Regular Expression:

If we don't ~~to~~ follow these rules there will be syntex / compilation error

Alphabet → symbols, letters, number

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

$\Sigma = \{A\text{-}Z, a\text{-}z\}$

Regular Expression / Regular Language:

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

$L = \{0^*\} \rightarrow \epsilon$, "0", "00", "000", ----

$*$ → zero time or multiple times

$L = \{0 + 1\} \rightarrow$ "0" / "1"

$\cdot$ → and

$L = \{1^*\}$

$+$ → or

$L = \{\epsilon\} \rightarrow$ " " → null string

$L = \{0\} \longrightarrow$ "0"

$L = \{1\} \longrightarrow$ "1"

$L = \{(01)^*\} \rightarrow$ " ", "01", "0101", "010101", - - -

Scanned by CamScanner

$L = \{(0+1)^*\} \to$ "", "0", "1", "00", "11" . . . . . .
"01", "10", "0101", "1001" . . .

$(1.0) = (10)$ same

$L = (0+1)^* 11 \Rightarrow$ "11", "011", "111", "0111", "1011",
"0011", "1111"

$L = (00)^* \to$ null ~~or~~, even number of zero

$L = (00)^* 0 \to$ "0", "000" ~~even~~ odd number of zero

Design regular expression for

i. String with all zero's followed by 1's
ii. strings that have at least one 1 . $(0+1)^* 1$
iii. " " " " " " two 1 $(0+1)^* 11$
iv. string with even no. of followed by odd no of zeros.
v. " " " length.

1. All strings of 0's and 1's

A : $L = (0+1)^*$

2. All string of 0's and 1's that end with at least two 1's

A : $L = (0+1)^* 11$

3. All strings where all 0's followed by 1s.

0101, 01, 010111, 1

A : $L = (1+01)^*$

4. String that end with three consecutive 1's.

A : $L = (0+1)^* 111$

5. Strings that have at least one 1.

A : $L = (0+1)^* 1 (0+1)^*$

6. " " " two 1's.

A : $L = (0+1)^* 1 (0+1)^* 1 (0+1)^*$

7. " " " at most one 1.

A : $L = 0^* 1 0^* + 0^*$

8. String with even no. of 1s only.

A : $L = (11)^*$

9 " " odd on of 1s only.

A : $L = (11)^* 1$

10. string where even no. of 1s followed by odd no 0's.

A : $(11)^* (00)^* 0$

11. String with even length

A : $L = (00+01+10+11)^*$ / $L = (00)^* + (01)^* + (10)^* + (11)^*$

12. String where second or third position from the end is 1.

A : $L = (0+1)^* 1 (0+1) + (0+1)^* 1 (0+1)(0+1)$

13. String with exactly three 1's.

A : $L = 0^* 1 0^* 1 0^* 1 0^*$

14. String Length divisible by two.

A : $L = (00+01+10+11)^*$ / $L = (00)^* + (01)^* + (10)^* + (11)^*$

15. String with no of 1's divisible by two.

A : $(0^* 1 0^* 1 0^*)^*$

16. String with alternating 0 and 1

0101 , 1010 , 101 , 010

A : $L = (01)^* + (10)^* + 1(01)^* + 0(10)^*$

or $L = (\epsilon + 1)(01)^*(\epsilon + 0)$

17. String that have at most three 1

$L = 0^* 1 0^* 1 0^* 1 0^* + 0^* 1 0^* 1 0^* + 0^* 1 0^* + 0^*$

for alternating :

* start with 0 , end with 1
* " " 1 , " " 0
* " " 1 , " " 1
* " " 0 , " " 0
* null is also acceptable

$01^* + 1 \rightarrow$ either 1 or the strings starts with zero.
$(0, 01, 01111, 1 - - - -$

$(01)^* + 1$
$\epsilon, 1, 01, 0101, 010101, - - - - - -$

$0(1^* + 1)$ either 0 or the strings starts with zero
$0, 1, 01, 011, 01111, - - - - - -$

DFA - Deterministic Finite Automata

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

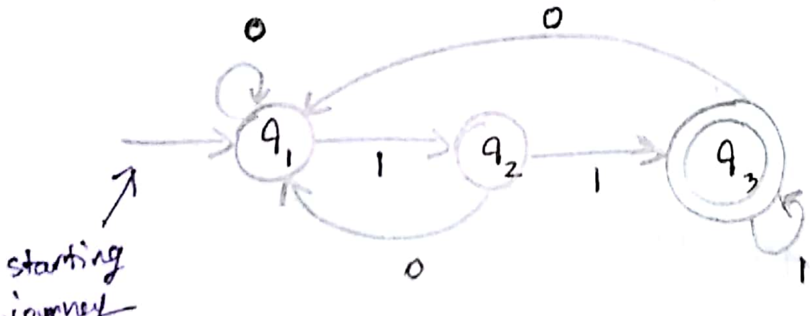
Q → finite set of state

Σ → finite set of symbols called alphabet

δ → transition function
$\delta(q, a) = ?$

q_0 → starting state / initial state ($q_0 \in Q$)

F → set of final states / accepting states ($F \subseteq Q$)



◎ → accepting state

0011 → acceptable

00111 → not acceptable

Here,

$Q \to \{q_1, q_2, q_3\}$

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

$q_0 \to \{q_1\}$

$F \to \{q_3\}$

Regular Expression Practice Problem:

1. The set of strings with at most one pair of consecutive 1's.

11 1

Ans: ~~$0^* 110^* + 0^*$~~ $(10+0)^*(11)(0+01)^* + 1$

$(0+10)^*(11+\epsilon)(0+10)^*(\epsilon+110)$ 1101

2. The set of strings whose fifth symbol from the right end is 1

Ans: $(0+1)^* 1 (0+1)(0+1)(0+1)(0+1)$

3. The set of strings not containing 101 as a substring.

Ans: $(\epsilon+0^*+1^*)(00+11)^*(\epsilon+0^*+1^*)$

4. The set of strings having 0 at every 3rd position.

Ans: $(0+1)(0+1)0(0+1)^*$

5. The set of strings of that ends with 0 and does not contain the substring 11.

Ans:

6. ~~The~~ set of all strings that are at least of lenght 4 and contains even number of 1's

$(0+1)(0+1)(0+1)(0+1)$

Ans: $(0^*10^*10^*10^*10^*)^*$

16

(7). Set of all strings with number of 0's is divisible by 4 and number of 1's is divisible by 5.

Ans :

(9). The set of all strings such that each block of five consecutive symbols contains at least two 0's.

Ans :

10. L = {w | w has even number of 1's and one or two 0's}

Ans : $(11)^* 0 (11)^* + (11)^* 0 (11)^* 0 (11)^*$

~~11.~~ L = {w | w contains ~~neig~~ neither the substrings 01 nor 10}

Ans : $0^* + 1^*$ 011

12. The set of strings of 0's and 1's that contain exactly 4 1's.

Ans : $0^* 1 0^* 1 0^* 1 0^* 1 0^*$

13. The set of strings of 0's and 1's with odd number of 0's.

Ans : $(1^* 0 1^* 0 1^*)^* 0 + 1^* 0 1^*$

Femastin™

≥ 2

14. The set of strings of 0's and 1's with at least two 1's or exactly two 0's.

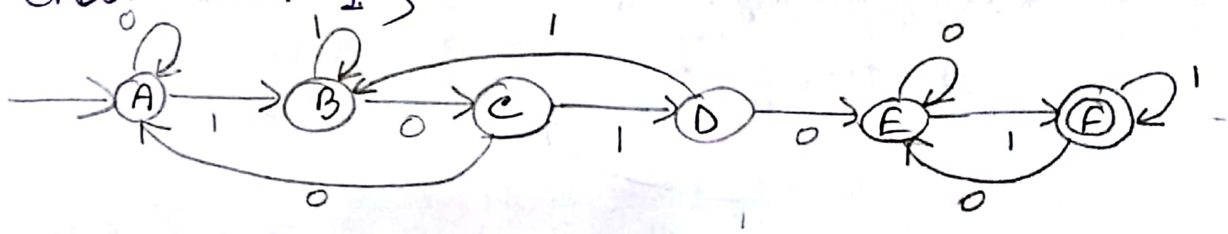
Ans: $(0+1)^* 1 (0+1)^* 1 (0+1)^* + 1^* 0 1^* 0 1^*$

15. The set of strings of 0's and 1's that contain the substring 10 or substring 01.

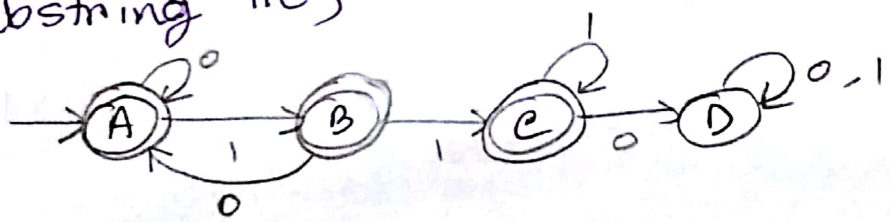
Ans: $(1+0)^* 10 (1+0)^* + (1+0)^* 01 (1+0)^*$

DFA Design:

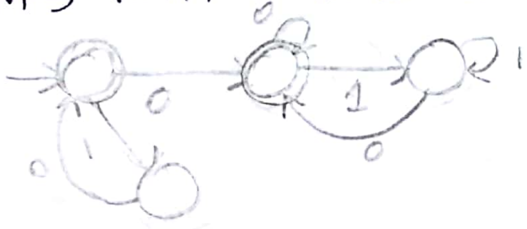
(i) $L = \{W \mid W$ contains 1010 as substring and ends with 1$\}$



(ii) $L = \{$ All strings that don't contain the substring 110$\}$

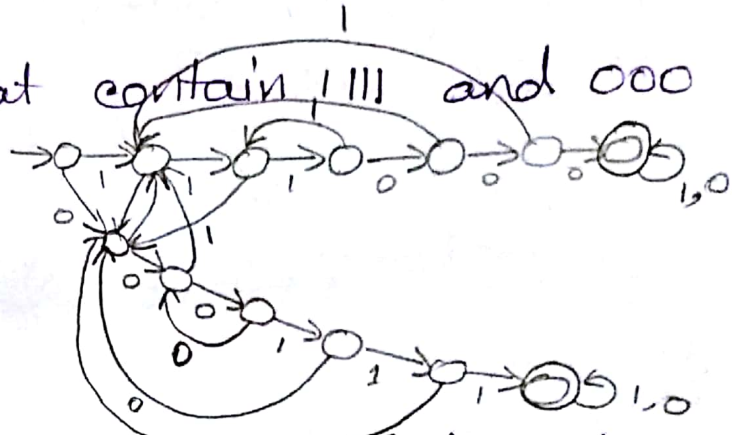


(iii) $L = \{ W \mid W$ does not contain as substrings, starts with 0 and ends with 1$\}$



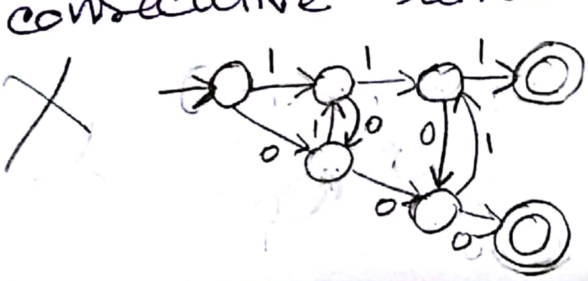
(iv) $L = \{$ All strings that contain the substring 110 at least two 0's$\}$

(V) L = { All strings that contain 1111 and 000 as substrings }



(VI) L = L = { W | W has odd length and ends with consecutive same symbol }

0100



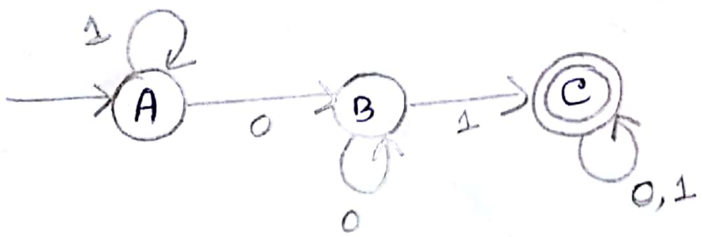
00111

00011

010010

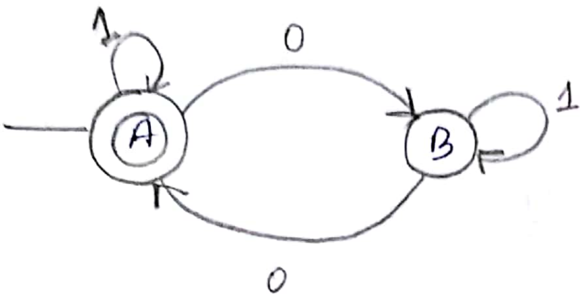
Design a dfa to accept the set of all strings with substring 01

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

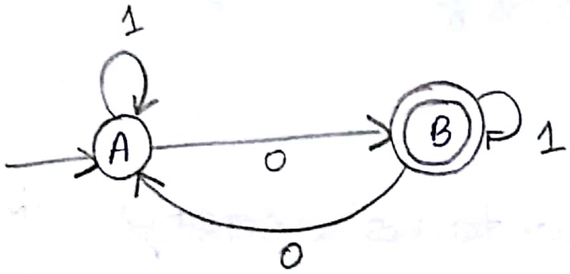


Design a dfa to accept even no. of 0.

$(1^* 0 1^* 0 1^*)^*$

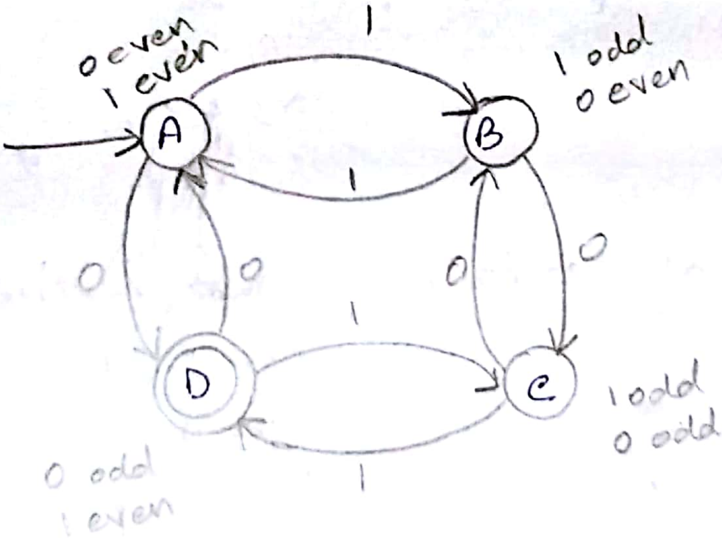


Design a dfa of accept odd no. of 0.

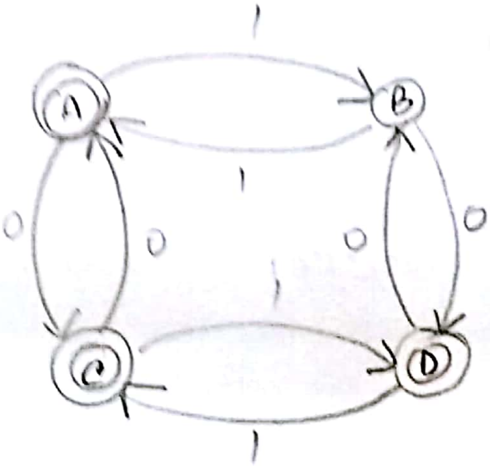


DfA to accept even no of 1s and odd no of 0's.

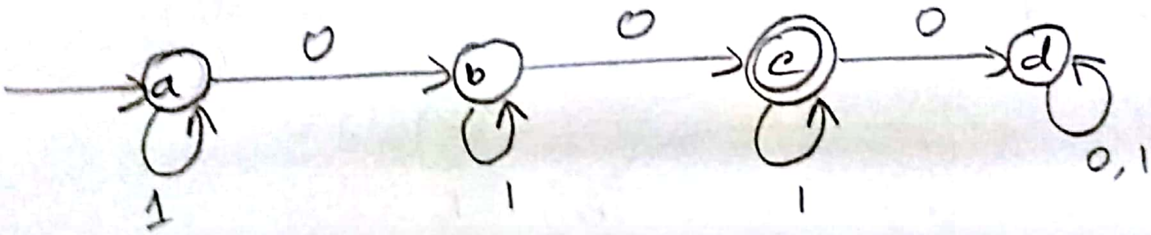
1100 ×
11100 ×
1100 ✓
11000 ×



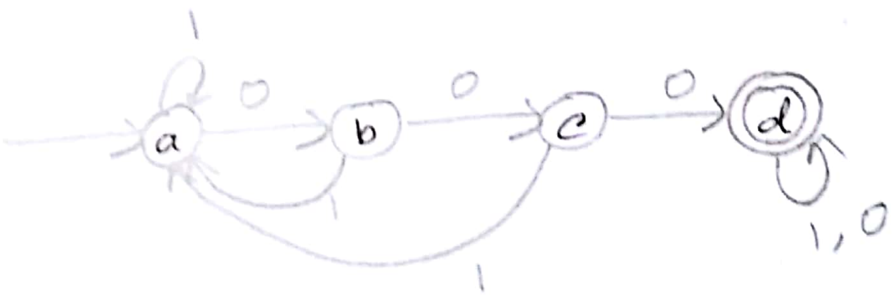
☐ DFA to accept even no of 1s or odd no of 0's.



☐ String that contains exacty two 0's $\Sigma = \{0,1\}$



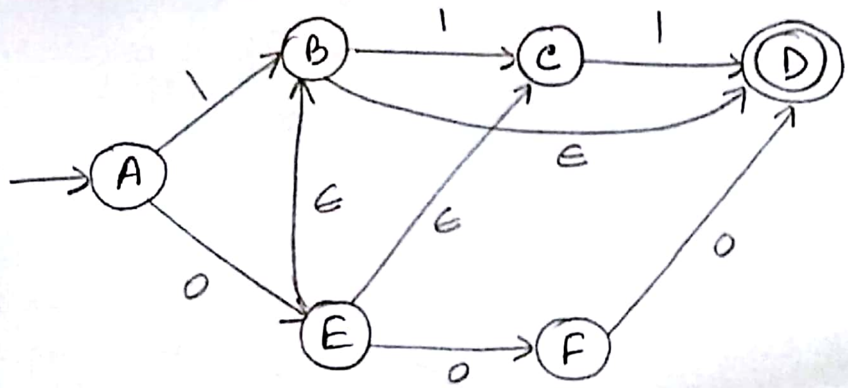
☐ Strings that contains 3 consecutive 0's



NFA - Non deterministic Finite Automata

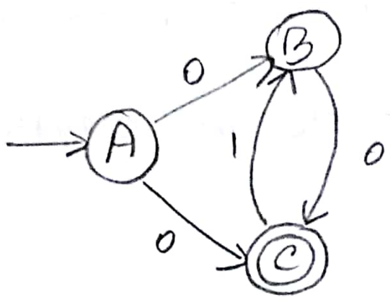
↳ NFA has the ability to be in several states simultaneously.

NFA with ∈-transition: allow state-to state transition on ∈ input.



Transition function :

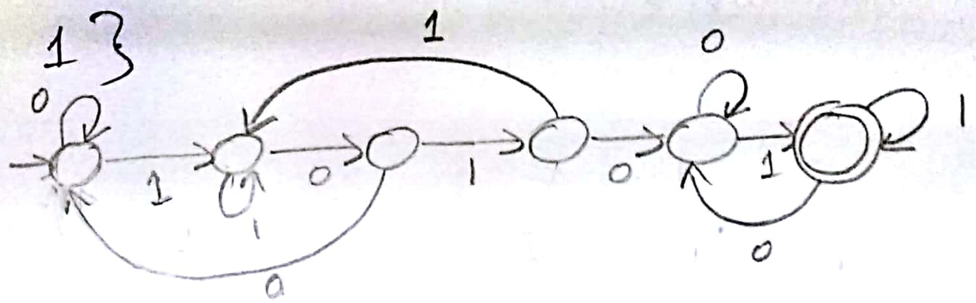
		0	1	∈
→	A	{E}	{B}	∅
	B	∅	{C}	{D}
	C	∅	{D}	∅
*	D	∅	∅	∅
	E	{F}	∅	{B, C}
	F	{D}	∅	∅



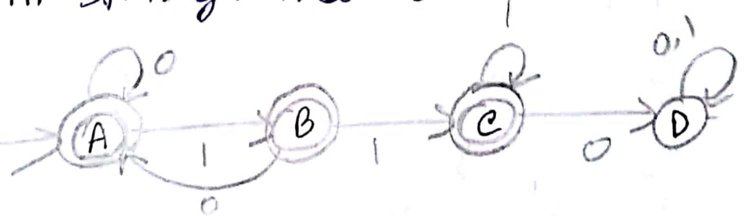
DFA	NFA
1. all Transition fixed	1. all Transition is not fixed
11. Multiple transition impossible	11. Multiple transition possible
111. ∈ transition is not possible	1 ∈ transition is possible

DFA Design :

* L = {W | W contains 1010 as substring and ends with 1}



* L = {All strings that don't contain the substring 110}



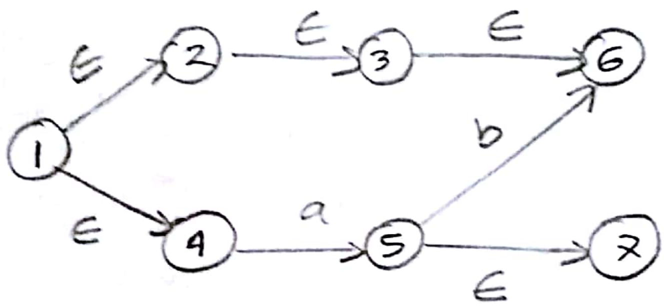
ϵ Closure of States:

closure

$CL(q) \rightarrow$ set of states you can reach from state q following only arcs leveled ϵ.

$CL(A) = \{A\}$

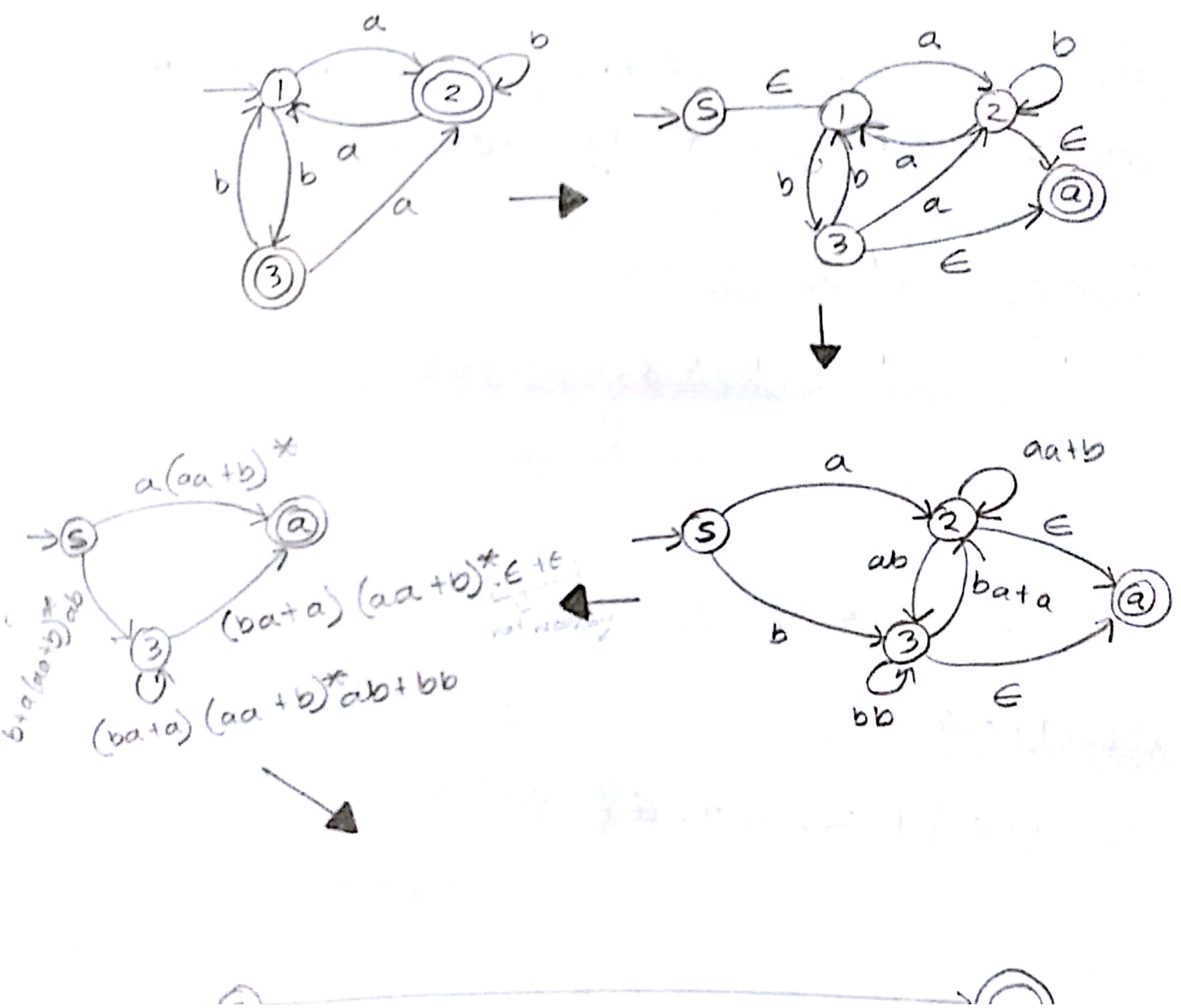
$CL(E) = \{B, C, D, E\}$



$CL(1) = ?$

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

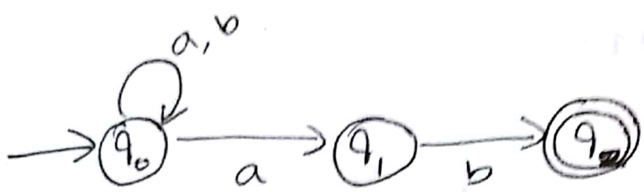
DFA to Regular Expression : by state elimination



$NFA \rightarrow \langle Q, \Sigma, q_0, \delta, F \rangle$

$\downarrow$

$DFA \rightarrow \langle Q', \Sigma, q_0, \delta', F' \rangle$



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

$F = \{q_2\}$

~~Tramsform~~ Transition table δ:

Start	a	b
q_0	$q_0\ q_1$	q_0

start	a	b
q_0	$\{q_0, q_1\}$	$\{q_0\}$

Now $Q' = \{q_0, \{q_0, q_1\}\}$

$$\delta'(\{q_0, q_1\}, a) = \delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \{q_0\}$$

$=$

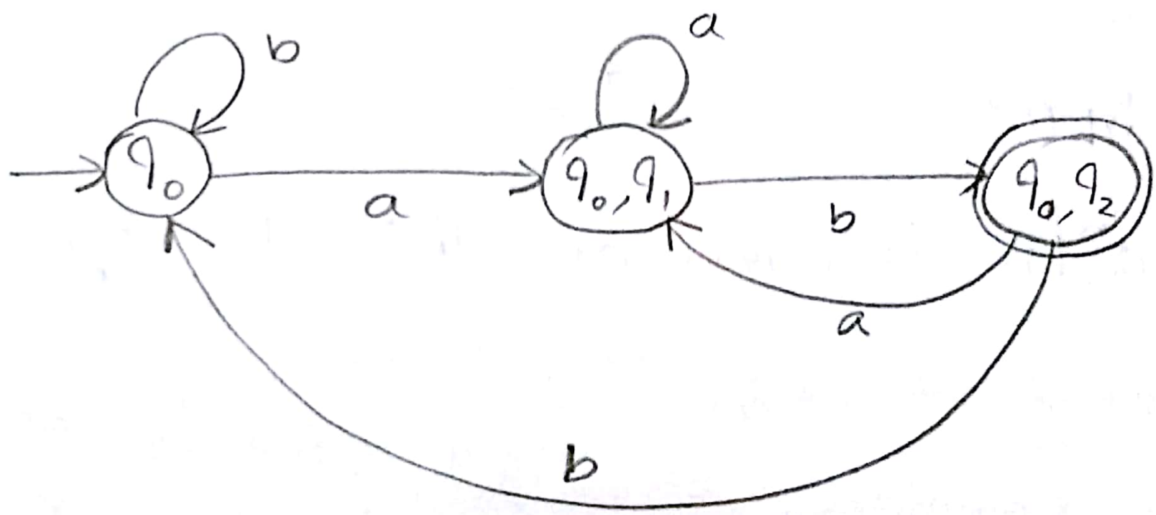
Start	a	b
q_0	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

$$\delta'(\{q_0, q_1\}, b) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_0\} \cup \{q_2\}$$

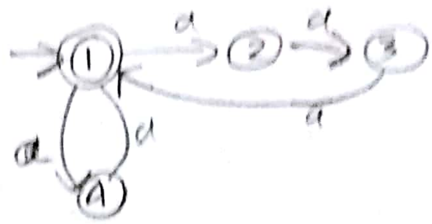
$=$

Now $Q' = \{q_0, \{q_0, q_1\}, \{q_0, q_2\}\}$

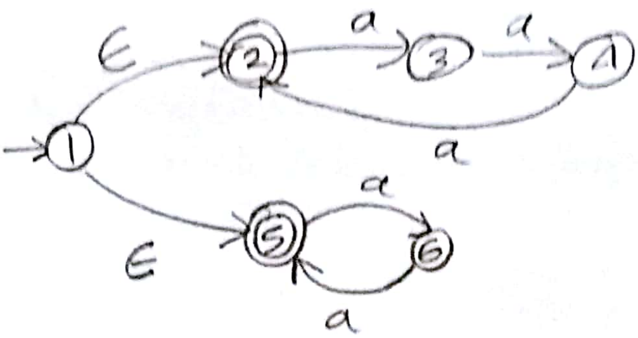
	a	b
q_0	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$


b
a
q_0
q_0, q_1
q_0, q_2
a
b
a
b

Design NFA:

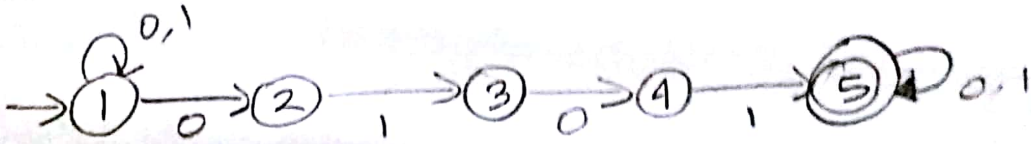


* $L = \{ a^n / n$ is even or divisible by $3 \}$



ϵ ✓
aa ✓
aaa ✓
aaaa ✓
aaaaa ✗

* Design NFA to accept strings with substring 0101



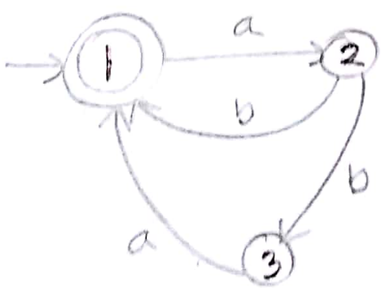
$0 \rightarrow \{1, 2\}$

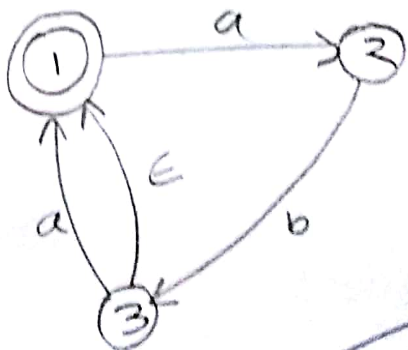
$01 \rightarrow \{1, 3\}$

$010 \rightarrow \{1, 4, 2\}$

$0101 \rightarrow \{1, 3, 5\}$ → accepting state

* Design NFA for $L = (ab + aba)^*$





Mid

CFG → Context Free Grammar

$\boxed{S \rightarrow \underline{01}}$

production rule

terminal

$S \rightarrow 0S1$

NON terminal

I. Production rule

II. terminal (alphabates are always terminal) on 0,1

III. non terminal (alway be on left side in production rules)

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

$S = \epsilon$

$S = 0S1$

/ $S = 0S1 / \epsilon$

$S \rightarrow 0S1$

$S \rightarrow 00S11$

$S \rightarrow 000S111$

$S \rightarrow 000111$

Unsigned integers:

$U \rightarrow UD \mid D$

$D \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$D \rightarrow UD \mid D$

$U \rightarrow \underline{U}D$ is this accept 123 ?

$\rightarrow \underline{U}DD$

$\rightarrow DDD$

$\rightarrow 123$

Left most Derivations :

$S \rightarrow SS \mid (S) \mid ()$ is (())() is acceptable?

$S \rightarrow SS$

$\rightarrow (S)S$

$= (())S$

$\rightarrow (())()$

Right most :

$S \rightarrow SS$

$\rightarrow S()$

$\rightarrow (S)()$

$\rightarrow (())()$

Scanned by CamScanner

Palindreme CFG:

$P \rightarrow aPa \mid bPb \mid \epsilon$ string $\rightarrow$ abba

$P \rightarrow aPa$

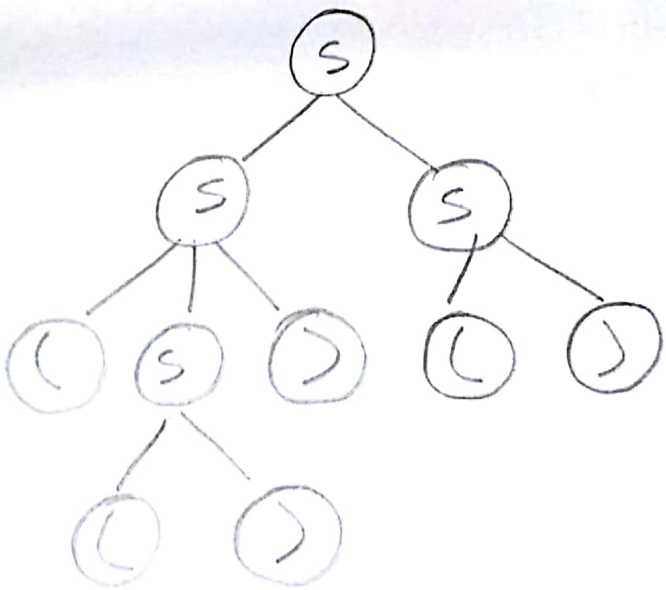
$\rightarrow abPba$

$\rightarrow abba$

Parse Tree:

$S \rightarrow SS \mid (S) \mid ()$

for (())()



Ambigious Grammar :

If a tree has multiple parse tree than this grammar is called ambigious grammar. also it is called LL(1) grammar.

(()) () for

$B \to (RB \mid \epsilon$

$R \to) \mid (RR$

$B \to (RB$

$((RRB$

$(())B$

$(())(RB$

$(())()$