Unit - II

1) List and explain the closure properties of Regular Grammar.

The closure properties of regular grammar are:

1) Union
2) Concatenation
3) Kleene star (closure)
4) Reversal.

1) **Union:-**

let G_1 generated the language

$L_1 = \{0, 01, 001, 0001 \ldots\}$

let G_2 generated the language

$L_2 = \{1, 10, 110, 1110 \ldots\}$

Then

$G_1 \cup G_2 = L_1 \cup L_2 = \{0, 01, 1, 10, 001, 110 \ldots\}$

2) **Concatenation:-**

let G_1 generate the language and G_2 generate the language

$L_1 = \{a, bb\}$

$L_2 = \{x, y\}$

Then concatenation

$G_1 G_2 = L_1 \cdot L_2 = \{ax, ay, bbx, bby\}$

3) **Kleene star (closure):-**

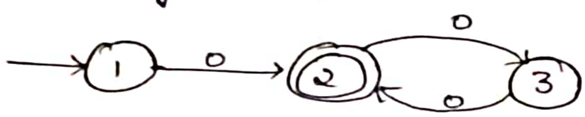
let G generate the language $L = \{a, b\}$. Then kleene

Closure (G^*) generates } $L^* = \{\epsilon, a, b, aa, ab, ba, bb ----\}$

4, Reversal:-

Let G generate the language $L = \{abc, abcd, abcde\}$ the reversal G^R generates $L^R = \{cba, dcba, edcba ---\}$

2, Compute the regular expression for the following Machine.



Step1:- Construct state equations for all states based on incoming edges.

$$1 = \epsilon \longrightarrow ①$$

$$2 = 1\cdot 0 + 3\cdot 0 \longrightarrow ②$$

$$3 = 2\cdot 0 \longrightarrow ③$$

Step2:- Add ϵ to the initial state equation

$$1 = \epsilon$$

Step3:- simplify the final state equation using Ardens theorem and find regular expression.

$$2 = 1\cdot 0 + 3\cdot 0$$

Substitute ① in eq ②

$$2 = 2\cdot 0 + \epsilon\cdot 0$$

$$2 = \epsilon 0 \; 0^*$$

$$= 00^*.$$

$$\boxed{RE = 00^*}$$

$$\boxed{R = Q + RP \Longrightarrow QP^*}$$

$Q = \epsilon \cdot 0$

$R = 2.$

$P = 0$

3 Write the regular expression for the language L over $\Sigma = \{0,1\}$ such that all the strings do not contain the substring 01.

Sol:

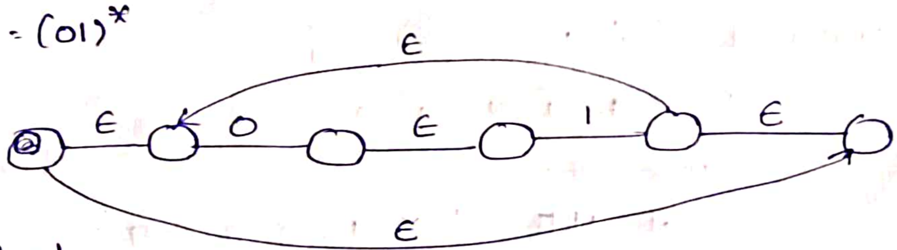
Given condition as strings do not contain the substring 01. that means 0 never follows 1. So, to satisfy the given condition we take strings as 1 followed by '0'.

$$\boxed{R.E = 1^* 0^*}$$

4. Draw the DFA for the following Regular Expressions

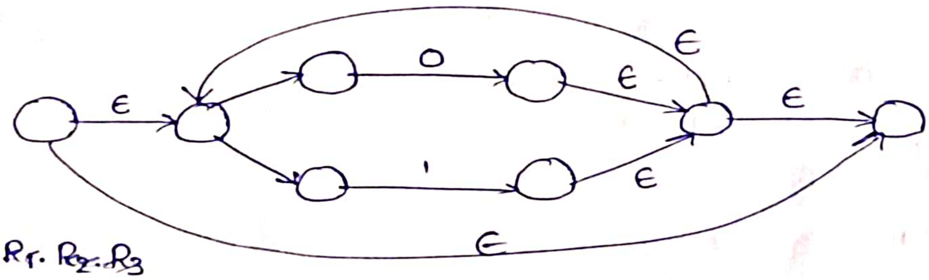
(i) $(01)^* 1 (0+1)^*$

Let $R_1 = (01)^*$



$R_2 = 1$

$R_3 = (0+1)^*$



$R_4 = R_1 . R_2 . R_3$

$R_4 = R_1 \cdot R_2 \cdot R_3$

5. Explain the pumping lemma for regular sets. show that $L = \{a^P / P \text{ is a Prime}\}$ is not regular.

Sol:-

Given $L = \{a^P / P \text{ is a Prime}\}$

$L = \{a^2, a^3, a^5, a^7, a^{11} \ldots\}$

Assume $n = 3$

$W = aaa$ $\{x = a,\ y = a,\ z = a\}$

(i) $|xy| < n \Rightarrow |aa| < 3$

$2 < 3$ ✓

(ii) $|y| > 0 \Rightarrow |a| > 0$

$1 > 0$ ✓

(iii) $xy^iz \in L$

for $i = 0 \Rightarrow xy^0z \Rightarrow aa^0a \Rightarrow aa \in L$ ✓

for $i = 1 \Rightarrow xy^1z \rightarrow aa^1a \Rightarrow aaa \in L$ ✓

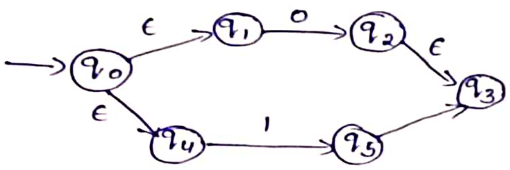
for $i = 2 \Rightarrow xy^2z \rightarrow aa^2a \Rightarrow aaaa \notin L$

Hence it is not Regular

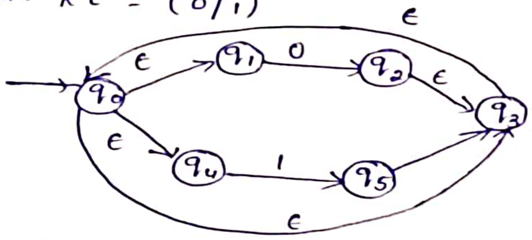
6. Draw NFA for this Regular expression on $(0/1)^*011$ with ϵ- closures and convert it into NFA.

RE $= (0/1)^*011$

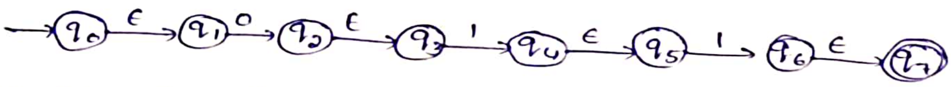
1. RE $= (0/1)$



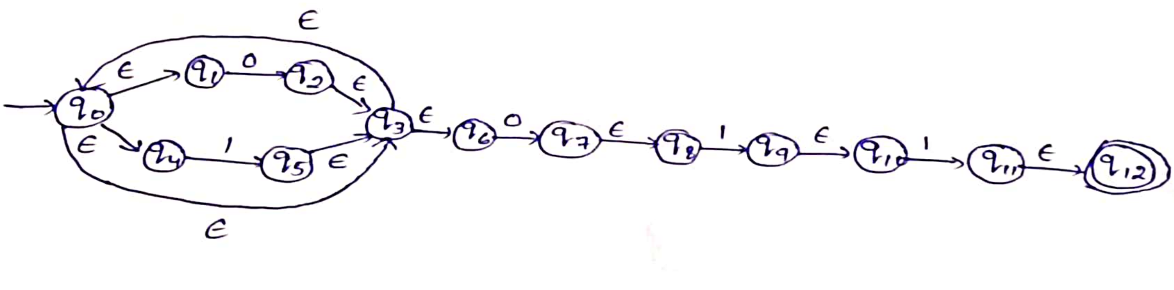
2. RE $= (0/1)^*$



3. RE $= 011$

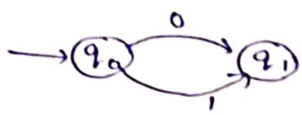


4. RE $= (0/1)^*011$



NFA without ε:

1. RE = (0/1)



2. RE = $(0/1)^*$

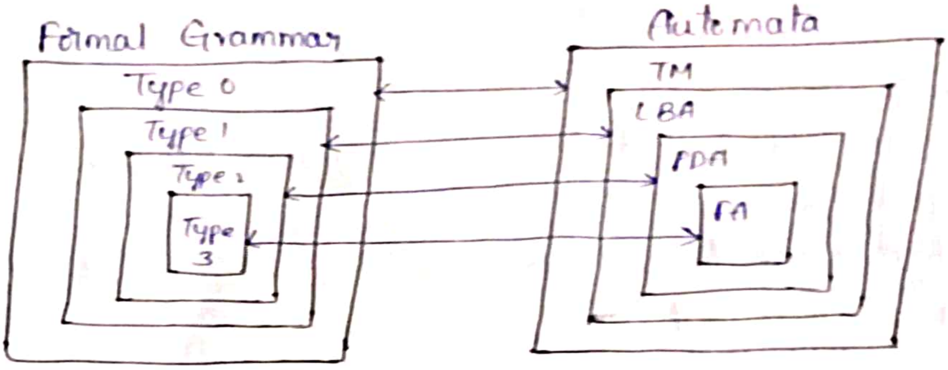
3. RE = 011

4. RE = $(0/1)^*$ 011

→ This is the final NFA.

7. Illustrate the chomsky hierarchy with a neat Sketch

A) Chomsky Hierarchy:-



Type 3 Grammar:-

→ It is also Called regular grammar
→ It is Used to generate Regular language
→ Type 3 Grammar accepts the Finite Automata i.e, NFA (or) DFA
→ The grammar is in the form $\alpha \rightarrow \beta$

Type 2 Grammar:-

→ It is also Called Content free grammar
→ It is used to generate Context Free language
→ Context - Free language recognizes the push down Automata

The grammar is in the form $\alpha \rightarrow \beta$

where $\alpha \in V$

$\beta \in (V \cup T)^*$

Type 1 Grammar:-

→ It is also Called as Context Sensitive grammar
→ It is Used to generate Context-Sensitive language.

→ It is accepted by linear bounded Automata

The grammar is in the form $\alpha \longrightarrow \beta$

where $\alpha \in (V \cup T)^{+}$

$\beta \in (V \cup T)^{*}$

Type 0 Grammar:-

→ It is also Called Recursive grammar / Recursively enumerable grammar.

→ It is used to generate recursively enumerable language.

→ It is accepted by Turing machine

The grammar is in the form $\alpha \longrightarrow \beta$

where $\alpha \in (V \cup T)^{+}$

$\beta \in (V \cup T)^{*}$ $\because |\alpha| \geq |\beta|$

8. What is regular expression? Draw the regular expression for the language over $\{0,1\}$ Such that set of all Strings that Contain exactly three 1's.

A) Regular Expression:- (R.E)

Regular expressions are mathematical expressions which describes a language which is accepted by FA

→ Regular Expression describing a language Called Regular language.

Regular Expression for this string exactly three one' is

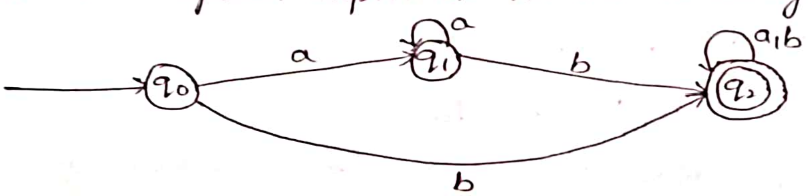
Regular Expression (R.E) = $\{0^{*}10^{*}10^{*}10^{*}1\}$

9. Write the regular expression for the language L over $\Sigma = \{0,1\}$ Such that all the Strings should have at least One 0 at least One 1.

A) Regular Expression for the language L over $\Sigma = \{0,1\}$ Such that all the Strings should have at least One 0 at least One 1 is

Regular Expression (R.E) $= [(0+1)^* 0 (0+1)^* 1 (0+1)^*] + [(0+1)^* 1 (0+1)^* 0 (0+1)^*]$

10. Derive the regular expression for the following DFA.



A) Step 1:- Write state Equations for all States:

$$q_0 = \epsilon \longrightarrow ①$$

$$q_1 = q_0 a + q_1 a \longrightarrow ②$$

$$q_2 = q_0 b + q_1 b + q_2 a + q_2 b \longrightarrow ③$$

Step 2:- Add ϵ to the initial state.

$$q_0 = \epsilon$$

Step 3:- Simplify the Equation using "Ardens theorem" and find regular Expression.

Ardens theorem is $\boxed{R = Q + RP}$ then we Can ~~Say~~ a·Simplify as

$$\boxed{R = QP^*}$$

Here, q_2 is the final state.

$q_2 = q_0 b + q_1 b + q_2 a + q_2 b$

First simplify q_1

$$\underline{q_1} = \underline{q_0 a} + q_1 \underline{a}$$
$$R \qquad Q \qquad P$$

$$\boxed{q_1 = q_0 a a^*} \quad \text{——— (4)}$$

let $q_2 = q_0 b + q_1 b + q_2 a + q_2 b$

$= q_2[a+b] + q_0 b + q_1 b$

substitute q_0 & q_1 (eq (1), (4)) in eq (2)

$q_2 = q_2[a+b] + \epsilon . b + q_0 a a^* b$

$= q_2[a+b] + \epsilon . b + \epsilon a a^* b$ (substitute eq (1))

$= q_2[a+b] + \epsilon[b + a a^* b]$

$$\underline{q_2} = q_2 \underline{[a+b]} + \underline{[b + a a^* b]}$$
$$R \qquad P \qquad Q$$

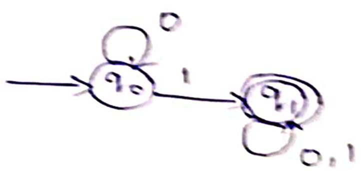
$q_2 = (b + a a^* b)(a+b)^*$

Regular Expression for the given Finite Automata is

$$\boxed{R.E = (b + a a^* b)(a+b)^*}$$

4. Draw the DFA for the following Regular Expressions.

i) $(01)^* \mid (0+1)^*$



ii) $(ab)^* + (a+b)^*$