02 Regular Expression & Regular language

Regular Expression (RE)

- Let Σ be an alphabet which is used to denote the input set.
- The regular expression over Σ can be defined as follows.

1. ϕ is a regular Expression which denotes the empty set.

2. ϵ is a regular expression which denotes set $\{\epsilon\}$ and it is null string.

3. for each a in Σ 'a' is a regular expression and denotes the set $\{a\}$

4. 1. Union → $r+s$ is equivalent to $L_1 \cup L_2$.

   2. concatenation → $r.s$ is equivalent $L_1 L_2$.

   3. closure → r^* is equivalent to L_1^*

   r^* is known as kleen closure or closure, which indicates occurrence of r for ∞ number of times.

   r^+ is known as positive closure

Eg if $\Sigma = \{a\}$, $RE = a^*$ then $R = \{\epsilon, a, aa, aaa \cdots\}$
$RE = a^+$ then $R = \{a, aa, aaa \cdots\}$

$$a^* = \epsilon . a^+$$

1. RE for the language accepting all combination of a's

2. Write RE for the language accepting all combinations of a's except the null string.

3. Write RE for the language containing all the strings containing any number of a's and b's

4. Write RE for the language containing all string having any number of a's and b's except the null string.

5. construct the RE for the language accepting all the string which are ending with 00 over the set $\Sigma = (0,1)$

6. Write RE for the language accepting the strings which are starting with 1 and ending with 0

7. write RE for the language stating and ending with a and having any combination of b's in between them

8. Write RE to denote the language L over $\Sigma = \{a,b,c\}$ in which every string will be such that any number of a's is followed by any number of b's followed by any number of c's

9. Write RE to denote a language L which is accepts all the string which begin or end with either 00 or 11.

RE = $[(00+11)(0+1)^*] + [(0+1)^*(00+11)]$

10 construct RE for the language L which accepts all the strings with atleast two b's over the $\Sigma = \{a, b\}$

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

11 Exactly two b's

RE = $a^* b a^* b a^*$

12 Even length of string $\Sigma = \{0\}$

RE = $(00)^*$

13 odd length of string $\Sigma = \{1\}$

RE = $1(11)^*$

14 Write RE for the language L over the $\Sigma = \{a, b\}$ in which total number of a's divisible by 3

RE = $(b^* a b^* a b^* a b^*)^*$

15. Write RE for the language L such that all the string do not contain the substring "ab"

RE = $\{b^* a^*\}$

16 Write RE for the following language.
i. $\Sigma = \{0,1\}$ containing all possible combination of 0's & 1's but not having consecutive 0's

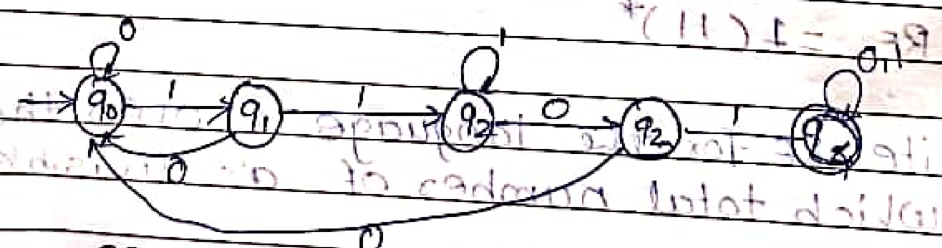
ii The set of all string of 0's & 1's such that every pair of adjacent 0's appears before any pair of adjacent 1's

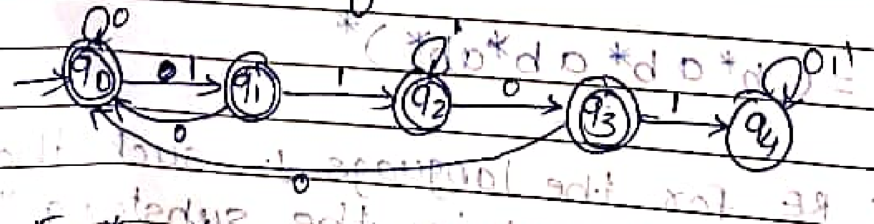
iii The set of all string over $\Sigma = \{0,1\}$ without length of two

17 obtain RE such that $L(R) = \{w \mid w \in \{0,1\}\}$ with at least three consecutive 0's

18 write RE for the following language.
i. The set of all the strings such that the number of 0's is odd

ii The set of all the strings that do not contain 1101





$\boxed{0^*(1\,0)^* 1\,1\,0\,0^*}$

1. Solve & Design DFA for ~~positive~~ contains

2. do not contains means complement of m/c

3) complement - we will change final state to non final state and non final state to final state

19. find RE over $\Sigma = \{0,1\}$

1. containing atleast 2 0's

2. begin or end with 00 or 11

3. Both the numbers of 0's and 1's are even

20 Describe the language denoted by following regular expression

$RE = (b^* (aaa)^* b^*)^*$

21 Describe in simple English the language represented by the following RE.

$L(R) = RE = (a+ab)^*$

Ardens Theorem

The Ardens theorem is useful for checking the equivalence of two regular expression as well as in conversion of DFA to RE.

Let P & Q be the two regular expression over the input set Σ.

The Regular Expression R is given as

$$\boxed{R = Q + RP}$$

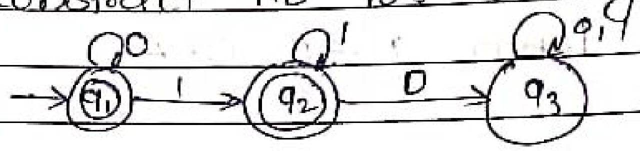
Which as unique solution as P

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

Equivalence of RE and DFA

1. Let q_1 be the initial state.
2. There are $q_2, q_3, q_4 \ldots q_n$ number of states. The final state may be some q_j where $j<=n$
3. Let α_{ji} represents the transition from q_j to q_i
4. Calculate q_i such that
   $q_i = \alpha_{ji} \cdot q_j$
   if q_i is start state $q_i = \alpha_{ji} \cdot q_j + \epsilon$
5. similarly compute the final state which ultimatly gives the RE.

Q. 1. Construct RE for the given DFA



→ let us build the Regular Expression for each state

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

Since final states are q_1 & q_2 we are interested in solving q_1 & q_2 only

$q_1 = \epsilon + q_1 0$ $\quad\quad$ $R = Q + RP$

$q_1 = \epsilon . (0)^*$ $\quad\quad$ $\Rightarrow QP^*$

$\boxed{q_1 = 0^*}$ — (1)

substituting this into q_2

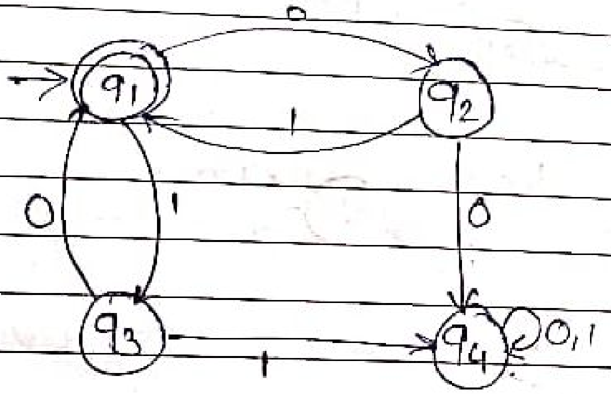
$q_2 = 0^* 1 + q_2 1$ $\quad\quad$ $R = Q + RP$

$q_2 = 0^* 1 . 1^*$ $\quad\quad$ $R = QP^*$

$RE = q_1 + q_2 = 0^* + 0^* 1 . 1^*$

$\boxed{RE = 0^* + 0^* 1^+}$ $\quad\quad$ $1 . 1^* = 1^+$

Q.3. construct RE. for the DFA given in fig.
2.b)



Let us see the equation

$q_1 = q_2 1 + q_3 0 + \epsilon$
$q_2 = q_1 0$
$q_3 = q_1 1$
$q_4 = q_2 0 + q_3 1 + q_4 (0+1)$

Let us solve ~~we~~ q_1 first

~~q_1 = [illegible]~~

$q_1 = q_1 01 + q_1 10 + \epsilon$
$q_1 = q_1 (01 + 10) + \epsilon$ $R = Q + RP$

$q_1 = \epsilon + q_1 (01 + 10)$ $\Rightarrow QP^*$

$q_1 = \epsilon (01 + 10)^*$

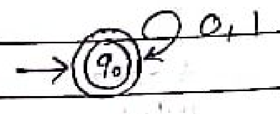
$q_1 = (01 + 10)^*$

Thus the Regular Expression (RE) is

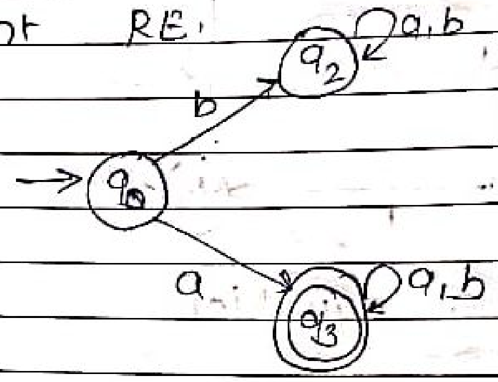
$RE = (01 + 10)^*$

since q_1 is a final state, we are in trested in q_1 only

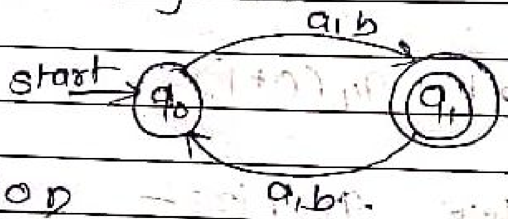
Q. 3. Represent RE



Q. 4. Represent RE.



Q. 5. Describe in english language indicated by following RE.



→ Equation

$$q_0 = q_1(a+b) + \epsilon$$
$$q_1 = q_0(a+b)$$

We solve q_1

$$q_1 = q_0(a+b)$$
$$q_1 = [\epsilon + q_1(a+b)](a+b)$$
$$q_1 = (a+b) + q_1(a+b)(a+b) \qquad R = Q + RP$$
$$q_1 = (a+b).((a+b).(a+b))^* \qquad \Rightarrow = QP^*$$

$$RE = (a+b)\boxed{[(a+b).(a+b)]}^*$$

The language given by this DFA is the language in which it accepts all the strings of odd length.

Decision properties of regular language.

A decision property is a (Boolean) question about a language.

→ Is the language empty?

→ is the language a subset of another language?

Decision properties

1) Membership

2) Empty set & Empty string

3) Equivalence & subset

4) language size.

Closure Properties of Regular language

Closure properties on regular language are defined as certain operations on regular language which are guaranteed to produce regular language.

Regular language are closed under following operation.

Consider L & M are regular language.

1. The union of two RL is Regular ($L \cup M$)
2. The intersection of two RL is Regular ($L \cap M$)
3. The complement of a RL is Regular ($\bar{L}$) or ($\bar{M}$)
4. The difference of two RL is Regular ($L - M$)
5. The Reversal of a RL is Regular (L^R)(M^R)
6. The closure operation on a RL is Regular (L^*)
7. The concatension of RL is regular ($L \cdot M$)
8. A homomorphism of RL is regular
9. The inverse homomorphism is of RL is Regular

Pumping Lemma.

This is a basic & important theorem used for checking whether given string is accepted by regular expression or not.

This lemma tells us whether given language is Regular or not.

Q. 1 $L = \{0^{2n} \mid n \geq 1\}$

This is a language length of string is always even.

$n = 1$; $L = 00$

$n = 2$; $L = 0000$ and so on.

Let $L = uvw$

$L = 0^{2n}$

$|x| = 0^{2n} = uv^iw$

case I

Q.2 $L = \{a^n b^n \mid n >= 1\}$

we assume L is regular language.

we will consider $w \in L$ if

$w = xyz$

case I $w = aabb$

mapped with xyz

$$\underset{x}{a}\ \underbrace{ab}_{y}\ \underset{z}{b}$$

$w = xy^iz \in L$

now $i = 2$

$w = a\ ab\ ab\ b$

$= a^2 b a b^2 \notin L$

case 2 $w = aabb$

mapped with xyz

$$w = \underset{x}{a}\ \underset{y}{a}\ \underbrace{bb}_{z}$$

$w = xy^iz \in L$

now $i = 2$

then $w = aaabb \notin L$

Not

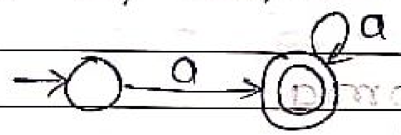
Q L? = Regular

Finite language is Regular

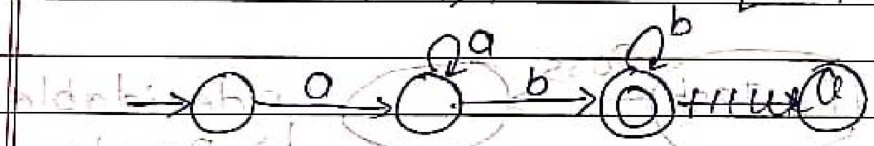
Infinite language → If you have pattern/loop then you design Finite Automata then you called as Regular

→ pumping lemma.

1. $a^n \mid n \geq 1$ $L = \{a, aa, aaa \ldots\}$



2. $a^n b^m$ $n, m \geq 1$ $L = \{ab, aabbb, abb \ldots\}$



3. $a^n b^n \mid n \leq 10^{10^{10}}$

4. $a^n b^n \mid n \geq 1$ ← Not Regular & we go for pumping lemma.

5. $\{ww^R \mid |w| = 2$ $\Sigma = \{a, b\}$

aaaa

abba

baab

bbbb

Regular Because FA have finite state.

ww^R / $w \in (a,b)^*$

~~abbabb~~
abbbba

$a^n b^m c^k$ / $n,m,k \geq 1$ Regular.

$a^n b^n c^n$ / $n \geq 1$ Not Regular

$a^i b^{2j}$ / $i,j \geq 1$

what is pumping lemma
It is Negative Test

L $\rightarrow$ PL Test $\xrightarrow{\text{Pass}}$ undecidable
$\rightarrow$ Regular
$\rightarrow$ Not Regular.

$\downarrow$ fail

Not Regular $\leftarrow$ Decidable

pumping lemma.

If L is an infinite language then there exist some positive integer 'n' (pumping length) such that any string $w \in L$ has length greater than equal to 'n' i.e. $|w| \geq n$ then w can be devided into three parts, $w = xyz$ satisfy following condition.

i for each $i \geq 0$, $xy^iz \in L$

ii $|y| > 0$

iii $|xy| \leq n$