Regular Expressions:

The lang. accepted by F.A are easily described by simple expression c/d regular expression. The regular expression is the most effective way to represent any lang. The lang accepted by some regular expression is known as regular lang.

R E Definition:

If Σ is an alphabet which is used to denote the input set, then the regular expression over this alphabet Σ can be defined as follows:

Properties of Regular Expression:

(1) ϕ is a regular expression which denotes the empty set.

(2) ε is a regular expression & denotes the set $\{\varepsilon\}$ and it is an Empty string.

(3) For each 'a' in Σ, 'a' is a regular expression and denotes the set $\{a\}$.

(4) If R_1 and R_2 are two regular expression, then Union of these represented as $r_1 \cup r_2$ or $r_1 + r_2$ is also a regular expression.

5) If r_1 and r_2 are two regular expression, then Concatenation is also regular expression and representation as $r_1 . r_2$.

6) The Kleene closure of regular expression ~~are~~ 'r' is denoted by r^* is also a regular expression

7) If 'r' is a regular expression then (r) is also a regular expression.

Regular set:

A set represented for a regular Expression is c/d a regular set. Regular lang. are also represented in set notation. We use the { } curly brackets while representing the set.

For example - If $\Sigma = \{a, b\}$ be an alphabet then

Regular Set	Regular Expression.
$\{\}$	ϕ
$\{a\}$	a
$\{a, b\}$	$a+b$
$\{\epsilon, a, aa, aaa --\}$	a^*
$\{a, aa, aaa, aaaa --\}$	a^+
$\{\epsilon\}$	ϵ
$\{00, 10, 11, 01\}$	$00+10+11+01$

Example : If $\Sigma = \{$all the words over $\{a, b\}\}$.

$(a+b)^* = \{\epsilon, a, b, ab, ba, aab, baa, abab --\}$.

Ques A lang. consisting of all strings over $\{a, b\}$ ending with a. find the regular Expression.

Soln $(a+b)^*.a$

Ques A lang consisting of all the words over $\{a, b\}$ ending with aa.

Soln $(a+b)^* a.a$

Ques Construct the regular Exp. for the lang. containing all strings having any no. of a's and b's. ~~expect~~ except null string.

Soln $R.E = (a+b)^*$

Ques Construct the regular exp. for the lang. Starting & ending with a and having any combination of b's in b/w.

Soln $R.E = ab^*a$

Ques Find the regular Exp to the lang. of all string over the alphabet {0,1} that contains atleast two 0's.

Soln $R.E = (0+1)^*0(0+1)^*0(0+1)^*$

Ques The set of all strings with even no. of a's followed by an odd no. of b's.

Soln $(aa(aa)^* + b(bb)^*)^*$

Ques The set of all strings ending with bb or single a.

Soln $((a+b)^*bb + (a+b)^*a)$

Ques Write a RE to denote the lang. L over Σ^* where $\Sigma = \{a, b\}$ such that the third symbol from the right end of the string is always a.

Ques Write a regular expression for the following langs over the alphabet $\{0,1\}$. The set of all strings in which every pair of adjacent zeroes (0's) appears before any pair of adjacent 1's.

Ques Write regular expression for following lang.

(i) $L = \{b^m a^n \mid (n+m) \text{ is even}\}$

(ii) $L = \{w \in (a,b)^* \mid (\text{no. of a's in } w) \bmod 3 = 0\}$

Sol^n (i) $L = \{\varepsilon, ab, aabb, abbb, aaab, aaabbb - -\}$.

$$RE = [a(aa)^* b (bb)^*]^* + [(aa)^*(bb)^*]$$

Sol^n (ii) $L = \{b, ababab, aaabbbaaab, - -\}$.

$$R.E = [b^*ab^*ab^*ab^*]^* + b^*$$

Ques $L = \{ab^n w \mid n \geq 3, w \in (a+b)^+\}$

Sol^n $R.E = ab^3 b^* (a+b)^+$

Ques $L = \{w : |w| \bmod 3 = 0, w \in (a,b)^*\}$

Sol^n $RE = [(a+b)^3]^*$

Ques $L = \{a^n b^m \mid n \geq 4, m \leq 3\}$

Sol^n $R.E = a^4 a^* (\varepsilon + b + b^2 + b^3)$

Ques $L = \{a^{2n} b^{2m+1} \mid n \geq 0, m \geq 0\}$

Sol^n $RE = (aa)^*(bb)^* b$

CamScanner

Q All words that end with double letter over $\Sigma = \{a,b\}$

Sol^n $RE = (a+b)^*aa + (a+b)^*bb$

$RE = (a+b)^*[aa+bb]$

Operators of Regular Expression & their Precedence:

There are 3 main operator that can be use in R.E are Union, Concatenation and Kleene cl. For R.E, the following is the order of precedence for the operators.

(1) The star operator (or Kleene closure):

It having highest precedence.

(2) Dot operator (or concatenation):

After star operator comes dot operator a/c to precedence

(3) Union operator:

Final precedence is given to Union.

Algebric Laws for Regular Expression:

let P, Q and R are R.E then the identity rules are as given below -

$I_1 \rightarrow \epsilon R = R\epsilon = R$

$I_2 \rightarrow \epsilon^* = \epsilon$

$I_3 \rightarrow (\phi)^* = \epsilon$

$I_4 \rightarrow \phi R = R\phi = \phi$

$I_5 \rightarrow \phi + R = R$

$I_6 \rightarrow R + R = R$

$I_7 = RR^* = R^*R = R^+$

$I_8 \rightarrow (R^*)^* = R^+$

$I_9 \rightarrow \epsilon + RR^* = R^* = \epsilon + R^*R$

$I_{10} \rightarrow (P+Q)^* = (P^*Q^*)^* = (P^* + Q^*)^*$

$I_{11} \rightarrow (P+Q)R = PR + QR$

$I_{12} \rightarrow R^*(\epsilon + R) = (\epsilon + R)R^* = R^*$

$I_{13} \rightarrow (R+\epsilon)^* = R^*$

$I_{14} \rightarrow \epsilon + R^* = R^*$

$I_{15} \rightarrow (PQ)^*P = P(QP)^*$

$I_{16} \rightarrow R^*R + R = R^*R$

$I_{17} \rightarrow R^*R^* = R^*$ //

Ques Prove that the RE, $R = \wedge + 1^*(011)^*(1^*(011)^*)^*$ also describes the same string $(1+011)^* = \epsilon + 1^*(011)^*(1^*(011)^*)^*$

Sol^n $RHS = \epsilon + 1^*(011)^*(1^*(011)^*)^*$ let $P_1 = 1^*(011)^*$

$= \epsilon + P_1 P_1^*$ (using identity $\epsilon + RR^* = R^*$)

$= P_1^*$

$= (1^*(011)^*)^*$ [let $P_2 = 1$ and $P_3 = (011)$]

$= (P_2^* P_3^*)^*$ [$\because (P^*Q^*)^* = (P+Q)^*$]

$= (P_2 + P_3)^*$

$= (1+011)^*$ <u>LHS</u>

Ques Prove that $(1+00^*1) + (1+00^*1)(0+10^*1)^*(0+10^*1) = 0^*1(0+10^*1)^*$

Sol^n $(1+00^*1)+(1+00^*1)(0+10^*1)^*(0+10^*1) = LHS$

$= (1+00^*1)[\epsilon + (0+10^*1)^*(0+10^*1)]$ [$\because \epsilon + RR^* = R^*$]

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

$= [\epsilon + 00^*]1[0+10^*1]^*$ [$\because \epsilon + RR^* = \epsilon + R^*R = R^*$]

$= 0^*1[0+10^*1]^*$
<u>RHS</u>.

Ques Prove that $P + PQ^*Q = a^*bQ^*$ where $P = b + aa^*b$ and Q is any RE.

Sol^n. $LHS = P + PQ^*Q$

$= P(\epsilon + Q^*Q)$ [$\because \epsilon + R^*R = R^*$]

$:= P(\theta^*)$

$= (b+aa^*b)\theta^*$

$= b(\epsilon + aa^*)\theta^*$

$= ba^*\theta^*$ RHS.

Ques Simplify the following RE $r_1(r_1^* r_1 + r_1^*) + r_1^*$

Soln $\Rightarrow r_1[r_1^*(r_1+\epsilon)] + r_1^*$ $[R^*(\epsilon+R) = R^*]$

$\Rightarrow r_1 r_1^* + r_1^*$

$\Rightarrow r_1^*[r_1+\epsilon]$ $[R^*(\epsilon+R) = R^*]$

$\Rightarrow r_1^*$

Ques Simplify $RE = (r_1+r_2+r_1r_2+r_2r_1)^*$

Soln $= (r_1+r_2+r_1r_2+r_2r_1)^*$

$= (r_1+r_2+r_2(r_1+r_1))^*$

$= (r_1+r_2+r_2r_1)^*$

OR

$= \{\epsilon, r_1, r_2, r_1r_2, r_2r_1 - - \}$

$= \{$ any combination of r_1 and $r_2 \}$.

$= \{r_1+r_2\}^*$

Ques show that $(r+s)^* \neq (r^*+s^*)$

Soln LHS $= (r+s)^*$

$= \{\epsilon, r, s, rs, sr, srs, rrs - - \}$

RHS $= r^*+s^*$

$= \{\epsilon, r, rr, rrr, --\} \cup \{\epsilon+s, ss, sss --\}$.

$= \{\epsilon, r, s, rr, ss, rrr, sss - - \}$.

LHS $\neq$ RHS.

Construction of R.E from DFA:

Arden's Theorem:-

let P and Q be two RE over alphabet Σ. If P does not contain null string "ϵ", $R = Q + RP$ has a unique soln i.e, $R = QP^*$. It can be understand as: $R = Q + RP$ put the value of R in RHS

$$R = Q + (Q+RP).P = Q + QP + RP^2$$

When we put the value of R again & again, we got the following eqn

$$R = Q + QP + QP^2 + QP^3 + \text{---}$$

$$R = Q[\epsilon + P + P^2 + P^3 + \text{---}]$$

$$R = QP^*$$

Condition of Arden's Theorm for F.A:

(1) The FA should not contain ϵ moves.

(2) There should exactly one initial state.

(3) There should be more than one final states.

Ques Consider the DFA shown in fig. & find the RE recognized by this.

Soln

$$q_0 = q_0 a + q_0 b + \epsilon$$

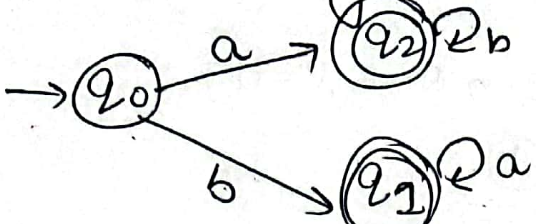
$$q_0 = \epsilon + q_0(a+b)$$

$R = Q + RP$ has soln $R = QP^*$

$q_0 = \epsilon(a+b)^*$ where $R = q_0$, $Q = \epsilon$, and $P = (a+b)$

$$q_0 = (a+b)^*$$

Ques Find the RE recognised by DFA shown in fig.



Sol^n

$q_0 = \varepsilon$ —①

$q_1 = q_0 b + q_1 a$ —②

$q_2 = q_0 a + q_2 b$ —③

put ~~the~~ value of q_0 from eqn ① to eqn ②

$q_1 = \varepsilon . b + q_1 a$ [using Arden's theorem $R = Q + RP$ has a unique sol^n $R = QP^*$]

$R = Q + RP$

where $R = q_1$, $Q = \varepsilon . b$, $P = a$

$q_1 = \varepsilon . ba^*$

$q_1 = ba^*$ —④

put the value of q_0 from eqn ① to eqn ③

$q_2 = \varepsilon . a + q_2 b$ [using the Arden's theorem]

$q_2 = a + q_2 b$

$R = Q + RP$

$R = q_2$, $Q = a$ and $P = b$

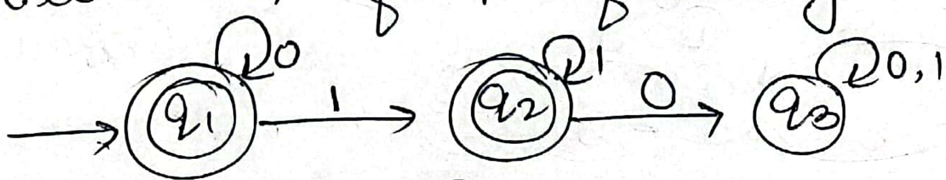
$q_2 = ab^*$ —⑤

$RE =$ eqn ④ ~~and~~ + eqn ⑤

$= q_1 + q_2$

$= ba^* + ab^*$ Ans

Ques Find ~~the~~ RE for the following FA



$q_1 = q_1 0 + \varepsilon$ —①

$q_2 = q_1 1 + q_2 1$ —②

$q_3 = q_2 0 + q_3 0 + q_3 1$ —③

not

Now apply Arden's Theorem to eqn ①

$q_1 = \varepsilon + q_1 0$

$R = Q + RP$ $[R = q_1, Q = \varepsilon, P = 0]$

$q_1 = \varepsilon 0^* = 0^*$ — ④

Now put the value of q_1 from eqn ④ to eqn ②

$q_2 = q_2 1 + q_1 1$

$q_2 = q_1 1 + 0^* 1$

$q_2 = 0^* 1 + q_2 1$

$R = Q + RP$ $[R = q_2, Q = 0^*1, P = 1]$

$R = QP^*$

$q_2 = 0^*11^*$ — ⑤

RE = eqn ④ + eqn ⑤

$= q_1 + q_2$

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

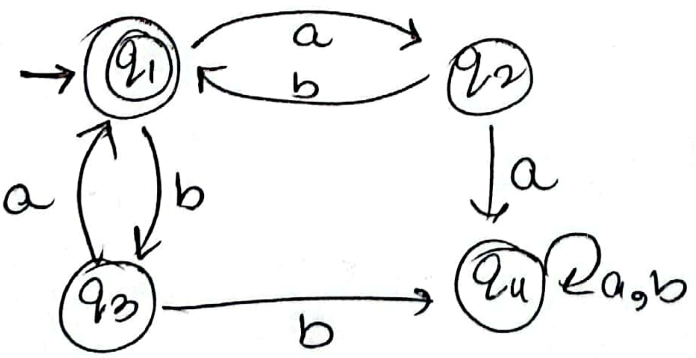
$= 0^*[\varepsilon + 11^*]$ $[\because \varepsilon + RR^* = R^*]$

$= 0^* 1^*$

$= 0^*11^*$

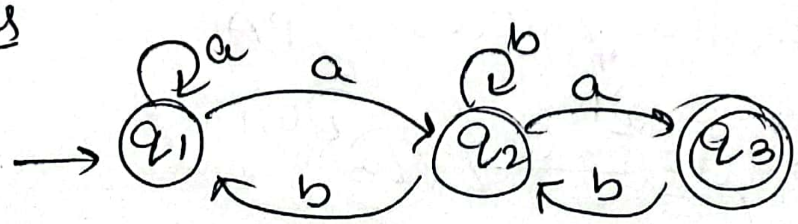
Ans

Ques



Sol^n $(ab+ba)^*$

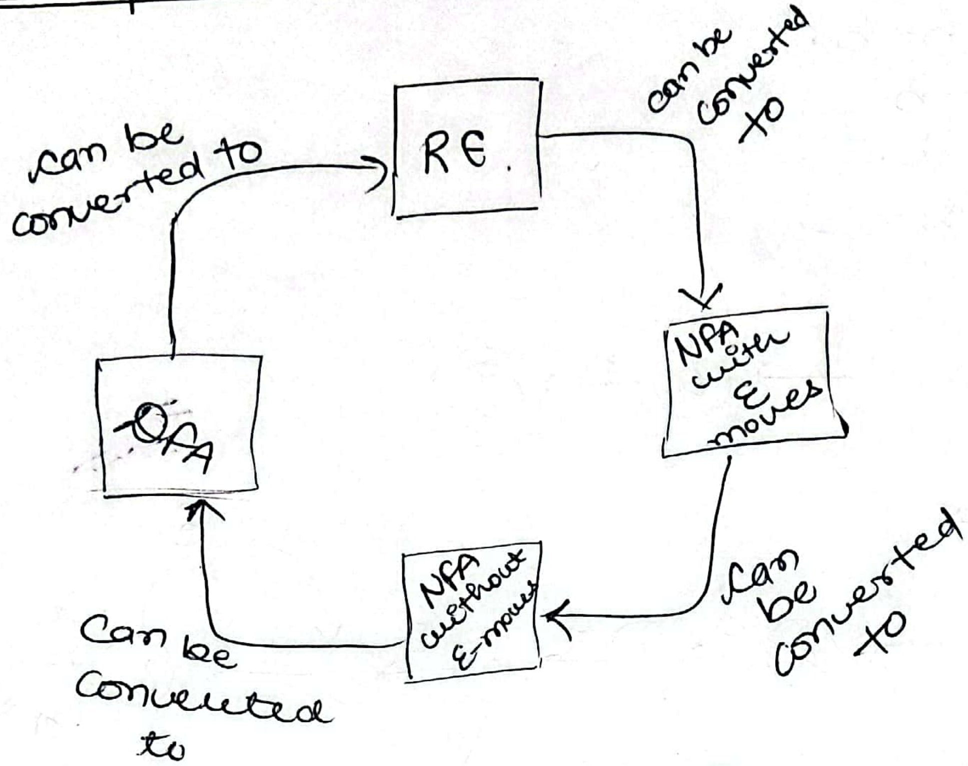
Ques



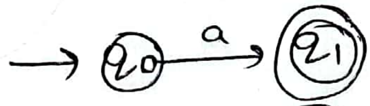
Sol^n $(a + a(b+aa)^*b)^* a$

$(b+aa)^* a$

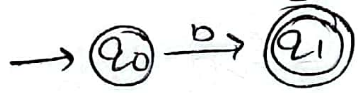
Regular Expression And Finite Automata:



1. R·E = a



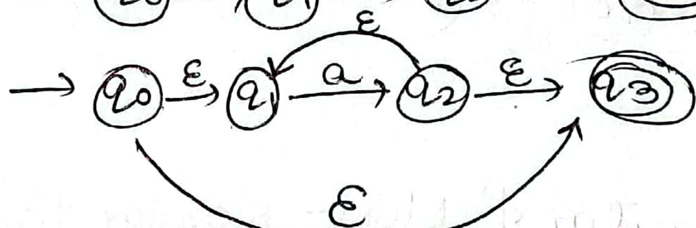
2. R·E = b



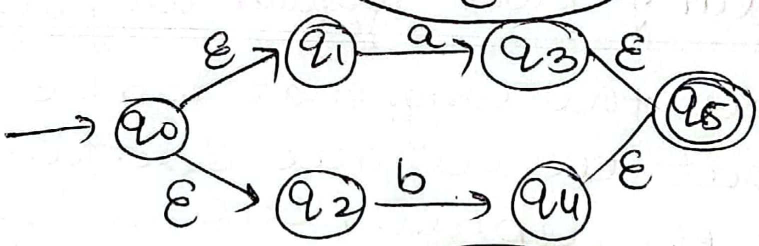
3. RE = ab

4. RE = a^+

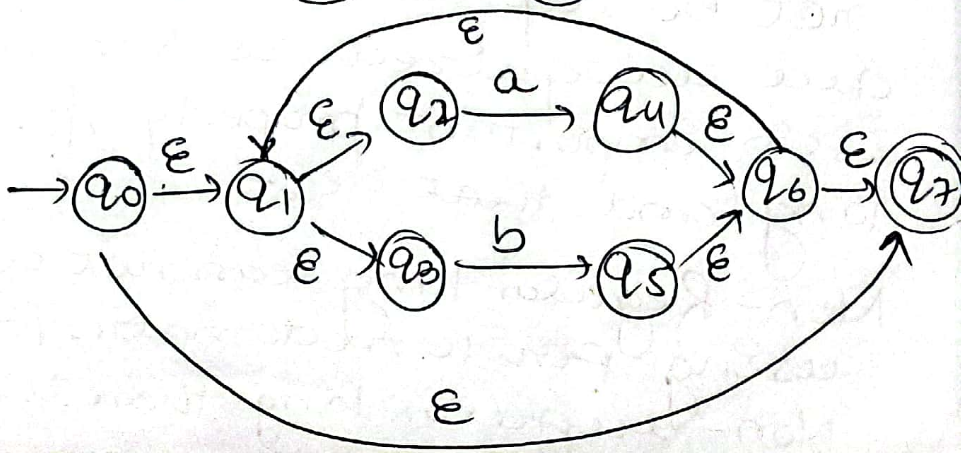
5. RE = a^*



6. R·E = a+b



7. RE = $(a+b)^*$



Q Const. the NFA for the RE = b+ba*

Sol^n

$r = b + ba^*$

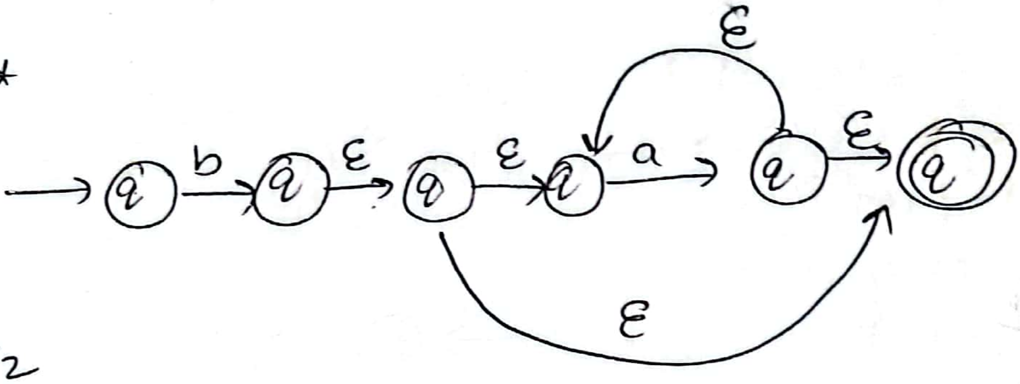
$r_1 = b$

$r_2 = ba^*$

$r = r_1 + r_2$

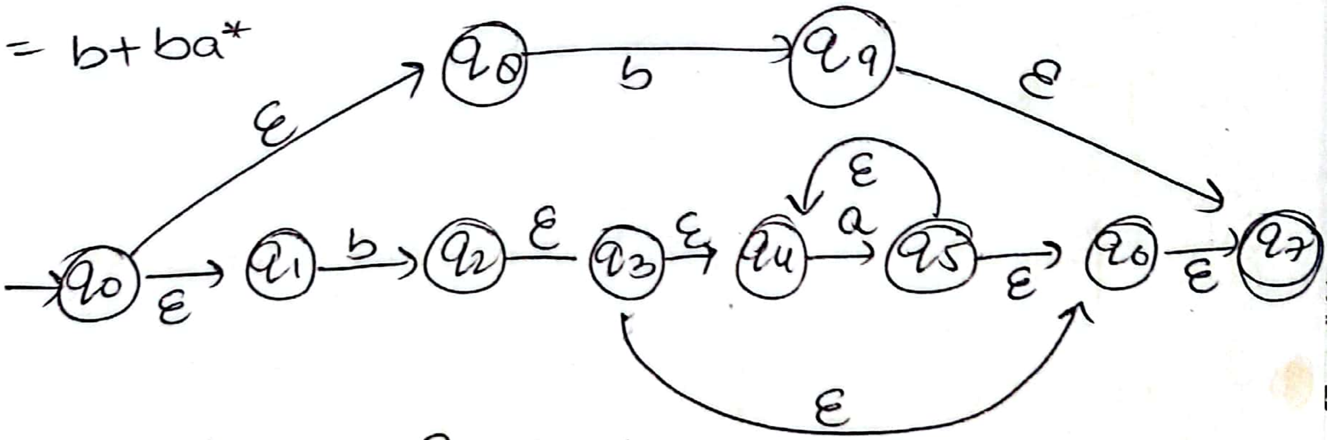
$r_1 = b$

$r_2 = ba^*$



$r = r_1 + r_2$

$= b + ba^*$



• Regular & Non-Regular Languages:

RE are the lang. that can be represented by RE but there are certain langs. which can not be represented using RE. Such lang. are recognized as Non-regular lang. There is an important property for Non-regular lang and ~~that~~ i.e:-

Non-Regular lang can not be represented using finite ~~Automata~~ Automata. For recognizing Non-regular lang there is a theorem or tool

FA pumping lemma is used.

Pumping lemma for Regular languages:

Pumping lemma is a powerful tool for proving certain lang. Non-regular. It is useful in the development of algo. to answer certain question concerning FA such as whether the lang. accepted by a given FA is finite or Infinite.

Statement:

Let 'L' be a lang. then there exist a constant 'n' (which depends on 'L') such that for every string w in 'L'. $|W| \geq n$, We can break W into 3 sub-strings, $w = xyz$, such that:

(i) $y \neq \varepsilon$

(ii) $|xy| \leq n$

(iii) for all $i \geq 0$ the string xy^iz is also in 'L'.

Application of Pumping lemma:

This algo can ~~not~~ be used to prove that certain sets are regular. We now give the steps for proving that a given set is not regular.

(1) Assume 'L' is regular. let 'n' be the no. of states in the corresponding FA.

(2) choose a string w, such that $|W| \geq n$. Use pumping lemma to write $w = xyz$ with $|xy| \leq n$ & $|y| > 0$.

(3) Find a suitable integer 'i' such that $xy^iz \notin L$ this contradiction our assumption. Hence L is not regular.

5

Ques Show that $L = \{a^{i^2} \mid i \geq 1\}$ is not regular.

Soln By using pumping lemma. for each $i \geq 0$ & $y > 0$

$L = \{a^{i^2} \mid i \geq 1\}$

$L = \{a, aaaa, aaaaaaaaa - - \}$.

$w = a$

$w = xy^iz$, for each $i \geq 0$ & $y \neq \epsilon$

let $y = a$

$x = \epsilon$

$z = \epsilon$

$w = \epsilon a^i \epsilon = a^i$

for $i = 0$, $w = a^0 = \epsilon$ which is not in 'L'

for $i = 1$, $w = a^1 = a$ which is in 'L'

for $i = 2$, $w = a^2 = aa$ which is not in 'L'

Therefore $L = \{a^{i^2} \mid i \geq 1\}$ is not regular.

Ques $\{a^n \mid n \geq 0\}$ is regular or not.

Soln $L = \{\epsilon, a, aa, aaa - - \}$.

By using pumping lemma $w = xy^iz$ for all $i \geq 0$ and $y \neq \epsilon$

$w = a$

let $y = a$

$x = \epsilon$

$z = \epsilon$

$w = xy^iz$

$w = \epsilon a^i \epsilon = a^i$

for $i = 0$, $w = a^i = a^0 = \epsilon$, which is in 'L'

for $i = 1$, $w = a^i = a^1 = a$, " " " "

for $i = 2$, $w = a^i = a^2 = aa$, " " " "

Therefore $L = \{a^n \mid n \geq 0\}$ is regular.

Ques Show that $L = \{0^i 1^i \mid i \geq 1\}$ is not regular.

Sol^n $L = \{0^i 1^i \mid i \geq 1\}$

$L = \{01, 0011, 000111 - - \}$

By using pumping lemma.

$W = xy^i z$ for all $i \geq 0$ & $y \neq \epsilon$

Case 1: $y = 1$
$x = 0$ $\because \{y \neq \epsilon\}$
$\epsilon = z$

$W = 0(1)^i \epsilon$
$= 0(1)^i$

When $i = 0$, $W = 0(1)^i = 0 . \epsilon = 0$, which is not in 'L'.
When $i = 1$, $W = 0(1)^i = 0.(1)^1 = 01$, " " " "
When $i = 2$, $W = 0(1)^i = 0.(1)^2 = 011$, " " " "
When $i = 3$, $W = 0(1)^i = 0.(1)^3 = 0111$, " " " "

Therefore, $L = \{0^i 1^i \mid i \geq 0\}$ is not regular.

Case 2: $y = 0$
$x = \epsilon$ $\because \{y \neq \epsilon\}$
$z = 1$

$W = xy^i z = \epsilon(0)^i 1 = 0^i 1$

When $i = 0$, $W = 0^i 1 = 0^0.1 = \epsilon.1 = 1$ which is not in 'L'
When $i = 1$, $W = 0^i 1 = 0^1.1 = 01$ " " " " "
When $i = 2$, $W = 0^i 1 = 0^2.1 = 001$ " " " " "
When $i = 3$, $W = 0^i 1 = 0^3.1 = 0001$ " " " " "

Therefore, $L = \{0^i 1^i \mid i \geq 0\}$ is not regular.

Generalised Transition Graph (GTG):

A GTG is a collection of things:-

- A finite set of states of which atleast one is a start state and some are final states.
- An alphabet Σ of input letters.
- Directed edges connecting some pair of states each labelled with a regular expression.

Closure properties of regular languages:

The closure properties of regular lang. are as given below:-

(1) The Union of two regular lang. is regular.
(2) The Intersection of two regular lang is regular.
(3) The Compliment of regular lang is regular.
(4) The difference of two regular lang is regular.
(5) The reversal of a regular lang. is regular.
(6) The closure operation on a regular lang is regular.
(7) The concatination of two regular lang is regular.

Decision properties of Regular languages:

Following is the set of questions for determining the decision properties of regular lang:-

(1) Is a particular string w belongs to some lang 'L'?
(2) Is the given lang empty?
(3) Do the two descriptions of a lang. represent the same lang? or whether two languages are equivalent.