UNIT - 2

Regular Expressions:

* An alternative representation of finite automata is regular expression (text representation form).
* Regular Expressions are applied in
  (i) Token separation (complier)
  (ii) Pattern matching (searching)

Definitions:

* Regular Expressions can be considered as a programming languages that can be used to perform simple applications such as token separation and pattern matching.
* Regular expressions are closely related with NFA and can be considered as an alternative to NFA notation for describing an software components.
* Regular expression is defined exactly by the same language that is obtained by the automata model.
* It serves as an fundamental inputs for many real time systems
  (i) Grep command in Linux
  (ii) Lexical analyzer (first phase of complier)

operators of Regular Expression:

operators of regular expressions are

* - closure
. - dot (or) and
+ - cat (or) OR

(i) Union of two languages

If L and M are two languages with L = {001, 10, 111} and M = {ε, 001} then

L∪M or L+M or L/M = {ε, 001, 10, 111}

(ii) concatenation of languages

If L = {001, 10, 111} and M = {ε, 001} then

LM or L.M = {001ε, 10ε, 111ε, 001001, 10001, 111001}

(concatenate M with L)

(iii) closure

If L = {0, 11} then L^* = {ε, 0, 11, 000, 1111, 011, 001111, 011000}

Example: If L = {0, 11} find L^2 and L^3

$L^2 = L.L = \{0, 11\}.\{0, 11\}$

$= \{00, 011, 110, 1111\}$

$L^3 = L^2.L = \{00, 011, 110, 1111\}.\{0, 11\}$

$= \{000, 0011, 0110, 01111, 1100, 11011, 11110, 111111\}$

Building of Regular Expression Basis:

Basis:

(i) For constant ε (→ null string) & ϕ (→ Empty), the languages are denoted as

L(ε) = {ε} and L(ϕ) = ϕ

(ii) If 'a' is any input symbol then the regular expression for 'a' is denoted as L(a) = {a}

Induction:

(i) If E and F are regular expressions then E+F is a regular expression with $L(E+F) = L(E) + L(F)$

(ii) If E and F are RE then E.F is a RE with $L(E.F) = L(E).L(F)$

(iii) If E is an RE then E^* is also a RE with $L(E^*) = (L(E))^*$

Regular languages to Regular Expression:

Problems

1) set of all strings with one or more zeroes followed by 1.

$L = \{01, 001, 0001, 00001, \ldots\}$

$RE = 0^+ . 1$ → one or more times

2) set of all strings ending with 11 over $\{0,1\}$

$L = \{11, 011, 111, 0011, 1111, \ldots\}$

$RE = (0+1)^* . 11$ → zero or more times

3) set of all strings that starts with 'a' and ends with 'b' over $\{a,b\}$

$L = \{ab, aab, abb, abab, \ldots\}$

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

4) set of all strings that contain '101' over $\{0,1\}$

$L = \{101, 1101, 0101, 10101, \ldots\}$

$RE = (0+1)^* 101 (0+1)^*$

5) set of all strings that contain exactly 2 0's over $\{0,1\}$

$L = \{00, 100, 1100, 0011, \ldots\}$

$RE = 1^* 01^* 01^*$

c) set of all strings not ending with '00' over $\{0,1\}$

$RE = (0+1)^* (01+10+11+0+1)$

Regular Expression to Regular Language.

i) convert the following RE to RL

1) $b(a+b)^*$

$L = \{b, ba, bb, bab, baa, bbbb, \ldots\}$

$L = \{$set of all strings that starts with b over $\{a,b\}\}$

2) $0(0+1)^* (10)$

$L = \{010, 0010, 0110, 00110, \ldots\}$

$L =$ set of all strings that starts with '0' and ends with '10' over $\{0,1\}$

3) $(a+b)^* abb(a+b)^*$

$L = \{abb, aabb, baabb, ababb, \ldots\}$

$L =$ set of strings with substring 'abb' as a substring over $\{a,b\}$

4) $1^*0(0+1)^* 1(0+1)^*$

$L = \{01, 101, 001, \ldots\}$

$L =$ set of all strings with atleast one '0' and one '1' over $\{0,1\}$

5) $(1+\epsilon)(00^*1)0^*$

$L = \{101, 01, 001, 01000, 100010, \ldots\}$

$L =$ set of all strings that contains '01' with atmost 2 1's over $\{0,1\}$

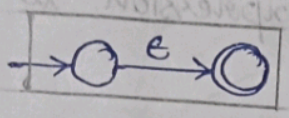
4/9/24

Regular Expression to ϵ-NFA

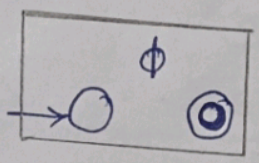
Thomson's construction method

Basis:

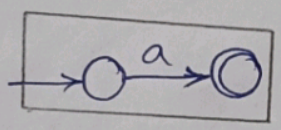
(i) If ϵ is a regular expression, ϵ-NFA is drawn as



(ii) ϕ is an ~~ϵ-NFA~~ regular expression, ϵ-NFA is drawn as



(iii) If 'a' is a regular expression, ϵ-NFA is drawn as

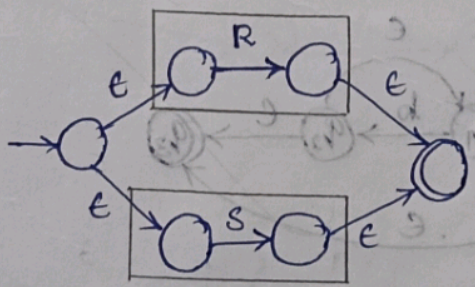


Induction:

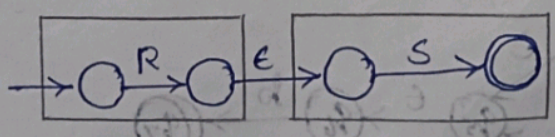
If R+S is a regular expression then ϵ-NFA for R+S can be drawn as

(i) R+S → ϵ-NFA

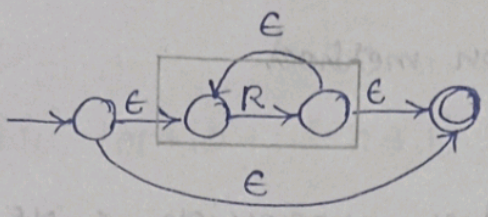
(or) R|S



If R.S (RS) is a regular expression then ϵ-NFA is drawn as

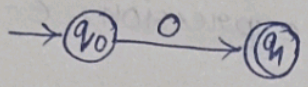


If R^* is a regular expression then ϵ-NFA is drawn as

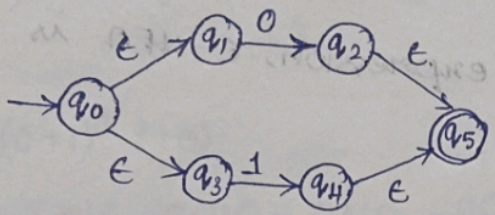


convert the following regular expression to ϵ-NFA

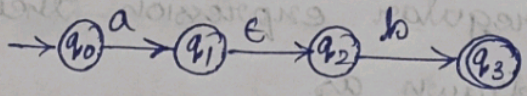
(i) 0



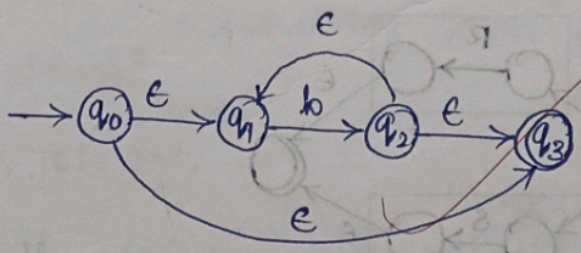
(ii) 0+1



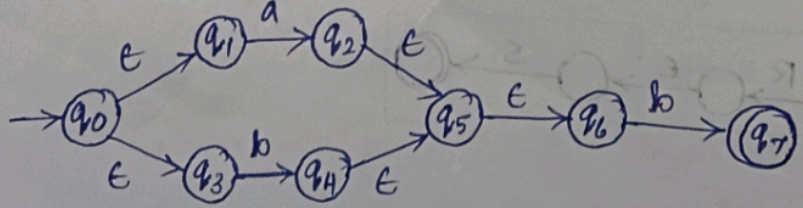
(iii) a.b



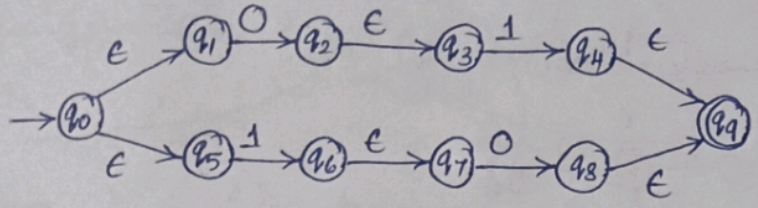
(iv) b^*



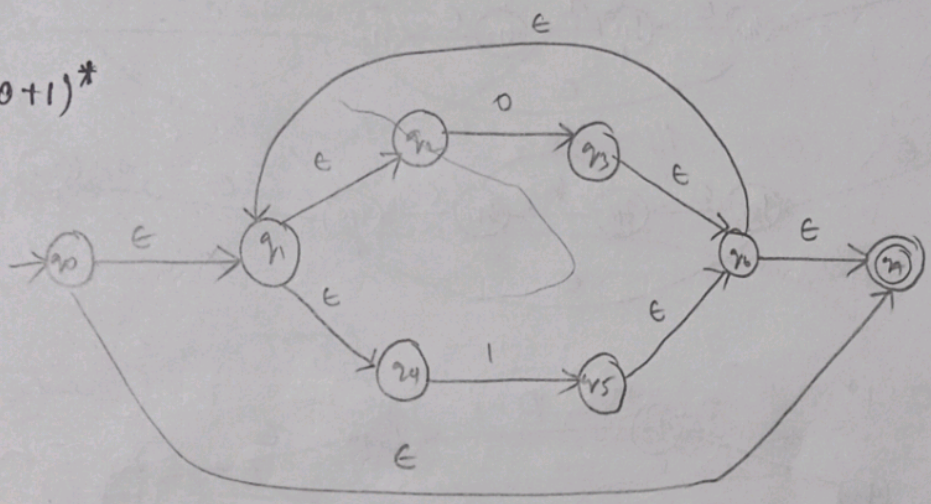
(v) $(a+b).b$



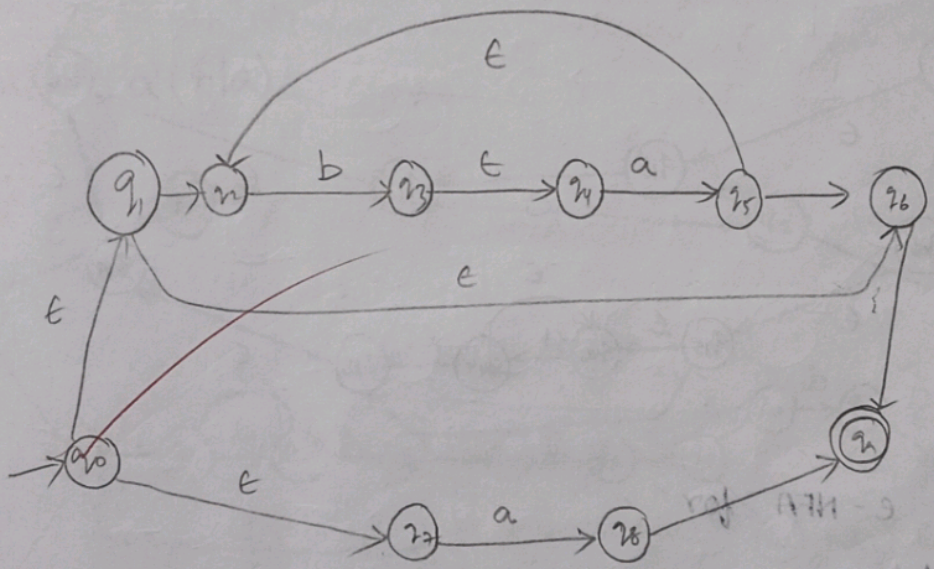
(vi) 0.1/1.0



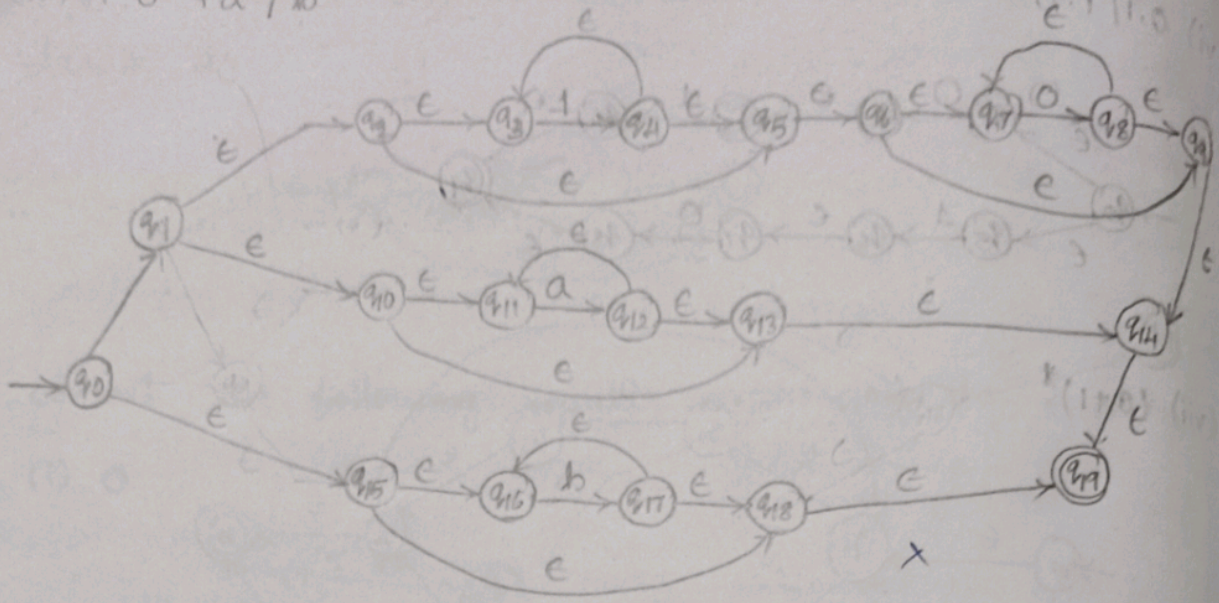
(vii) (0+1)*



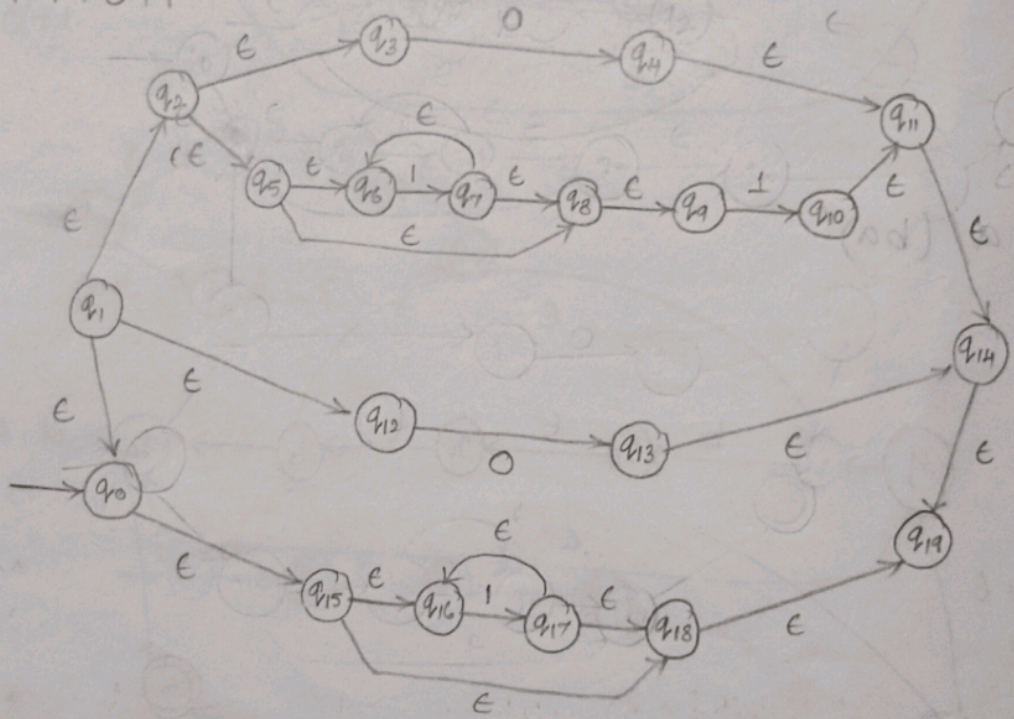
viii) a/(ba)*



(ix) $1^*0^* + a^*/b^*$



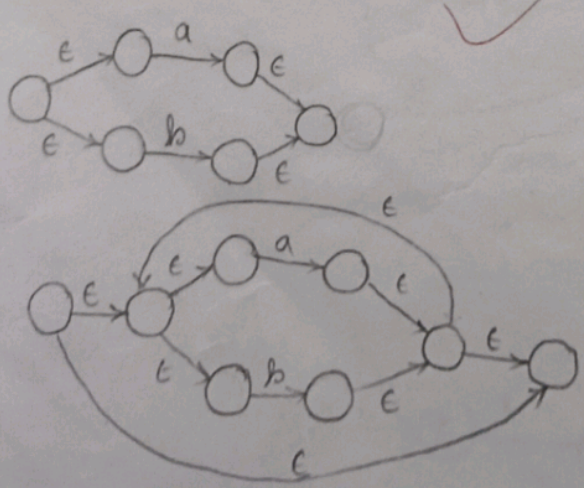
x) $0/1^*1+0+1^*$

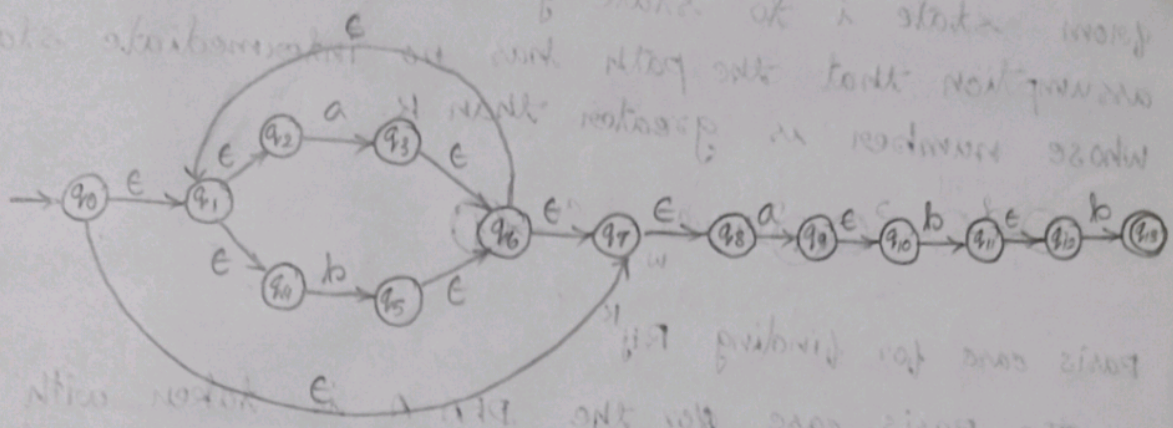


9/9/24

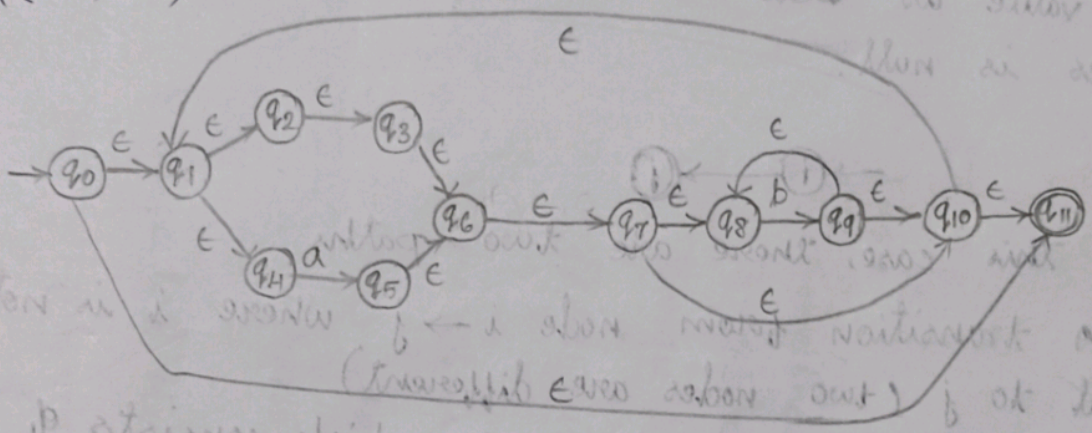
Draw the ε-NFA for

(i) $(a/b)^*abb$

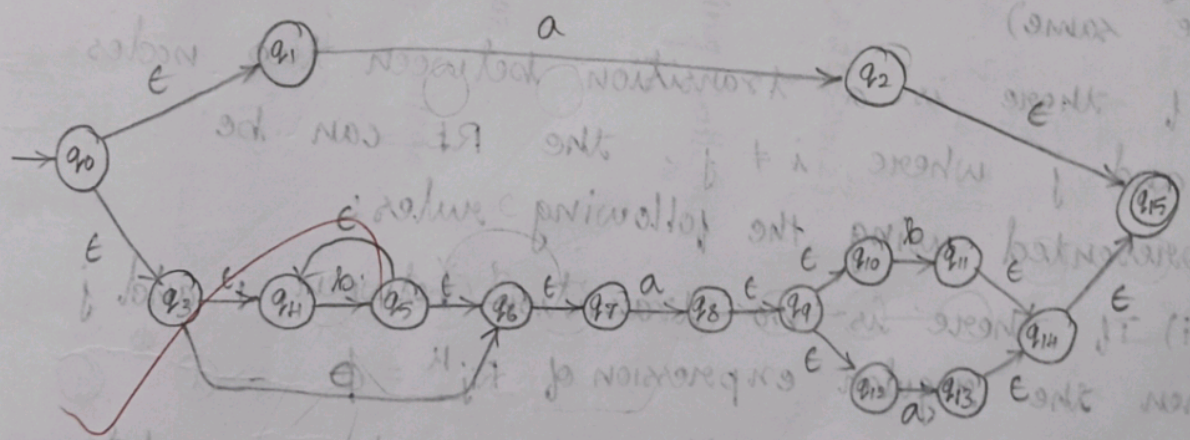




(ii) $((\epsilon/a)b^*)^*$



(iii) $a/b^*\,a(b/a)$

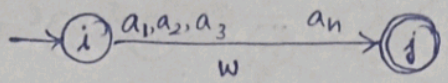


10/9/24

DFA to Regular Expression:

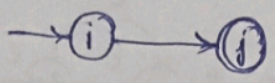
* If $L = L(A)$ for some DFA A, then there exist a regular expression R such that $L(R) = L(A)$ Where $L(R)$ denotes the set of all strings derived from the regular expression R.
* Let R_{ij}^{k} be a regular expression whose language contains set of strings 'w' where 'w' is a label of path

from state i to state j in DFA A, with assumption that the path has no intermediate states whose number is greater than k.



Basis case for finding R_{ij}^{k}

The Basis case for the DFA A is taken with 'k' value as zero which is the no. of intermediate nodes is null.



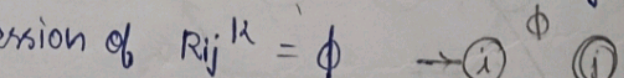
For this case, there are two paths

(i) A transition from node $i \rightarrow j$ where i is not equal to j (two nodes are different)

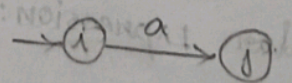
(ii) A transition of length zero which consists of only one node where i is equal to j (two nodes are same)

If there is a transition between two nodes i and j where $i \neq j$, the RE can be represented using the following rules:

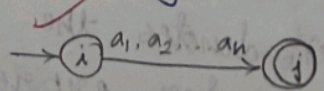
(i) If there is no transition between i and j then the regular expression of $R_{ij}^{k} = \phi$



(ii) If there is exactly one symbol from state $i \rightarrow j$ then $R_{ij}^{k} = a$

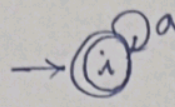


(iii) If there are more than one symbols on the transition state $i \rightarrow j$, then regular expression $R_{ij}^{k} = a_1 + a_2 + \ldots + a_n$



Note:

If the DFA is of the following form



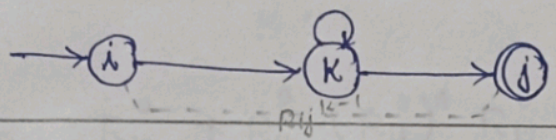
then $R_{ij}^{k} = \epsilon + a$

Assumption:

Assume that the condition is true for $k-1$ intermediate nodes and prove it for k.

Induction:

R_{ij}^{k} can be recursively defined as



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

When the input string is given to the above DFA model

case-1:

It moves from state $i \rightarrow j$ without passing through any intermediate states. (R_{ij}^{k-1})

case-2:

It moves through some intermediate state k from i to j, it can be represented as

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

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

Note:

i) If DFA has only one accepting state then the regular expression is represented as R_{ij}^{k} where

i - initial state, k - no. of states of DFA

j - final state

2) If there are more than one final state in a DFA, then RE can be represented as:

$$Rip^k + Riq^k + \dots + \dots$$

where i - initial state

p, q - final states of DFA

k - no. of states of DFA

11/9/24

Transition Rules:

1) $\phi + r = r$

2) $\epsilon . r = r . \epsilon = r$

3) $\phi . r = r . \phi = \phi$

4) $\epsilon^* = \epsilon . \phi^* = \epsilon$

5) $r + r = r$

6) $r^* + r^* = r^* = (r^*)^*$

7) $r . r^* = r^* . r$

8) $\epsilon + r . r^* = r^*$

9) $(pq)^* . p = p . (pq)^*$

10) $(p+q) r = pr + qr$

11) $r(p+q) = rp + rq$

12) $r^* + \epsilon = r^*$

13) $r^* (a+b) = r^* (a+b) + (a+b)$

14) $(r + \epsilon)^* = r^*$

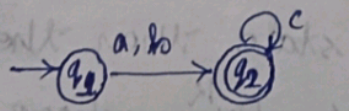
15) $r^* r + r = r^* r$

16) $(r + \epsilon) r^* = r^*$

17) $\phi + \epsilon = \epsilon$, 18) $(\phi + \epsilon)^* = \epsilon^* = \epsilon$

problems:

1) Find the regular expression for the given DFA.



$(a+b)c^*$

one final state

R_{ij}^{k} where $i = 1$ (initial state no)
$j = 2$ (final state no)
$k = 2$ (no. of states of DFA)

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

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

R_{12}^{1} where $i = 1$
$j = 2$
$k = 1$

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

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

$R_{11}^{0} = \phi + \epsilon$ (include ϵ bcz i and j are same)
$R_{12}^{0} = (a+b)$
$R_{21}^{0} = \phi$
$R_{22}^{0} = c + \epsilon$

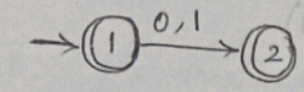
$$R_{12}^{1} = (a+b) + (\phi+\epsilon).(\phi+\epsilon)^* (a+b)$$
$$= (a+b) + \epsilon . \epsilon^* (a+b) \quad (\epsilon^* = \epsilon)$$
$$= (a+b) + (a+b) \quad \text{using the rule } r.\epsilon = r$$
$$= (a+b) \quad r + r = r$$

$$R_{22}^{1} = (c+\epsilon) + \phi . (\phi+\epsilon)^* . (a+b)$$
$$= (c+\epsilon) + \phi \quad \phi . r = \phi$$
$$= (c+\epsilon) \quad r + \phi = r$$

$$R_{12}^{2} = (a+b) + (a+b).(c+\epsilon)^* (c+\epsilon)$$
~~$= (a+b) + (a+b).c^*.(c+\epsilon)$~~
$$= (a+b)(\epsilon + (c+\epsilon)^* (c+\epsilon)) \quad \epsilon + r.r^* = r^*$$

$= (a+b)(c+\epsilon)^*$

$R_{12}^2 = (a+b)c^*$

2) 

It has two final state.

$R_{ip}^k + R_{iq}^k$ where $i = 1$ (initial state no)

$p, q = 1, 2$ (final state no)

$k = 2$ (no of states of DFA)

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

$RE = R_{11}^2 + R_{12}^2$

$R_{11}^2 = R_{11}^1 + R_{12}^1 (R_{22}^1)^* R_{21}^1$

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

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

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

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

$R_{11}^0 = \phi + \epsilon$

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

$R_{21}^0 = \phi$

$R_{22}^0 = \phi + \epsilon$

$R_{21}^1 = \phi + \phi \cdot (\phi+\epsilon)^* \cdot (\phi+\epsilon)$ $\quad \phi \cdot r = \phi$

$= \phi + \phi \cdot \epsilon = \phi + \phi = \phi$

$R_{22}^1 = (\phi+\epsilon) + \phi \cdot \underbrace{(\phi+\epsilon)^*}_{r} \cdot (0+1)$

$= (\phi+\epsilon) + \phi$ $\quad \phi \cdot r = \phi$

$= \epsilon + \phi = \epsilon$ $\quad \phi + \epsilon = \epsilon$

$R_{12}^1 = (0+1) + (\phi+\epsilon) \cdot \underbrace{(\phi+\epsilon)^*}_{r} \cdot (0+1)$

$= (0+1) + \underbrace{(\phi+\epsilon)^*}_{\epsilon} \cdot (0+1)$ $\quad \epsilon \cdot r = r$

$= (0+1) + (0+1) = (0+1)$ $\quad (\phi+\epsilon) = \epsilon = \epsilon$

$r + r = r$

$R_{11}^1 = (\phi+\epsilon) + (0+\epsilon) \cdot (\phi+\epsilon)^* \cdot (\phi+\epsilon)$

$= \epsilon + \epsilon \cdot \epsilon \cdot \epsilon$

$= \epsilon + \epsilon = \epsilon$

$R_{11}^2 = \epsilon + (0+1) \cdot \underbrace{\epsilon^* \cdot \phi}_{\phi}$ $\quad \epsilon^* = \epsilon \cdot \phi^* = \epsilon$

$= \epsilon + \phi$

$= \epsilon$

R_{12}^{2}

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

$$= (0+1) + (0+1) \cdot \epsilon \cdot \epsilon$$

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

$$= 0+1$$

$r \cdot \epsilon = r$

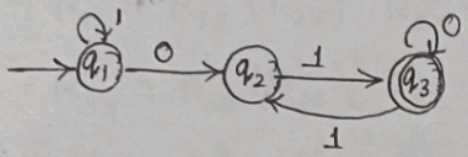
$r + r = r$

$$RE = R_{11}^{2} + R_{12}^{2}$$

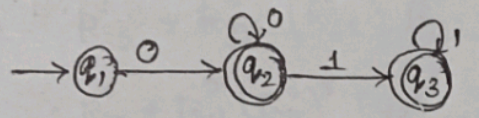
$$\boxed{RE = \epsilon + 0 + 1}$$

18/9/24

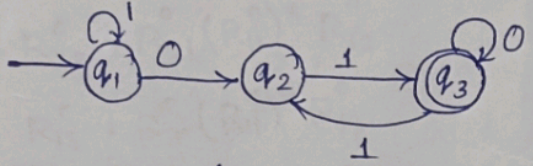
Find the RE for

(i) 

$1^{*}01(0+11)^{*}$

(ii) 

$00^{*}11^{*}$

(i) 

one final state

R_{ij}^{k} where $i = 1$

$j = 3$

$k = 3$

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

$$R_{13}^{2} = R_{13}^{1} + R_{12}^{1} (R_{22}^{1})^{*} R_{23}^{1}$$

$$R_{33}^{2} = R_{33}^{1} + R_{32}^{1} (R_{22}^{1})^{*} R_{23}^{1}$$

$$R_{13}^{1} = R_{13}^{0} + R_{11}^{0} (R_{11}^{0})^{*} R_{13}^{0}$$

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

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

$$R_{23}^{1} = R_{23}^{0} + R_{21}^{0} (R_{11}^{0})^{*} R_{13}^{0}$$

$$R_{33}^{1} = R_{33}^{0} + R_{31}^{0} (R_{11}^{0})^{*} R_{13}^{0}$$

$$R_{32}^{1} = R_{32}^{0} + R_{31}^{0} (R_{11}^{0})^{*} R_{12}^{0}$$

$R_{11}^0 = \epsilon + 1$

$R_{12}^0 = 0$

$R_{13}^0 = \phi$

$R_{21}^0 = \phi$

$R_{22}^0 = \epsilon + \phi$

$R_{23}^0 = 1$

$R_{31}^0 = \phi$

$R_{32}^0 = 1$

$R_{33}^0 = \epsilon + 0$

$R_{32}^1 = R_{32}^0 + R_{31}^0 (R_{11}^0)^* R_{12}^0$

$= 1 + \phi . (\epsilon + 1)^* . 0 \quad 1 + \phi . 1^* 0$

$= 1 + \phi \cdot 1$

$R_{33}^1 = (\epsilon + 0) + \phi . (\epsilon + 1)^* . \phi$

$= (\epsilon + 0) + \phi = \epsilon + 0$

$R_{23}^1 = 1 + \phi . (\epsilon + 1)^* . \phi$

$= 1 + \phi = 1$

$R_{22}^1 = (\epsilon + \phi) + \phi . (\epsilon + 1)^* . 0$

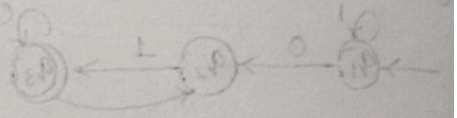
$= \epsilon + \phi = \epsilon$

$R_{12}^1 = 0 + (\epsilon + 1) . (\epsilon + 1)^* . 0 = 0(\epsilon + 1 . 1^*)$

$= 0 . 1^* = 1^* . 0$

$R_{13}^1 = \phi + (\epsilon + 1) . (\epsilon + 1)^* . \phi$

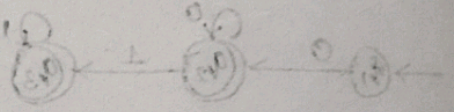
$= \phi + \phi = \phi$



$R_{33}^2 = (\epsilon + 0) + 1 . \epsilon^* . 1$

$= (\epsilon + 0) + 1 . 1$

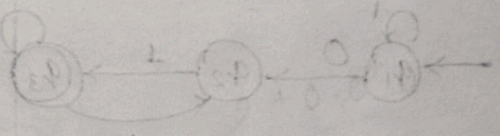
$= \epsilon + 0 + 1 . 1$



$R_{13}^2 = \phi + 1^* . 0 . \epsilon^* . 1$

$= \phi + 1^* . 0 . 1$

$= 1^* . 0 . 1$

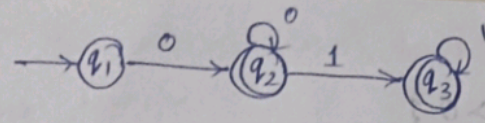


$R_{13}^3 = 1^* . 0 . 1 + 1^* . 0 . 1 . (\epsilon + 0 + 1 . 1)^* . (\epsilon + 0 + 1 . 1)$

$= 1^* . 0 . 1 \, (\epsilon + (\epsilon + 0 + 1 . 1)^* . (\epsilon + 0 + 1 . 1))$

$= 1^* . 0 . 1 \, . (\epsilon + 0 + 1 . 1)^*$

$= 1^* . 0 . 1 \, (0 + 1 . 1)^*$

(ii) 

doubt

It has two final states

$R^{k}_{ip} + R^{k}_{iq}$ where $i = 1$

$p, q = 2, 3$

$k = 3$

$$RE = R^{3}_{12} + R^{3}_{13}$$

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

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

$$R^{2}_{13} = R^{1}_{13} + R^{1}_{12}(R^{1}_{22})^{*} R^{1}_{23}$$

$$R^{2}_{33} = R^{1}_{33} + R^{1}_{32}(R^{1}_{22})^{*} R^{1}_{23}$$

$$R^{2}_{32} = R^{1}_{32} + R^{1}_{32}(R^{1}_{22})^{*} R^{1}_{22}$$

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

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

$$R^{1}_{13} = R^{0}_{13} + R^{0}_{11}(R^{0}_{11})^{*} R^{0}_{13}$$

$$R^{1}_{23} = R^{0}_{23} + R^{0}_{21}(R^{0}_{11})^{*} R^{0}_{13}$$

$$R^{1}_{33} = R^{0}_{33} + R^{0}_{31}(R^{0}_{11})^{*} R^{0}_{13}$$

$$R^{1}_{32} = R^{0}_{32} + R^{0}_{31}(R^{0}_{11})^{*} R^{0}_{12}$$

$R^{0}_{11} = \epsilon + \phi$

$R^{0}_{12} = 0$

$R^{0}_{13} = \phi$

$R^{0}_{21} = \phi$

$R^{0}_{22} = \epsilon + 0$

$R^{0}_{23} = 1$

$R^{0}_{31} = \phi$

$R^{0}_{32} = \phi$

$R^{0}_{33} = \epsilon + 1$

$R^{1}_{32} = \phi + \phi.(\epsilon+\phi)^{*}.0$

$= \phi + \phi.0$ (ϵ) $\quad \epsilon.r = r$, $\phi.r = \phi$

$= \phi + \phi = \phi$

$R^{1}_{33} = (\epsilon+1) + \phi.(\epsilon+\phi)^{*}.\phi$

$= (\epsilon+1) + \phi$ $\quad \epsilon.\phi = \phi$, $r+\phi = r$

$= (\epsilon+1)$

$R^{1}_{23} = 1 + \phi.(\epsilon+\phi)^{*}.\phi$

$= 1 + \phi$ $\quad \epsilon.\phi = \phi$, $r+\phi = r$

$= 1$

$R^{1}_{13} = \phi + (\epsilon+\phi).(\epsilon+\phi)^{*}.\phi$

$= \phi + \phi = \phi$

$R^{1}_{22} = (\epsilon+0) + \phi.(\epsilon+\phi)^{*}.0$

$= (\epsilon+0) + \phi.0 = (\epsilon+0) + \phi$

$= (\epsilon+0)$

$R^{1}_{12} = 0 + (\epsilon+\phi).(\epsilon+\phi)^{*}.0$

$= 0 + 0 = 0$ $\quad r+r = r$

$R^{2}_{32} = \phi + \phi.(\epsilon+0)^{*}.(\epsilon+0)$ (r)

$= \phi + \phi = \phi$ $\quad \phi.r = \phi$

$R^{2}_{33} = (\epsilon+1) + \phi.(\epsilon+0)^{*}.1$ (r)

$= (\epsilon+1) + \phi = (\epsilon+1)$

$R^{2}_{13} = \phi + 0.(\epsilon+0)^{*}.1$

$= 0.(\epsilon+0)^{*}.1$

$R^{2}_{12} = 0 + 0.(\epsilon+0)^{*}.(\epsilon+0)$

$= 0(\epsilon + (\epsilon+0)^{*}.(\epsilon+0))$ (r, r)

$= 0.(\epsilon+0)^{*}$ $\quad \epsilon + r.r^{*} = r^{*}$

$R^{3}_{12} = 0.(\epsilon+0)^{*} + 0.(\epsilon+0)^{*}1.(\epsilon+1)^{*}.\phi$

$= 0.(\epsilon+0)^{*}(\epsilon + 1.(\epsilon+1)^{*}.\phi)$ (r)

$= 0.(\epsilon+0)^{*}.\epsilon = 0.0^{*}$

$R^{3}_{13} = R^{2}_{13} + R^{2}_{13}(R^{2}_{33})\,R^{2}_{33}$

$= 0.(\epsilon+0)^{*}.1 + 0.(\epsilon+0)^{*}.1.(\epsilon+1)^{*}.(\epsilon+1)$

$= 0.(\epsilon+0)^{*}.1(\epsilon + (\epsilon+1)^{*}.(\epsilon+1))$

$= 0.(\epsilon+0)^{*}.1.(\epsilon+1)^{*}$

$= 0.0^{*}.1.1^{*}$

$RE = R^{3}_{12} + R^{3}_{13}$

$= 0.0^{*} + 0.0^{*}.1.1^{*}$

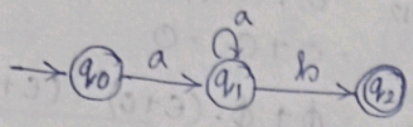
$= 0.0^{*}(\epsilon + 1.1^{*})$

$= 0.0^{*}.1.1^{*}$

Pumping lemma for RL

(i) Draw a DFA for $\{a^n b \mid n > 0\}$

$L = \{ab, aab, aaab, \ldots\}$



(ii) Draw a DFA for $L = \{a^n b^n \mid n > 0\}$

$L = \{ab, aabb, aaabbb, \ldots\}$

It is not possible to construct a DFA for this language as it accepts only equal no of a's and b's. since, finite automata is a simplest model it cannot remember the previous state.

---

*Pumping lemma is a tool which is used to test whether certain language are regular or not.

*Let L be a regular language, then there exist a constant 'n' (which depends on L i.e., no of states of the smallest automata that accepts L) such that for every string 'w' in L, w is divided into three parts as $w = uvw$ (or) $|xyz|$

where (i) $v \neq \epsilon$ i.e, $|v| \geq 1$

(ii) $|uv| \leq n$

(iii) $\forall k \geq 0$ $uv^k w$ should be in L

with $|w| \geq n$

for example,

if $w = aabb$

$w = uvw \Rightarrow u = a$

$v = ab$

$w = b$

steps to show that L is not regular using pumping lemma:

step-1: Assume that L is regular (Prove by contradiction)

step-2: Let n be a pumping lemma constant

step-3: Pick a suitable string w such that $|w| \geq n$ (lenght of w is greater than or equal to n)

step-4: show that for every legal decomposition of string w into uvw, there exist uv^iw that is not in L with a constraint of pumping lemma.

step-5: since uv^iw not in L, it can be concluded as L is not regular.

Problem:

1) Prove that $L = \{a^n b^n / n > 0\}$ is not regular.

solution:

step-1: Assume $L = \{a^n b^n / n > 0\}$ is regular.

step-2: Let n be a pumping lemma constant.

step-3: Let $w = a^n b^n$

$|w| = 2n \geq n$

$L = \{a^n b^n / n > 0\}$

$n=1 \quad , L = a^1 b^1 = ab$

$n=2 \quad , L = a^2 b^2 = aabb$

$w = xyz$

$x = a$

$y = a$

$z = bb$

(i) $y \neq \epsilon$

$a \neq \epsilon$

$\therefore$ condition is true

(ii) $|xy| \leq n$

$|aa| \leq 2$

$2 \leq 2$

∴ The condition is true

(iii) $xy^k z \in L$

$k = 0 \quad aa^0bb \in L$

$abb \notin L$

∴ The condition is False

∴ $L = \{a^n b^n / n > 0\}$ is not a regular

24/9/24

i) $L = \{0^n 1^n / n \geq 1\}$

$L = 0^n 1^n$

$n = 1 \rightarrow L = 0^1 1^1 = 01$

$n = 2 \rightarrow L = 0^2 1^2 = 0011$

$W = xyz$

$w = 0011$

$x = 0$

$y = 0$

$z = 11$

(i) $y \neq \epsilon$

$0 \neq \epsilon$

∴ condition is True

(ii) $|xy| \leq n$

$|00| \leq 2$

$2 \leq 2$

∴ The condition is true

(iii) $xy^k z \in L$

$k=0 \quad 00^0 11 \in L$

$011 \notin L$

∴ The condition is False

∴ $L = \{0^n 1^n \mid n \geq 1\}$ is not regular.

---

2) $L = \{a^p \mid p \text{ is Prime}\}$

$L = a^p$

$p = 2 \rightarrow L = a^2 = aa$

$p = 3 \rightarrow L = a^3 = aaa$

$w = xyz$

$w = aaa$

$x = a$

$y = a$

$z = a$

(i) $y \neq \epsilon$

$a \neq \epsilon$

∴ The condition is true

(ii) $|xy| \leq n$

$|aa| \leq 3$

$2 \leq 3$

∴ The condition is true

(iii) $xy^k z \in L$

$k=0 \quad aa^0 a \in L$

$aa \in L$

$k=1 \quad aa^1 a \in L$

$aaa \in L$

$L = \{00, 000, 00000, \ldots\}$

step-1: Assume that $L = \{a^p \mid p \text{ is Prime}\}$ is regular

step-2: Let 'n' be a pumping lemma constant.

step-3: $w = 0^p$

$|w| = p$ - Prime, $p > n$

$|w| = |uvw| = p$

~~$|uv^2w| = |uvvw|$~~

~~$= |uvw| + |v|$~~

Let $|v| = m$

so $|uw| = p - m$

Let v be pumped for $(p+1)$

$|uv^{(p+1)}w| = |uw| + (p+1)|v|$

$= (p-m) + (p+1)m$

$= p - m + pm + m$

$= p(1+m)$

A Prime number multiple with any other number (>1) will not be a Prime number.

$K = 2 \rightarrow aa^2a \in L$

$aaaa \notin L$ ∴ the condition is false.

∴ $L = \{a^p / p \text{ is Prime}\}$ is not regular.

---

3) $L = \{a^{2^i} / i \geq 1\}$

$L = a^{2^i}$

$i = 1 \rightarrow L = a^{2^1} = a^2 = aa$

$i = 2 \rightarrow L = a^{2^2} = a^4 = aaaa$

$w = aaaa$

$x = a$

$y = a$

$z = aa$

(i) $y \neq \epsilon$

$a \neq \epsilon$

∴ condition satisfied

(ii) $|xy| \leq n$

$|aa| \leq 2$

$2 \leq 2$

∴ condition satisfied.

(iii) $xy^K z \in L$

$K = 0$ $aa^0aa \in L$

$aaa \notin L$

∴ condition not satisfied

∴ $L = \{a^{2^i} / i \geq 1\}$ is not regular

(iii) $L = \{a^n b^m \mid m > n \text{ & } n \geq 1\}$

(iv) $L = \{ww^R \mid w \in (a+b)^*\}$

$w = xyz$ w^R - reverse a string

$n=1$ $L = abba$

4) $L = \{a^n b^m \mid m > n \text{ & } n \geq 1\}$

$L = a^n b^m$

$n=1, m=2 \rightarrow L = a^1 b^2 = abb$

$w = abb$

$x = \epsilon$

$y = a$

$z = bb$

(i) $y \neq \epsilon$

$a \neq \epsilon$

$\therefore$ condition satisfied

(ii) $|xy| \leq n$

$|\epsilon a| \leq 1$

$1 \leq 1$

(iii) $xy^k z \in L$

$k=0 \rightarrow \epsilon a^0 bb \in L$

$bb \notin L$

$\therefore$ condition is not satisfied

$\therefore L = \{a^n b^m \mid m > n \text{ & } n \geq 1\}$ is not regular.

5) $L = \{ww^R \mid w \in (a+b)^*\}$

$n=1$, $L = abba$

$w = xyz \rightarrow x = a$

$y = b$

$z = ba$

(i) $y \neq \epsilon$

$b \neq \epsilon$

$\therefore$ condition is satisfied.

(ii) $|xy| \leq n$

$|ab| \leq 1$

$2 \leq 1$

∴ condition not satisfied

∴ $L = \{ ww^R / w \in (a+b)^* \}$ is not regular.

25/9/24

CLOSURE PROPERTIES OF RL:

If the class of languages is closed under certain operation then the resultant of the operation also belongs to the same class of language. This property is called as closure property. There are 9 properties of closure property of regular language.

(i) The union of two RL is Regular
(ii) The Intersection of two RL is Regular
(iii) The complement of RL is Regular.
(iv) The difference between two RL is Regular
(v) The reversal of a RL is regular
(vi) The closure of a RL is regular
(vii) The concatenation of two RL is regular
(viii) The homomorphism of RL is regular
(ix) The Inversehomomorphism of RL is regular.

Theorem - 1:
Union of two RL is Regular.

Proof:

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ that

$M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$, Accepts languages L_1 and L_2 respectively.

construct a FA 'M' that accepts the language $L_1 \cup L_2$

Now $M = (Q, \Sigma, \delta, q_0, F)$

Where $Q = Q_1 \times Q_2$

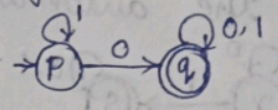
$\Sigma = \Sigma$

δ is defined as $\delta(Pq, a) = (\delta(P,a), \delta(q,a))$

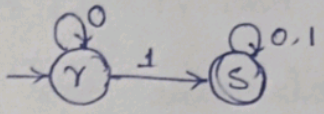
$q_0 = \{q_1, q_2\}$

$F = \{(Pq) \mid P \in F_1 \text{ or } Q \in F_2\}$

consider the automata M_1 =



M_2 =



$\delta(Pr,0) = \delta(P,0)\ \delta(r,0)$
$= qr$

$\delta(Pr,1) = \delta(P,1)\ \delta(r,1)$
$= PS$

$\delta(PS,0) = \delta(P,0)\ \delta(S,0)$
$= qs$

$\delta(PS,1) = \delta(P,1)\ \delta(S,1)$
$= PS$

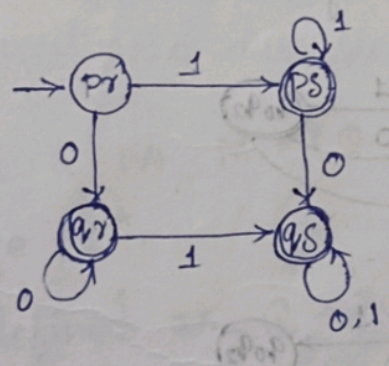
$\delta(qr,0) = \delta(q,0)\ \delta(r,0)$
$= qr$

$\delta(q,r,1) = \delta(q,1)\ \delta(r,1)$
$= qs$

$\delta(qs,0) = \delta(q,0)\ \delta(qs,0)$
$= qs$

$\delta(q,s,1) = \delta(q,1)\ \delta(s,1)$
$= qs$

M =



Theorem-2

Intersection of two RL is Regular.

Proof:

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ that accepts languages L_1 and L_2 respectively construct a FA 'M' that accepts the language $L_1 \cap L_2$.

Now $M = (Q, \Sigma, \delta, q_0, F)$ where $Q = Q_1 \times Q_2$

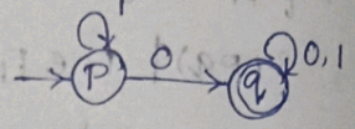
$\Sigma = \Sigma$

δ is defined as $\delta([p,q],a) = (\delta(p,a), \delta(q,a))$

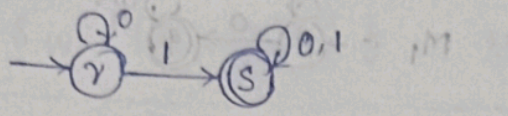
$q_0 = \{q_1, q_2\}$

$F = \{[pq] \mid p \in F_1 \text{ and } Q \in F_2\}$

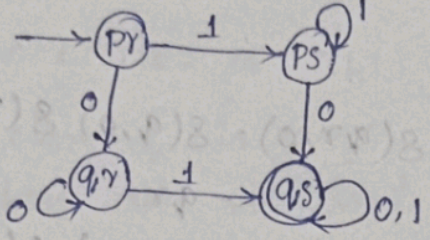
consider the automata M_1 =



M_2 =



M =



Theorem-3

complement of RL is Regular

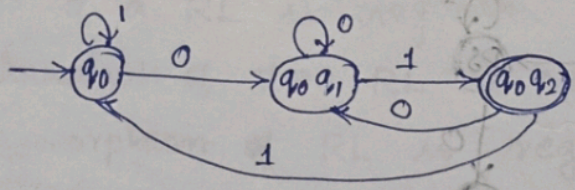
Proof:

Let L be a RL recognize by the DFA

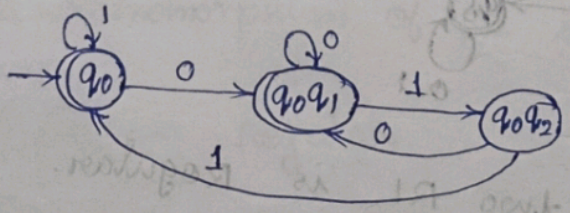
$A = (Q, \Sigma, \delta, q_0, F)$ and let $\bar{L} = L(B)$ where B is a DFA defined as $B = (Q, \Sigma, \delta, q_0, Q-F)$.

B is exactly same as A but the accepting states of A will be the non-accepting state in B

If A =



then B =



$B = Q-F$ of A B is a complement of A

Theorem - 4

Difference of two RL is Regular

Proof:

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$, $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ that accepts the language L_1 and L_2 respectively construct a FA 'M' that accepts the language $L_1 - L_2$

Now $M = (Q, \Sigma, \delta, q_0, F)$ Where $Q = Q_1 \times Q_2$

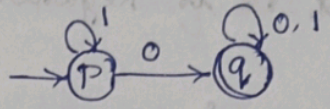
$\Sigma = \Sigma$

δ is defined as $\delta([pq], a) = (\delta(p, a)\ \delta(q, a))$

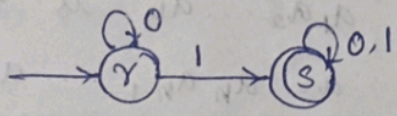
$q_0 = \{q_1 q_2\}$

$F = \{ pq \mid p \in F_1 \text{ and } q \notin F_2 \}$

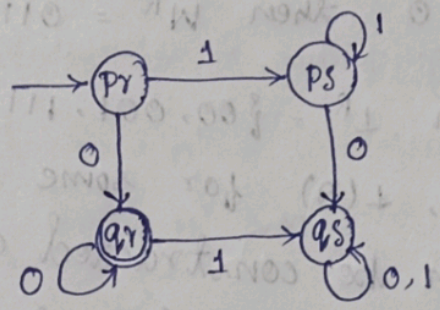
consider the automata $M_1 =$



$M_2 =$



$M =$



Theorem - 6

closure of RL is Regular

Proof:

Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ is recognize by language L_1

construct a FA $M = (Q, \Sigma, \delta, q_0, F)$ that recognize the language L^*.

M is defined as $Q = \{q_0\} \cup Q_1$

$\Sigma = \Sigma \cup \epsilon$

$q_0 = q_0$ (New start state)

$F = \{q_0\} \cup F_1$

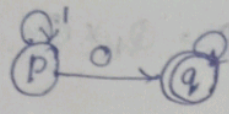
δ is defined as $\delta_1(q, a)$ $q \in Q_1$ & $q \notin F_1$

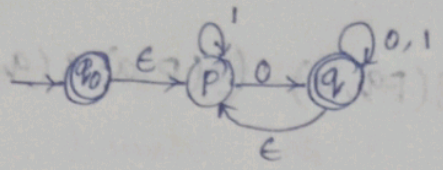
$\delta_1(q, a)$ $q \in Q$ & $q \in F_1$

$\delta_1(q, a) \cup \{q_1\}$ $q \in F_1$ & $a = \epsilon$

$\{q_1\}$ $q = q_0$ & $a = \epsilon$

ϕ $q = q_0$ & $a \neq \epsilon$

if L = 

then L^* is 

Theorem - 5

Reversal of RL is Regular

Proof:

The reversal of a string $a_1 a_2 \ldots a_n$ is the string return in backwards as $a_n a_{n-1} \ldots a_1$

For example, if $W = 00110$, then $W^R = 01100$

Let $L = \{00, 100, 111\}$ then $L^R = \{00, 001, 111\}$

Given a language L i.e. $L(A)$ for some finite automata A then L^R can be constructed as

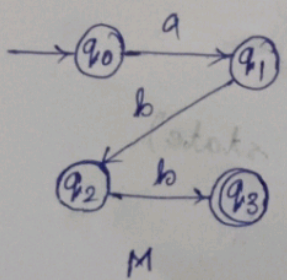
Step-1: Reverse all the arc (transition diagram) of A

Step-2: Make the start state of A as the only final state of new reversed automata.

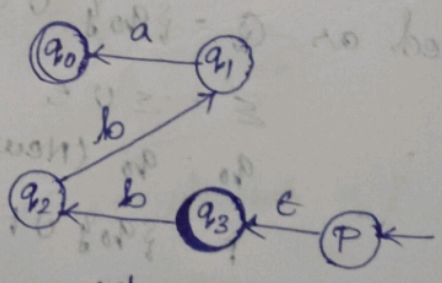
Step-3: create a new start state P_0 with a transition on ϵ to all the accepting states of A.

30/9/24

Given an automata. find the reversal



M



M'

Theorem - 8

Homomorphism of RL

It is a function on string that works on substituting a particular string for each symbol.

Let $h(0) = ab$, $h(1) = bb$

Find the homomorphic string for the given string 10101.

hs - bbabbbbabbbb

Prove that $L = \{0^{n^2} / n \geq 1\}$ is not regular

$L = \{0, 0000, \ldots\}$

step-1: Assume that $L = \{0^{n^2} / n \geq 1\}$ is regular

step-2: Let 'n' be a pumping lemma constant

step-3: Let $w = 0^{n^2}$

$|w| = n^2 > n$

step-4: $w = 0^{n^2}$

$w = uvw$

$|w| = |uvw| = n^2$, $\quad |uv^2w| > n^2$ — ①

$|uv^2w| = |uvvw|$

$= |uvw| + |v|$ $\quad |v|$ can take 1 to n

$|uv^2w| \leq n^2 + n$

$< n^2 + n + (n+1)$ ← Add one more term can be any value (constant)

$< n^2 + 2n + 1$

$|uv^2w| < (n+1)^2$ — ②

From ① & ②

$|uv^2w| > n^2$

but $< (n+1)^2$

$n^2 < |uv^2w| < (n+1)^2$

A perfect square cannot lie between two consecutive squares.

$L = \{0^{n^2} / n \geq 1\}$ is not regular