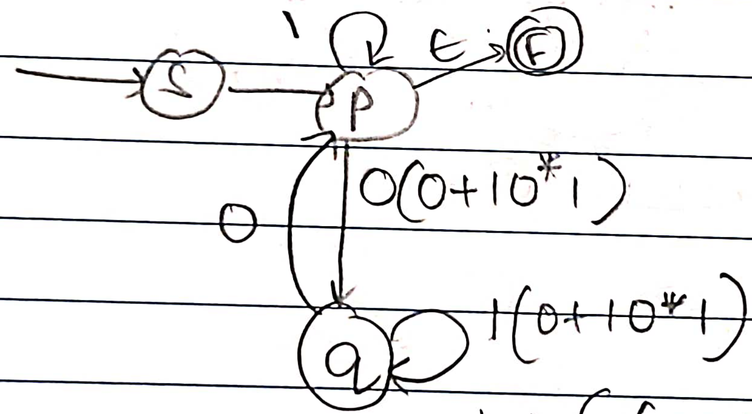
$(11^* + \epsilon)(0(0+11^*0)^*)$

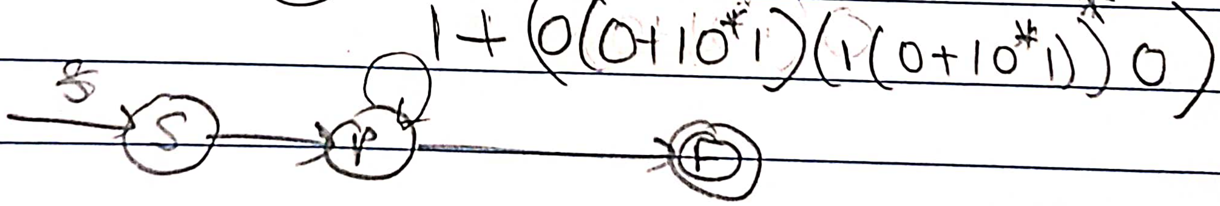
1^*

Q//


S
ε
p
1
ε
F
0
s
0
0
1
1
q
r
1
0


S
ε
p
1
ε
F
0
s
0
1
q
0+10*1

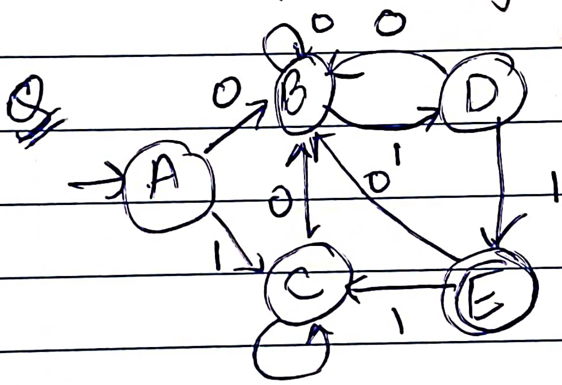

S
p
1
ε
F
0(0+10*1)
0
q
1(0+10*1)


1+(0(0+10*1)(1(0+10*1))*0)
S
p
F

Q	0	1
→ A	B	C
B	D	~~E~~ C
C	~~F~~ D	G
~~E~~	~~F D~~	~~G~~
* D	D	~~E~~ C
~~* F~~	~~D~~	~~E~~
* G	~~F~~ D	G

① D and F are similar states
② C and E are similar
See separately for accepting & non-accepting

Q



	0	1
A	B	C
B	B	D
C	B	C
D	B	E

B	X			
C	✓	X		
D	X	X	X	
E	X	X	X	X
	A	B	C	D

Q

	0	1
$\rightarrow q_1$	q_2	q_3
q_2	q_3	q_5
$*q_3$	q_4	q_3
q_4	q_3	q_5
$*q_5$	q_2	q_5

q_2				
q_3	x	x		
q_4		✓		
q_5	X			
	q_1	q_2	q_3	q_4

Q

	0	1
$\rightarrow$ A	B	A
B	A	C
C	D	B
* D	B	A
E	D	F
F	G	E
G	F	G
H	G	D

	A	B	C	D	E	F	G
B	X						
C	X	X					
D	X	X	X				
E	X	X	✓	X			
F	X	✓	X	X	X		
G	✓	X	X	X	X		
H	X	X	X	X	X	X	X

Myhill Nerode theorem

L x y

Q/ ~~0 (0+1)* 1~~

xz yz → distinguishing extension

A language is regular iff L has a finite number of equivalence classes.

n = no. of states in minimised DFA

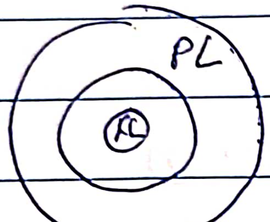
Pumping lemma

used to prove it is not regular
lang might contain infinite no. of strings

let L be a regular language. Then exists a constant n such that for every w in L $|w| \geq n$ we can break w into 3 strings

$$w = xyz$$

1. $y \neq \epsilon$
2. $|xy| \leq n$
3. $\forall k \geq 0$, string xy^kz is also in L



PL does not satisfy → not RL
PL satisfies – may/may not be RL
RL – satisfies PL
not RL – may/may not satisfy PL

Q. $L = \{0^n 1^n\}$

Q. P_1 (proving not reg) $\quad P_2$ (prov reg)

$\{0^n 1^n\}$ →

← picks n

picks w

$|w| \geq n$ → $w = xyz$

←

xy^kz

$8/2$

$L = \{a^p\} \longrightarrow$ picks n $\quad a = 1$

p is prime

picks w

$|w| \geq n \longrightarrow w = xyz$

$|y| = m$

$|xy| = p - m$ $\quad$! ! ! ! ! !

$xy^k z$

$\overbrace{|\,|\,\underbrace{|\,|\,|}_{m}\,|\,|\,|}^{p}$

$\cancel{x(1^n)^{p+m} z}$

$xz \Rightarrow p - m$

$p - m + m(p-m) = (1+m)(p-m)$

$x(1)^m z$

To disprove let $k = p - m$

$x\,(1^m)^{p-m}\,z$

should be prime

$p - m + m(p-m) = \underbrace{(m+1)(p-m)}_{\text{2 factors not prime}}$

Hence disproved

Not regular language

$L + M = M + L$

$L \cdot M \neq M \cdot L$

$L+(M+N) = (L+M)+N$

$(LM)N = L(MN)$

$L+\phi = L$

$L.\epsilon = L$ $\qquad L\phi = \phi$

$\phi^* = \epsilon$ $\qquad \epsilon^* = \epsilon$

$L+L = L$ Idempotency

$L.(M+N) = LM+LN$

$(M+N).L = ML+NL$

$L^+ = L^*L = LL^*$

$L^? = \epsilon + L$

Q $(1+00^*1)\epsilon + (1+00^*1)(0+10^*1)^*(0+10^*1)$

$(1+00^*1)(\epsilon + (0+10^*1)^*(0+10^*1))$

Q $(1+00^*1) + (1+00^*1)(0+10^*1)^*(0+10^*1)$

$(1+00^*1)(\epsilon + \underbrace{(0+10^*1)^*(0+10^*1)}_{\text{0 or more times}})$

$(1+00^*1)(0+10^*1)^*$

$(\epsilon+00^*)1(0+10^*1)^*$

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

languages

Closure properties of regular ~~expression~~

~~L+R~~

1. Closure under union
2. closure under complementation

$$\bar{L} = \Sigma^* - L$$

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

$\Sigma^* = \{\epsilon, 0, 1, 01, 10 \ldots\}$

for L: DFA $A(Q, \Sigma, \delta, q_0, F)$

for $\bar{L}$: ~~(B)~~ $B(Q, \Sigma, \delta, q_0, Q - F)$

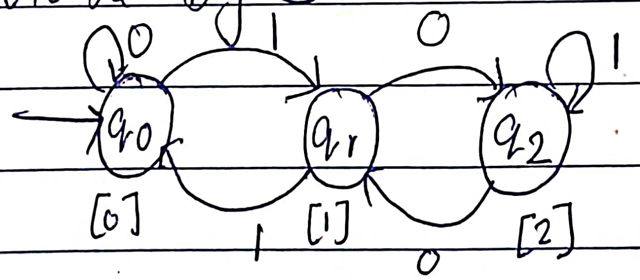
inver accepting & non accepting states

3. closure under intersection

$$L \cap R = \overline{\bar{L} \cup \bar{R}}$$

~~Mealy~~ moore machine -

Divisible by 3

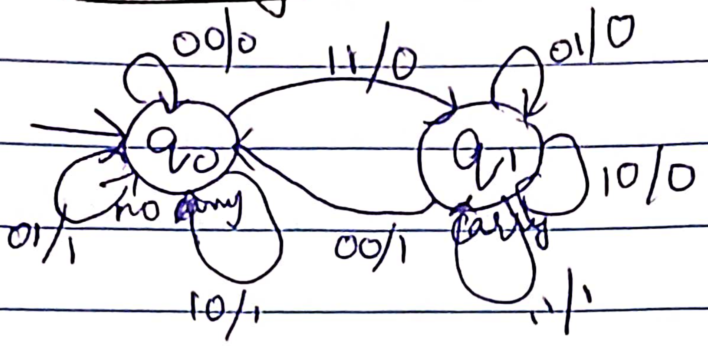


output does not depend on input

$(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

$\{0,1\}$ $\{0,1,2\}$ o/p func

Mealy machine



$$(Q,\ \Sigma,\ \Delta,\ \delta,\ \lambda,\ q_0)$$

Σ ↓ set of accepted alphabets $\{0,1\}$

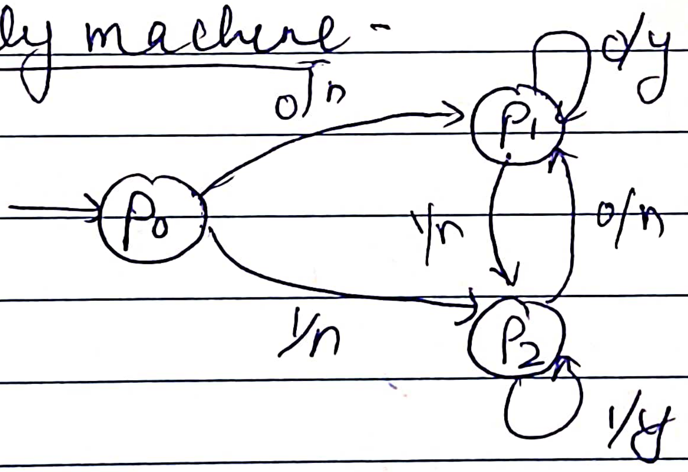
Δ ↓ So.

λ ↓ o/p func

$$\lambda : \Sigma \times Q \rightarrow \Delta$$

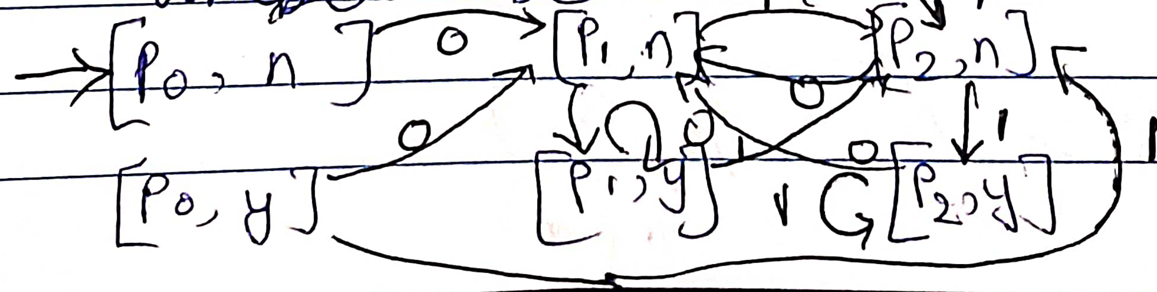
Q/ $(0+1)^*(00+11)$

Mealy machine -



Moore machine

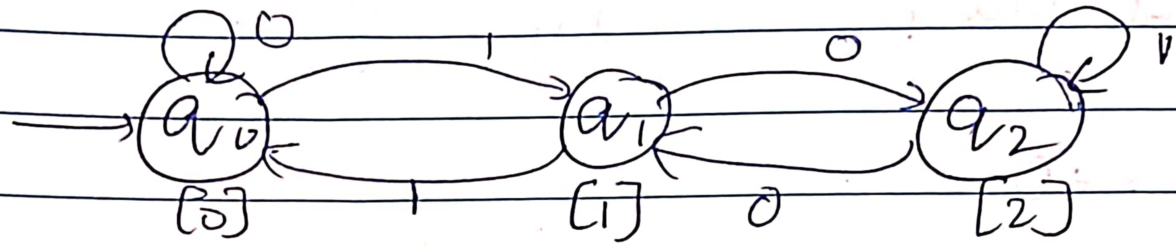
Combine state & outputs

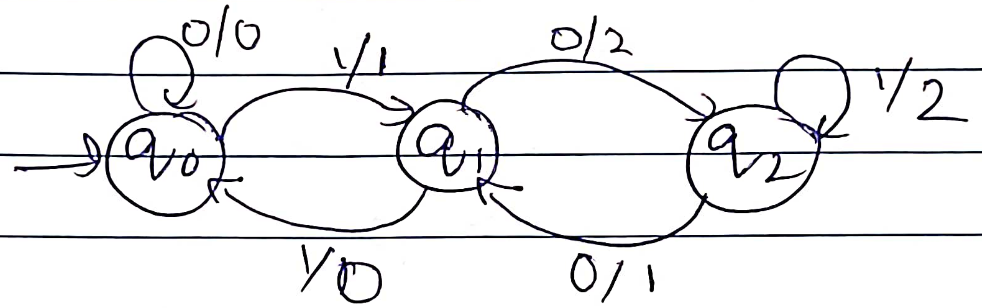


$(Q \times \Delta, \Sigma, \Delta, \delta', \lambda', [q_0\ b])$

$\delta'([q, b], a] = [\delta(q, a), \lambda(q, a)]$

Moore to mealy machine -





Pumping lemma

Proof strategy -

1) suppose L is regular
2) There exists a constant n for L
3) Choose one string $w \in L$ and $|w| \geq n$
4) Look at every decomposition of $w = xyz$ ST $|y| \geq 1$ and $|xy| \leq p$

Q) $L = \{0^i 1^j \quad i > j\}$

Let $w = 0^{n+1} 1^{n}$ $\quad |xy| \le n$

$= \underbrace{0^{\alpha}}_{x} \underbrace{0^{\beta}}_{y} \underbrace{0^{n+1-(\alpha+\beta)} \, 1^{n}}_{z}$

Find one k ST $xy^k z \notin L$

Let $k = 0$

$\cancel{|xy^k z| = \alpha + n + 1 - \alpha - \beta + n}$

$\cancel{= 2n + 1 - \beta}$

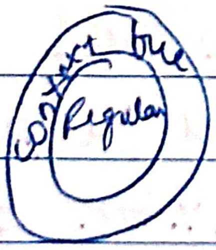
no. of 0s $= \alpha + n + 1 - \alpha - \beta$

$= n + 1 - \beta \le n$

$\beta \ge 1$ true

17/2

context free languages



grammar - expresses string in context free languages

non terminals - uppercase

terminals - alphabet

$S \rightarrow A01$ (production)

set of productions - grammar