Unit-2 Regular Expression.

The Regular Expression are the type of languages defining notation

The Regular Expression used to describes the Regular Language.

The Regular Expression is a notation to specify a string that describe the whole set of strings according to set of roots.

The regular expression operators are union, Concatination & closure.

Example:

1) Search Commands in Unix

2) lexical analyzes that recognize the set of tokens from the Source Program.

3) finding Pattern in the text.

Problems on Regular Expression:

1. Find Regular Expression for the set of string {a, b}

The Regular Expression $(a+b)^*$

2. Find Regular Expression for the set of string {a, b} Ending with 'ab'

ab
aab
abab
aabba

The Regular Expression $(a+b)^*ab$

3. Find Regular Expression the I/P String {0, 1} and Starting with 3 Consecuting 1's

111 01
111 10
111 001
111 0011
111 010101

The Regular Expression is $111(0+1)^*$

4. Find regular Expressions, the set of I/P String (a, b, c) the following of a's & b's & c's

The Regular Expression is $a^*b^*c^*$

Operators of Regular Expression:

Union:

The Union of two languages L & M are denoted by $L \cup M$, the set of strings that are either in L or M both

Eg:
$L = \{00, 10\}$
$M = \{11, 01\}$
$L \cup M = \{00, 10, 11, 01\}$

Concatination:

The Concatination of two languages L & M is the set of string that can be formed by taking any String in L and Concatinating in with any string in M.

Eg:
$L = \{00, 10\}$
$M = \{11, 01\}$
$L = \{0011, 0001, 1011, 1001\}$

Closure:

i) ~~Kleen~~ Kleen (or) Star closure

ii) Positive closure

Kleen (or) star closure:

The kleen (or) star closure of a Regular language 'L' include zero (or) more instance of the occurance of the String.

The Kleen closure of the Regular language denoted by L^* and it is defined as

$$\boxed{L^* = \bigcup_{i=0}^{\infty} L^i}$$

Example:

$L = \{0,1\}$

$L^0 = \{0,1\}^0 = \varepsilon$

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

$L^2 = \{0,1\}^2 = \{0,1\}\{0,1\}$

$= \{00, 01, 10, 11\}$

$L^3 = \{0,1\}^3 = \{0,1\}\{0,1\}\{0,1\} = \{0,1\}^2\{0,1\}$

$\vdots$

$= \{00, 01, 10, 11\}\{0,1\}$

$= \{000, 001, 010, 011, 100, 101, 110, 111\}$

$= \{L^0, L^1, L^2, L^3, \ldots \infty\}$

$L^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, \ldots\}$

ii) Positive Closure:

Positive closure of regular language L include one (or) more instance of the occaurance of the String.

Positive closure of regular language is denoted by L^+ and it is defined as

$$\boxed{L^+ = \bigcup_{i=1}^{\infty} L^i}$$

Example:

$L = \{0,1\}$

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

$L^2 = \{0,1\}^2 = \{0,1\}\{0,1\} = \{00, 01, 10, 11\}$

$L^3 = \{0,1\}^3 = \{0,1\}\{0,1\}\{0,1\} = \{0,1\}^2 \cdot \{0,1\}$

$\vdots$

$= \{00, 01, 10, 11\}\{0,1\}$

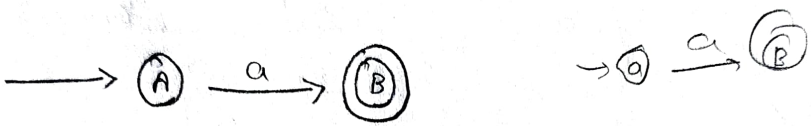
$= \{000, 010, 100, 110, 001, 011, 101, 111\}$

$= \{L^1, L^2, L^3, \ldots \infty\}$

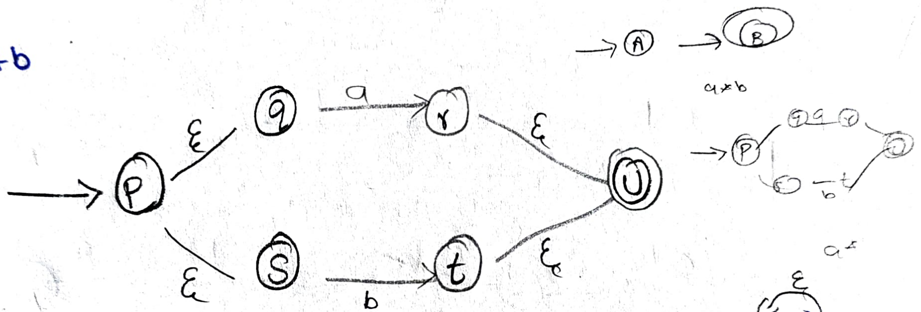
$L^+ = \{0, 1, 00, 01, 10, 11, 000, 010, 100, 110, 001, 011, 101, 111, \ldots\}$

Regular Expression to finite Automata:

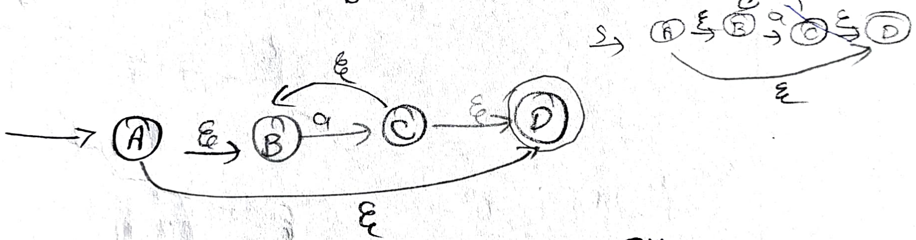
1) a



2) a+b

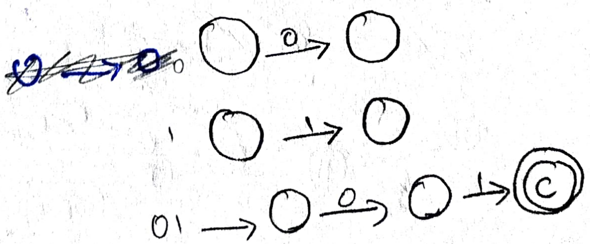


3) a*

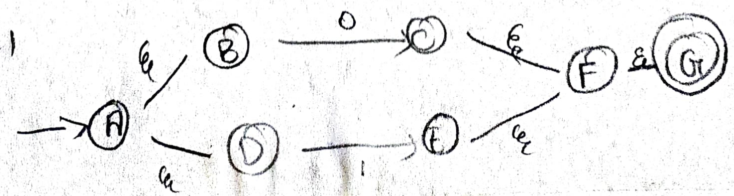


1. Convert the finite automata for the following regular expression {0 1}

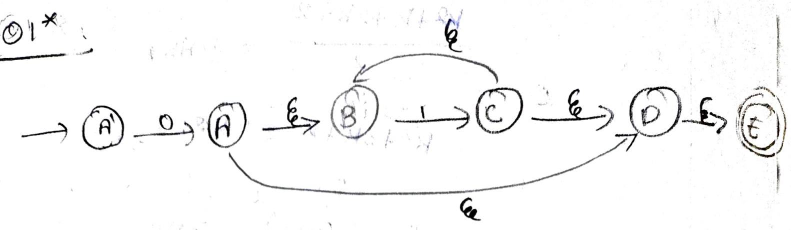
1) 0 1



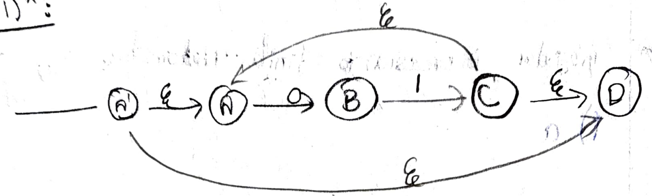
2) 0+1

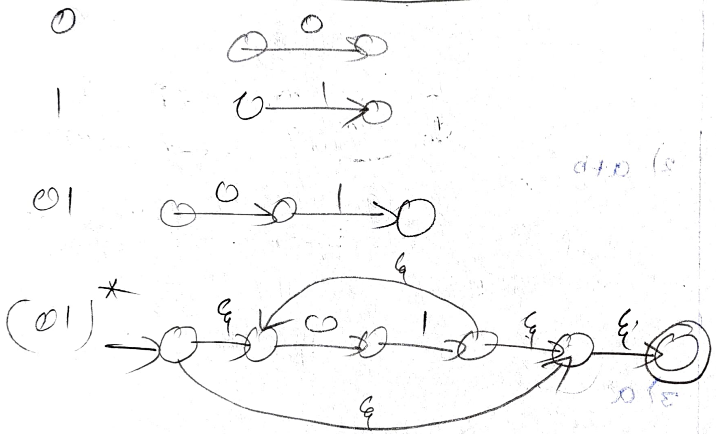


3) 01*:

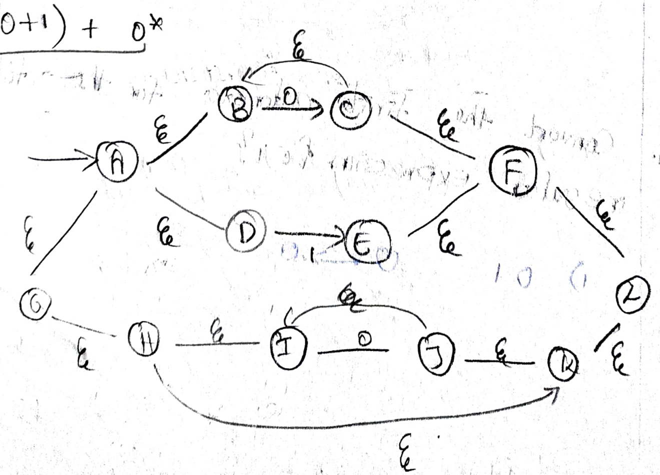


4) (01)*:





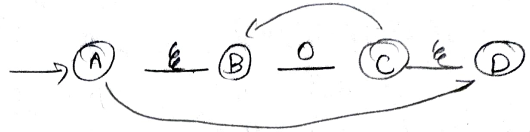
5) (0+1) + 0*



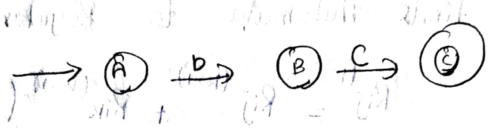
6) $a^* + bc$

a :

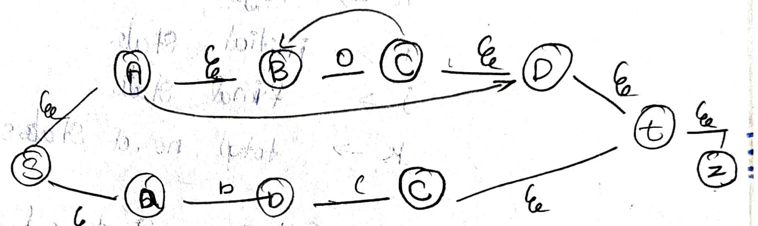
a^* :



bc :

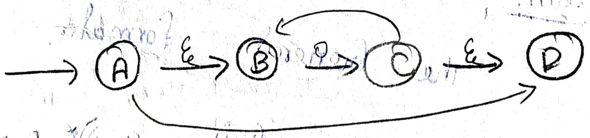


$a^* + bc$

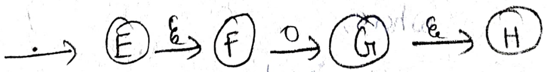


7) $0^* + 1^*$

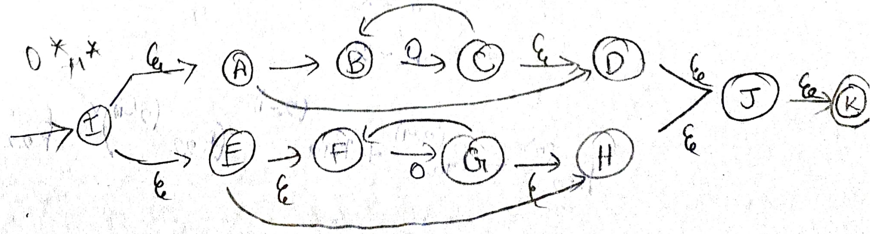
0^* :



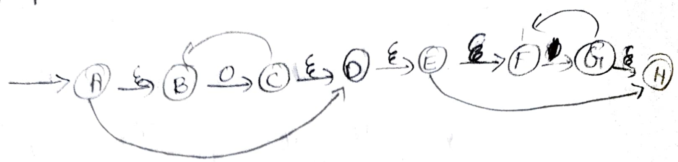
1^*



$0^*, 1^*$



8) $0^*, 1^*$



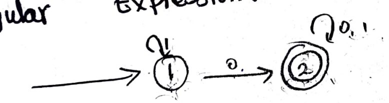
Finite Automata to Regular Expressions.

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

where,

- $R \rightarrow$ Regular Expression
- $i \rightarrow$ initial State
- $j \rightarrow$ final State
- $k \rightarrow$ total no. of States.

Convert the following finite Automata to regular Expression:



Soln:

The General formula,

$$\boxed{R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}}$$

where,

initial state $i = 1$

final state $j = 2$

total no. of state $K = 2$

$$R_{12}^{(2)} = R_{12}^{(2-1)} + R_{12}^{(2-1)} \left(R_{22}^{(2-1)}\right)^* R_{22}^{(2-1)}$$

$R_{12}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)} (R_{22}^{(1)})^* R_{22}^{(1)} \rightarrow (1)$

find the value for $R_{12}^{(1)}$ and $R_{22}^{(1)}$. so

we have find the value for all regular Expression.

$k=0, i=1, j=2$

$R_{ij}^{(k)} = R_{11}^{(0)} = 1$

$R_{12}^{(0)} = 0$

$R_{22}^{(0)} = 0+1$

$R_{21}^{(0)} = \phi$

$k=1, i=1, j=2$

$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)} (R_{11}^{(0)})^* R_{12}^{(0)}$

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

$= 0 + 1^* (0)$

$= 0[\epsilon + 1^*]$

$\boxed{R_{12}^{(1)} = 01^*}$

to find the $R_{22}^{(1)}$, so i, j, k are $i=2, j=2, k=1$

$R_{ik}^{k-1} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$

$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)} + (R_{11}^{(0)})^* R_{12}^{(0)}$

$= 0+1+\cancel{\phi} \; \cancel{(1)^*} \; 0$

$= 0+1$

$\boxed{R_{22}^{(1)} = 0+1}$

Sub in eqn (1)

$\Rightarrow R_{12}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)} (R_{22}^{(1)})^* R_{22}^{(1)}$

$= 01^* + 01^* \; (0+1)^* \; (0+1)$

$= 01^* + 01^* \; (0+1)^*$

$\boxed{\Rightarrow R_{12}^{(2)} = 01^* (0+1)^*}$

$= 01^* [\epsilon + (0+1)^*]$

The Regular Expression is $R_{12}^{(2)} = 01^*(0+1)^*$

formula:

1. $(\epsilon+1)^* = 1^*$
2. $1^* (\epsilon+1)^* = 1^*$
3. $(\epsilon+1)+1^* = 1^*$
4. $0+1^*0 = 1^*0$
5. $\epsilon+00^* = 0^*$
6. $(1+0)^* = (1^*0^*)^*$
7. $\phi . 0 = \phi$
8. $\phi + 0 = 0$
9. $\phi^* = \epsilon$
10. $\epsilon . 0 = 0$
11. $\epsilon + 0 = 0$
12. $11^* = 1^*$
13. $00^* = 0^*$
14. $RR^* = R^*$
15. $\epsilon + R^* = R^*$
16. $RR^* = R^* (R+\epsilon) = R^*$

minimization rules for Regular Expression:

1. $\phi + R = R$
2. $\phi R = R\phi = \phi$
3. $\epsilon R = R\epsilon = R$
4. $\epsilon^* = \epsilon$
5. $\phi^* = \epsilon$
6. $R+R = R$
7. $R^*R^* = R^*$
8. $RR^* = R^*R$
9. $(R^*)^* = R^*$
10. $\epsilon + RR^* = R^*$
11. $(P+Q)^* = (P^*Q^*) = P^* + Q^*$
12. $(PQ)^*P = P(QP)^*$
13. $(P+Q)R = PR + QR$
14. $R(P+Q)R = RP + RQ$
15. $R^+ + \epsilon = R^*$
16. $(R+\epsilon)^* = R^*$
17. $R^*R + R = R^*R$
18. $\phi = \epsilon = \epsilon$

1. Show that the language $L = \{a^n b^n / n \geq 1\}$ is not Regular:

Soln:

1) Let as Assume given language is a Regular language.

2) The total no. of State $L = \{a^n b^n / n \geq 1\}$ is

$n+n = 2n$

3) take one string 'W' and Calculate the length of the string $W = a^n b^n$

4) If the length of String $|W| \geq n$

$|a^n b^n| \geq n$

$2^n \geq n$

So, we can break into three string $W = xyz$

Let $W = a^i b^i$

Let as assume that

$xyz = a^i b^i$

$xy = a^m$

$z = a^{i-m} b^i$

$xyz = a^m a^{i-m} b^i$

$= a^i b^i$

To Prove $xyz = a^i b^i$

$xy = a^m$

$y = a^j$

$z = a^{i-m} b^i$

$xyz = a^m a^{i-m} b^i$

$\boxed{xyz = a^i b^i}$

5) (i) $|xy| \leq n$

$|a^m| \leq n$

$\boxed{m \leq n}$

(ii) $|y| > 0$ (or) $|y| \geq 1$

$|a^j| > 0$ (or) $|a^j| \geq 1$

$j > 0$ (or) $j \geq 1$

Since both the Condition are true for all $k \geq 0$, the string $k \geq 0$, xy^kz is also in 'L'

$\boxed{xy^kz = xyy^{k-1}z}$

$= a^m (a^j)^{k-1} a^{i-m} b^i$

$\boxed{xy^kz = (a^j)^{k-1} a^i b^i}$

Put $k=0 \Rightarrow a^{-j} a^i b^i \neq L$

Put $k=1 \Rightarrow a^i b^i = L$

Put $k=2 \Rightarrow a^j a^i b^i \neq L$

Since for $k=0$ and $k=2$ we have the string does not belongs to the language 'L'

So the language $\{a^n b^n / n \geq 1\}$ is not a Regular.

---

2. Show that the language $L = \{a^n / n = i^2, i \geq 1\}$ is not Regular:

1) Let as assume given language is a Regular language.

2) The total no. of State $L = \{a^n / n = i^2, i \geq 1\}$ is n.

3. take one String 'W' and ~~Calculate~~ Calculate the length of the String. $W = a^n$

4.) If the length of the ~~state~~ String $|W| \geq n$

$|a^n| \geq n$

$n \geq n$

So, we can break into three String

$W = xyz$ let $W = a^i$

To prove $xyz = a^i$

$xy = a^m$

$y = a^j$

$z = a^{i-m}$

$xyz = a^m . a^{i-m}$

$xyz = a^i$

5) (i) $|xy| \leq n$ (ii) $|v| > 0$ (or) $|y| > 1$

$|a^m| \leq n$ $|a^j| > 0$ (or) $|a^j| > 1$

$m \leq n$ $j > 0$ (or) $j > 1$

Since both the Condition are true for all $k \geq 0$, the String $k = 0$ $xy^k z$ is also in L

$\boxed{xy^k z = xyy^{k-1} z}$

$= a^m (a^j)^{k-1} a^{i-m}$

$= a^i (a^j)^{k-1}$

Put $k = 0 \Rightarrow a^i a^{-j} \notin L$

Put $k = 1 \Rightarrow a^i = L$

Put $k = 2 \Rightarrow a^i a^j \notin L$

Since for $k = 0$ and $k = 2$ we have the String does not belong to language L

So the language $L = \{a^n / n = i^2, i \geq 1\}$ is not regular.

9

1) Show that $L = \{0^n 1 0^n / n \geq 1\}$ is not a regular:

Soln:

1) let as assume that given language is a Regular language.

2) Calculate the total no. of State $L = \{0^n 1 0^n / n \geq 1\}$ is a n.

3) take one String 'W' and Calculate the length of the String $W = 0^n 1 0^n$

4) If the length of the String $|W| \geq n$

$|0^n 1 0^n| \geq n$

$2n+1 \geq n$

we can break into three String

$W = xyz$ let $W = 0^i 1 0^i$

assume that,

$xy = 0^m$

$y = 0^j$

$z = 0^{i-m} 1 0^i$

$xyz = 0^m 0^{i-m} 1 0^i$

$\boxed{xyz = 0^i 1 0^i}$

So our assumption is correct.

(5) (i) $|xy| \le n$ (ii) $|y| > 0$ (or) $|y| \ge 1$

$|0^m| \le n$ $|0^j| > 0$ (or) $|0^j| \ge 1$

$m \le n$ $j > 0$ (or) $j \ge 1$

Since both condition are true for all $k > 0$, the string xy^kz is also in 'L'

$xy^kz = xyy^{k-1}z$

$= \cancel{0^m (0^j)^{k-1} 0^{i-m}} 1 0^i$

$= 0^i 0^{j(k-1)} 1 0^i$

Put $k=0$ = $0^i 0^{-j} 1 0^i \notin L$

Put $k=1$ = $0^i 1 0^i = L$

Put $k=2$ = $0^i 0^j 1 0^i \notin L$

Since for $k=0$ & $k=2$ we have the string does not belong to language L

So the language $L = \{0^n 1 0^n / n \ge 1\}$ is not regular.

---

2) Show that the language $L = \{WW^R / W \in (a,b)\}$ is a not a regular

$W \in (a,b)$ $L = \{WW^R / W \in (a,b)\}$

$W = a^n b^n$ $L = \{a^n b^n b^n a^n / n \ge 1\}$

$W^R = b^n a^n$

1) let as assume that given language is a Regular language.

2) The total no of state $L = \{a^n b^n b^n a^n / n \ge 1\}$ is n

3) take one string 'w' and calculate the length of the string $W = a^n b^n b^n a^n$

4) If the length of the string $|W| \ge n$

$|a^n b^n b^n a^n| \ge n$

$4n \ge n$

we can break into three string

$\boxed{W = xyz}$ let $W = a^i b^i b^i a^i$

assume that, $xy = a^m$

$y = a^j$

$z = a^{i-m} b^i b^i a^i$

$xyz = a^m a^{i-m} b^i b^i a^i$

$\boxed{xyz = a^i b^i b^i a^i}$

So our assumption is correct

(5) (i) $|xy| \le n$ (ii) $|y| > 0$ (or) $|y| > 1$

$|a^m| \le n$ $|a^j| > 0$ (or) $|a^j| > 1$

$m \le n$ $j > 0$ (or) $j > 1$

Since both condition are true for all $k > 0$, the string xy^kz is also in 'L'

$xy^kz = xyy^{k-1}z$

$= a^{m} a^{j(k-1)} a^{l-m} b^i b^i a^i$

$\boxed{xy^kz = a^{j(k-1)} a^i b^i b^i a^i}$

Put $k=0 \Rightarrow a^{-j} a^i b^i b^i a^i \neq L$

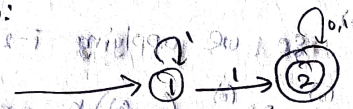
$k=1 \Rightarrow a^i b^i b^i a^i = L$

$k=2 \Rightarrow a^j a^i b^i b^i a^i \neq L$

Since for $k=0$ & $k=2$ we have the string does not belong to language L

So the language $L = \{ww^R / w \in (a,b)\}$ not Regular.

Construct the following finite automata to regular Expression:



Soln:

The General formula

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} (R_{kk}^{(k-1)})^* R_{kj}^{(k-1)}$$

$i=1, j=2, k=2.$

$$R_{12}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)} (R_{22}^{(1)})^* R_{22}^{(1)} \to ①$$

Find the value of $R_{12}^{(1)}$ and $R_{22}^{(1)}$ so we have find the values for all regular expressions.

$k=0, i=1, j=2$

$R_{ij}^{(k)} \Rightarrow R_{12}^{(0)} = 1$

$R_{11}^{(0)} = 1$

$R_{22}^{(0)} = 0+1$

$R_{21}^{(0)} = \phi$

$K=1, i=1, j=2$

$$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)} (R_{11}^{(0)})^* R_{12}^{(0)}$$

$= 1+1(1^*)1$

$= 1+1^*1$

$= 1[\epsilon+1^*]$

$= 1^*$

$\boxed{R_{12}^{(1)} = 1^*} \to ②$

to find $R_{22}^{(1)}$ so we applying $i=2, j=2, k=1$

$$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)} (R_{11}^{(0)})^* R_{12}^{(0)}$$

$= 0+1 + \underbrace{\phi\,(1)^*\,(1)}_{\phi}$

$= 0+1$

$\boxed{R_{22}^{(1)} = 0+1} \to ③$

Sub in eqn ①

$① = R_{12}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)} (R_{22}^{(1)})^* R_{22}^{(1)}$

$R_{12}^{(2)} = 1^* + 1^*(0+1)^*(0+1)$

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

$= 1^* [\varepsilon + (0+1)^*]$

$R_{12}^{(2)} = 1^* [0+1]^*$

The RE $\Rightarrow 1^* (0+1)^*$

Pumping lemma:

The Properties of regular languages are

1) Pumping lemma

2) Closure Properties.

1) Pumping lemma:

The Pumping lemma is a way to Prove that sudden language are not regular.

Steps in Pumping lemma:

Steps:

1) Assume that given language is a regular language.

2. Calculate the no. of states

3. Take one string 'w' & Calculate the length of the string

4. If the length of the string $|w| \geq n$, we can break into 3 strings $w = xyz$ (or) $z = uvw$

& i) $|xy| \leq n$

ii) $|v| > 0$ (or) $|v| \geq 1$

iii) for all $k \geq 0$, xy^kz is also in language 'L'

Show that the language $L = \{ww^R / w \in (0,1)\}$ is a not Regular

$w \in (0,1)$
$w = 0^n 1^n$
$w = 0^n 1^n$

$L = \{ww / w \in (0,1)\}$
$L = \{0^n 1^n 0^n 1^n / n \geq 1\}$

Soln:

1) let as assume that given language is a Regular language.

2) The total no of state $L = \{0^n 1^n 0^n 1^n / n \geq 1\}$ is n

3) take one string 'w' & Calculate the length of the string $W = 0^n 1^n 0^n 1^n$

4) If the length of the string $|W| \geq n$

$|0^n 1^n 0^n 1^n| \geq n$

$4n \geq n$

we can break into three string

$W = x, y, z$ let $W = 0^i 1^i 0^i 1^i$

assume that, $xy = a^m$

$y = a^j$

$z = a^{i-m} 1^i 0^i 1^i$

$xyz = a^m 0^{i-m} 1^i 0^i 1^i$

$\boxed{xyz = 0^i 1^i 0^i 1^i}$

So our assumption is Correct

(5) (i) $|xy| \leq n$

$|a^m| \leq n$

$m \leq n$

(ii) $|y| > 0$ (or) $|v| > 1$

$|a^j| > 0$ (or) $|0^j| > 1$

$j > 0$ (or) $j > 1$

Since both Condition are true for all $k>0$, the String xy^kz is also in 'L'

$$xy^kz = xyy^{k-1}z$$

$$= 0^{\cancel{k}} 0^{j(k-1)} 0^{i-\cancel{k}} 1^i 0^i 1^i$$

$$\boxed{xy^kz = 0^{j(k-1)} 0^i 1^i 0^i 1^i}$$

Put $k=0$ $\quad 0^{-j} 0^i 1^i 0^i 1^i \neq L$

$k=1$ $\quad 0^i 1^i 0^i 1^i = L$

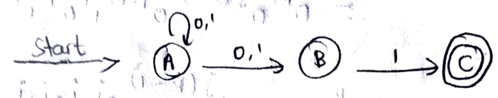
$k=2$ $\quad 0^j 0^i 1^i 0^i 1^i \neq L$

Since for $k=0 \neq k=2$ we have the String does not belong to Language 'L'

So the language $L = L = \{ww / w \in (0,1)\}$ not Regular.

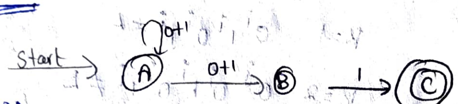
State elimination technique:

1) Find the regular expression for the given finite automata using state elimination technique.

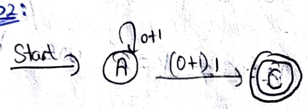


Soln:

Step 1:



Step 2:



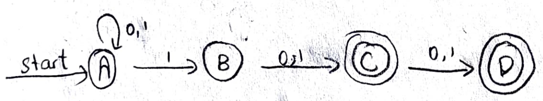
State 'B' is Removed

$RE \Rightarrow (0+1)^* (0+1) 1$

(or)

$C = (0+1)^* (0+1) 1$

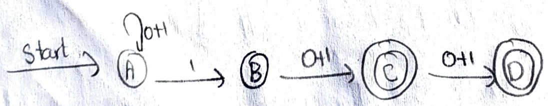
2. Find the regular Expression for the given finite automata using state elimination technique.



Soln:

Regular Expression $= (A \rightarrow C) + (A \rightarrow D)$

Step 1:



Step 2:

eliminate (B) in ~~node~~ state $(A \to C)$

$$\overset{\circlearrowleft\, 0+1}{A} \xrightarrow{1(0+1)} \text{(C)}$$

Step 3:

eliminate (C) is ~~node~~ state $(A \to D)$

$$\rightarrow \overset{\circlearrowleft\, 0+1}{A} \xrightarrow{1(0+1)} \text{(C)} \xrightarrow{0+1} \text{(D)}$$

Step 4:

$$\xrightarrow{\text{state}} \overset{\circlearrowleft\, 0+1}{A} \xrightarrow{1(0+1)(0+1)} \text{(D)}$$

$RE = (0+1)^*\, 1\, (0+1)\cdot(0+1)$

~~$D = (0+1)^* 1 (0+1)(0+1)$~~

Regular Expression $= (A \to C) + (A \to D)$

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

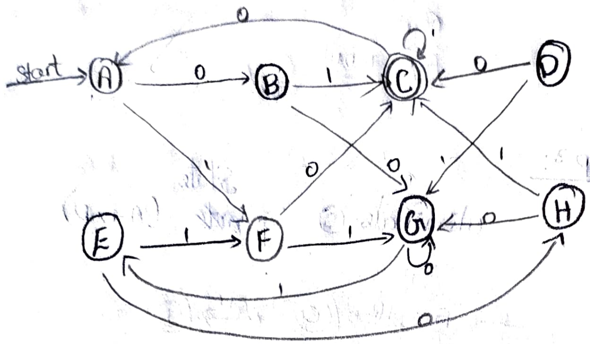
$\boxed{D = (0+1)^* 1 (0+1)(0+1)}$

Equivalance & minimization of finite automata

Table filling algorithm

DFA minimization Algorithm.

1. Construct DFA using DFA minimization Algorithm for the given finite Automata.



Soln:

transition table above DFA

	0	1
→ A	B	F
B	G	C
* C	A	C
D	C	G
E	H	F
F	C	G
G	G	E
H	G	C

now, the Pair state {D, F} as the same state for the transition of 0 & 1

Similarly, the pair of {B, H} as the Same state for the transition of 0 & 1

State are {A, B, C, D, E, F, G, H}

[A, (B,H), C, (D,F), E, G]

Since 'C' is the accepting state, it is always kept separately. now, the new transition table.

	0	1
→ A	(B,H)	(D,F)
(B,H)	G	C
* C	A	C
(D,F)	C	G
E	(B,H)	(D,F)
G	G	E

now, the state (A,E) as the same state of the transistion 0&1.

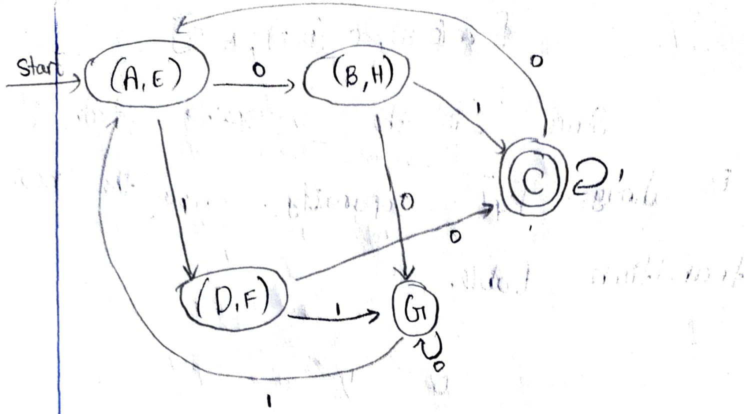
So the new state are

[(A,E) (B,H) C, (D,F), G]

the new transition table

	0	1
→ (A,E)	(B,H)	(D,F)
(B,H)	G	C
* C	(A,E)	C
(D,F)	C	G
G	G	(A,E)

~~DFA is a~~ The minimization of finite automata is



2m

Closure Property of Regular languages:

If Sudden language are Regular and a Language 'L' is formed from by Sudden Operations then 'L' is also Regular.

Closure Properties:

1. Union of two regular Languages is regular.

2. Intersection of two Regular Language is regular.

3. Complement of Regular Languages Is regular.

4. difference of two Regular Languages is regular

5. Reversal of regular Language is regular.

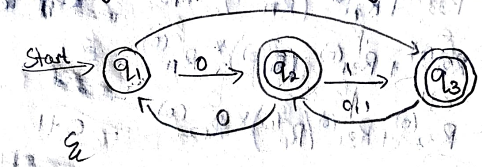
6. Closure of regular language is regular.

7. Concatenation of regular language is regular.

8. homomorphism of regular language is regular.

9. The inverse homomorphism of regular Language is regular.

1. Convert the given DFA to regular Expression:



Soln:

The General formula is

$$R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$$

i → q_1 → 1

j → q_2, q_3 → 2,3

k → 3 → 3

$R = R_{12}^{(3)} + R_{13}^{(3)}$

$R_{12}^{(3)} = R_{12}^{(2)} + R_{13}^{(2)} \left(R_{33}^{(2)}\right)^* R_{32}^{(2)}$

$R_{13}^{(3)} = R_{13}^{(2)} + R_{13}^{(2)} \left(R_{33}^{(2)}\right)^* R_{33}^{(2)}$

Put k=0

$R_{11}^{(0)} = \varepsilon$ $\quad R_{33}^{(0)} = \varepsilon$

$R_{12}^{(0)} = 0$ $\quad R_{31}^{(0)} = \phi$

$R_{13}^{(0)} = 1$ $\quad R_{32}^{(0)} = 0+1$

$R_{22}^{(0)} = \varepsilon$

$R_{21}^{(0)} = 0$

$R_{23}^{(0)} = 1$

Put k=1 $\quad R_{ij}^{(k)} = R_{ij}^{(k-1)} + R_{ik}^{(k-1)} \left(R_{kk}^{(k-1)}\right)^* R_{kj}^{(k-1)}$

$R_{11}^{(1)} = R_{11}^{(0)} + R_{11}^{(0)} \left(R_{11}^{(0)}\right)^* R_{11}^{(0)} = \boxed{\varepsilon}$ $\quad \varepsilon + \varepsilon(\varepsilon)^*\varepsilon$

$R_{12}^{(1)} = R_{12}^{(0)} + R_{11}^{(0)} \left(R_{11}^{(0)}\right)^* R_{12}^{(0)} = \boxed{0}$ $\quad 0 + \varepsilon(\varepsilon)^* 0$

$R_{13}^{(1)} = R_{13}^{(0)} + R_{11}^{(0)} \left(R_{11}^{(0)}\right)^* R_{13}^{(0)} = \boxed{1}$ $\quad 1 + \varepsilon(\varepsilon)^* 1$

$R_{22}^{(1)} = R_{22}^{(0)} + R_{21}^{(0)} \left(R_{11}^{(0)}\right)^* R_{12}^{(0)} = \boxed{00}$ $\quad \varepsilon + 0(\varepsilon)^* 0$

$R_{21}^{(1)} = R_{21}^{(0)} + R_{21}^{(0)} \left(R_{11}^{(0)}\right)^* R_{11}^{(0)} = \boxed{0}$ $\quad 0 + 0(\varepsilon)^*\varepsilon$

$R_{23}^{(1)} = R_{23}^{(0)} + R_{21}^{(0)} \left(R_{11}^{(0)}\right)^* R_{13}^{(0)} = \boxed{1+01}$ $\quad 1 + 0(\varepsilon)^* 1$

$R_{33}^{(1)} = R_{33}^{(0)} + R_{31}^{(0)} \left(R_{11}^{(0)}\right)^* R_{13}^{(0)} = \boxed{\varepsilon}$ $\quad \varepsilon + \phi(\varepsilon)^* 1$

$R_{31}^{(1)} = R_{31}^{(0)} + R_{31}^{(0)} \left(R_{11}^{(0)}\right)^* R_{11}^{(0)} = \boxed{0}$ $\quad 0 + \phi(\varepsilon)^*\varepsilon$

$R_{32}^{(1)} = R_{32}^{(0)} + R_{31}^{(0)} \left(R_{11}^{(0)}\right)^* R_{12}^{(0)} = \boxed{0+1}$ $\quad (0+1) + \phi(\varepsilon)^* 0$

Put k=2

$R_{11}^{(2)} = R_{11}^{(1)} + R_{12}^{(1)} \left(R_{22}^{(1)}\right)^* R_{21}^{(1)} = (00)^*$ $\quad \varepsilon + 0(00)^* 0$

$R_{12}^{(2)} = R_{12}^{(1)} + R_{12}^{(1)} \left(R_{22}^{(1)}\right)^* R_{22}^{(1)} = 0(00)^*$ $\quad 0 + 0(00)^* 00$

$R_{13}^{(2)} = R_{13}^{(1)} + R_{12}^{(1)} \left(R_{22}^{(1)}\right)^* R_{23}^{(1)} = 1(00)^*$ $\quad 1 + 0(00)^* 1+01$

$R_{22}^{(2)} = R_{22}^{(1)} + R_{22}^{(1)} \left(R_{22}^{(1)}\right)^* R_{22}^{(1)} = (00)^*$ $\quad 00 + 00(00)^* 00$

$R_{21}^{(2)} = R_{21}^{(1)} + R_{22}^{(1)} \left(R_{22}^{(1)}\right)^* R_{21}^{(1)} = 0(00)^*$ $\quad 0 + (00)(00)^* 0$

$R_{23}^{(2)} = R_{23}^{(1)} + R_{22}^{(1)} \left(R_{22}^{(1)}\right)^* R_{23}^{(1)} = 10^*$ $\quad (1+01) + (00)(00)^*(1+01)$

$R_{33}^{(2)} = R_{33}^{(1)} + R_{32}^{(1)} \left(R_{22}^{(1)}\right)^* R_{23}^{(1)} = (0+1)0^*,$ $\quad \varepsilon + (0+1)(00)^*(1+01)$

$R_{31}^{(2)} = R_{31}^{(1)} + R_{32}^{(1)} \left(R_{22}^{(1)}\right)^* R_{21}^{(1)} = (0+1)00^*$ $\quad 0 + (0+1)(00)^* 0$

$R_{32}^{(2)} = R_{32}^{(1)} + R_{32}^{(1)} \left(R_{22}^{(1)}\right)^* R_{22}^{(1)} = (0+1)00^*$ $\quad (0+1) + (0+1)(00)^* 00$

$R_{12}^{(3)} = R_{12}^{(2)} + R_{13}^{(2)} (R_{33}^{(2)})^* R_{32}^{(2)}$

$= 0(00)^* + 1(00)^* [(0+1)\ 0^*1]^* (0+1)\ 00^* \to ②$

$R_{13}^{(3)} = 1(00)^* [(0+1)\ 0^*1]^* (0+1)\ 0^*1 \to ③$

$RE = 0(00)^* + 10^* [(0+1)\ 0^*1\ (0+1)(00)^* + ((0+1)\ 0^*1)^*\ 0+1]$

$1+2+\dots+n = \frac{n(n+1)}{2}$ (or) $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$

Soln:

$n=1$

$1 = \frac{1(1+1)}{2} = \frac{2}{2} = 1$

LHS = RHS

$n=2$

$1+2 = \frac{2(2+1)}{2} = \frac{2(3)}{2} = \frac{6}{2} = 3$

$3=3$ LHS = RHS

If n is true then k is also true

$1+2+\dots+k = \frac{k(k+1)}{2}$

By using Induction method

k is true then (k+1) is also true

$1+2+3+4\dots+(k+1) = \frac{(k+1)\ ((k+1)+1)}{2}$

$1+2+\dots+(k+1) = \frac{(k+1)(k+2)}{2}$

LHS $= 1+2+\dots+(k+1)$

$= 1+2+\dots+k+(k+1)$

$= \frac{k(k+1)}{2} + (k+1)$

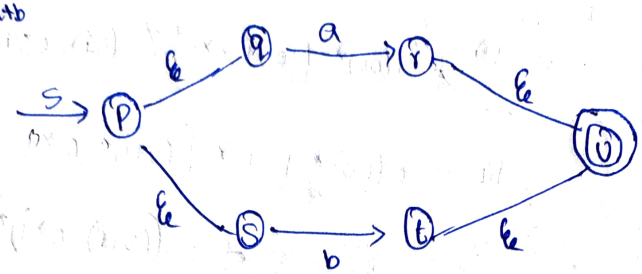
$= \frac{k(k+1)+2(k+1)}{2}$

$= \frac{(k+1)(k+2)}{2}$ = RHS.

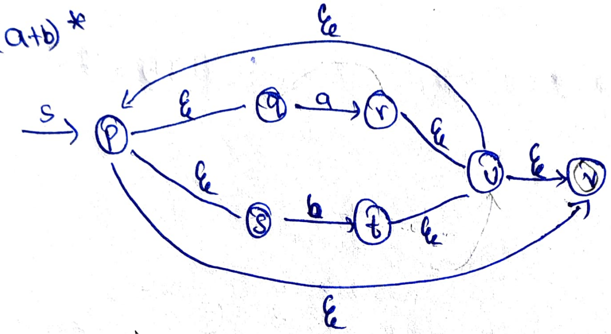
Regular Expression to finite Automata:

1) $(a+b)^*\ c+d$

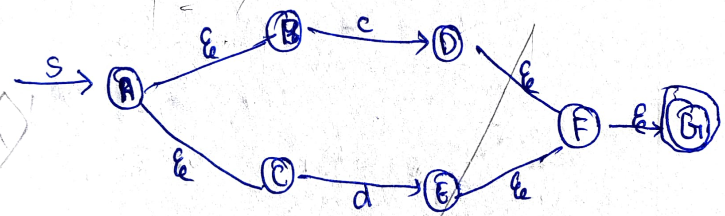
a+b



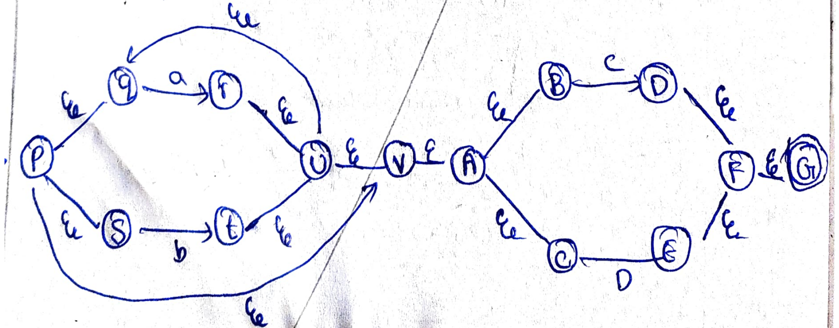
$(a+b)^*$



c+d

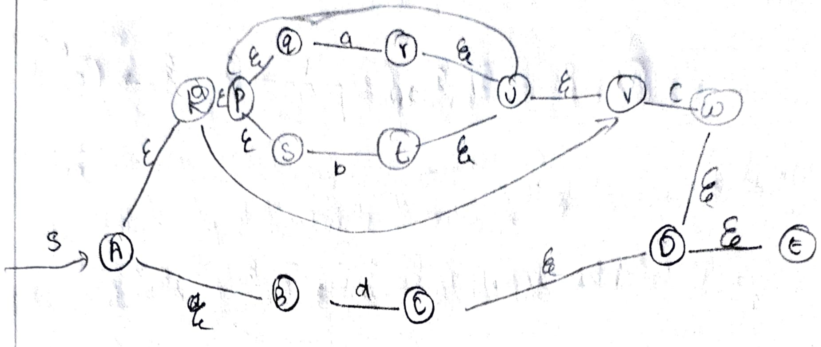


$(a+b)^* c+d$



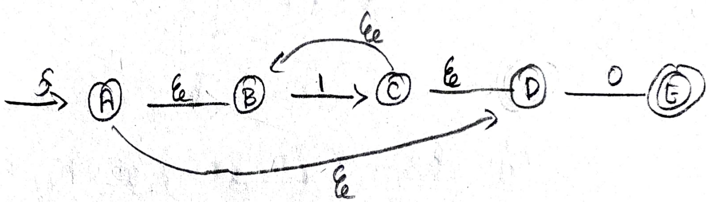
2. ~~abc* + d~~

(a+b)*c+d

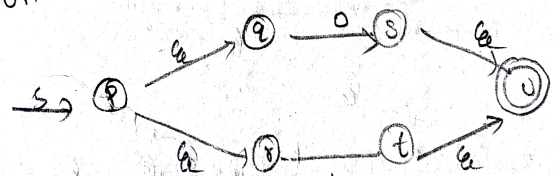


2. ~~abc* + d~~ (1*0) + (0+1)

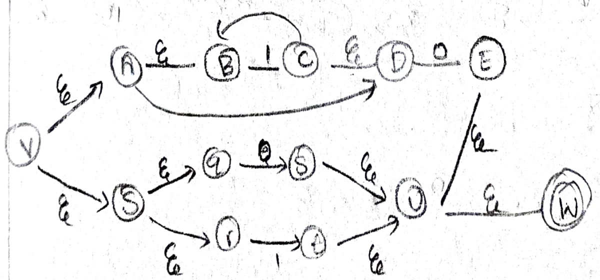
1*0



0+1

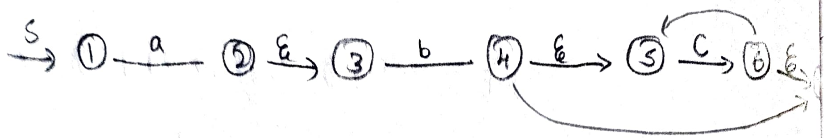


(1*0) + (0+1)

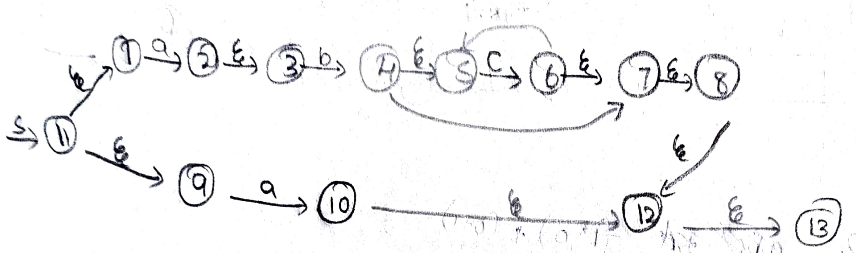


③ abc*+d

abc*

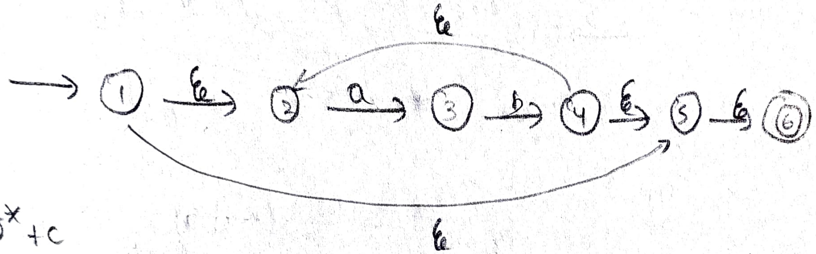


abc*d

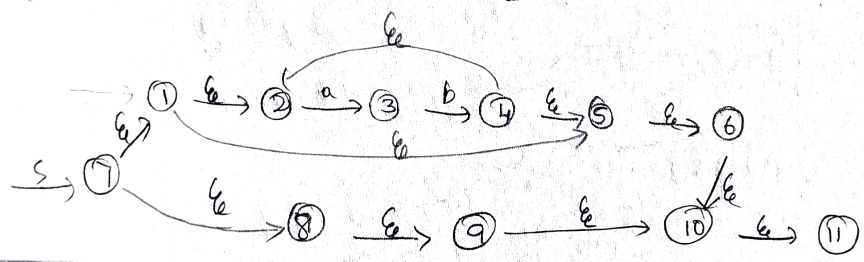


④ (ab)* + c

(ab)*

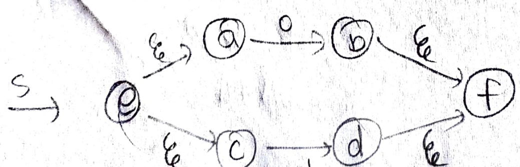


(ab)* + c

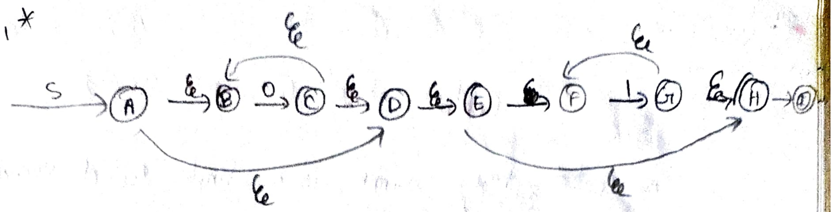


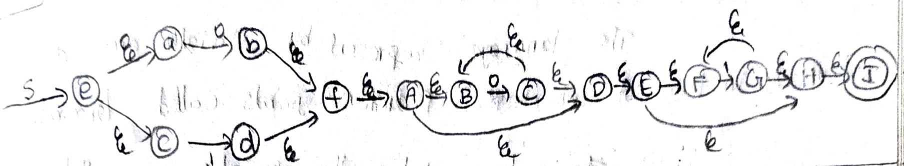
⑤ (0+1) 0* 1*

0+1



$0^*, 1^*$





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