"Honour is the reward of virtue"

Module : 2 : Regular Expressions and Language

1 **Regular Expression** :- (Module 1 Pg 7 - Pg 14)

2 **Kleen's Theorem** :-

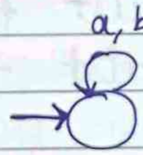
Statement :- A language is regular if and only if it is accepted by a finite automata

Proof :-

(1) If language is described by regular expression, then it is accepted by a finite automatan.

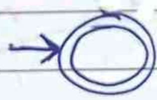
Best case :-

(i) ϕ



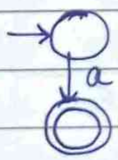
NFA possible with no accepting state

(ii) ϵ



NFA possible with start as final

(iii) a ($\forall a \in \Sigma$)



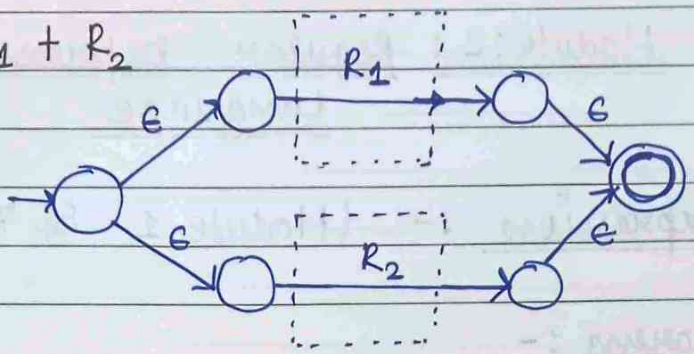
NFA possible with a transition on 'a' from start to final.

Inductive Hypothesis

Let Finite automata exist for the regular expression R_1 and R_2

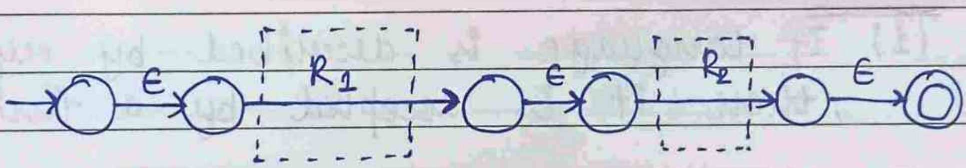
Inductive step

(i) $R_1 + R_2$



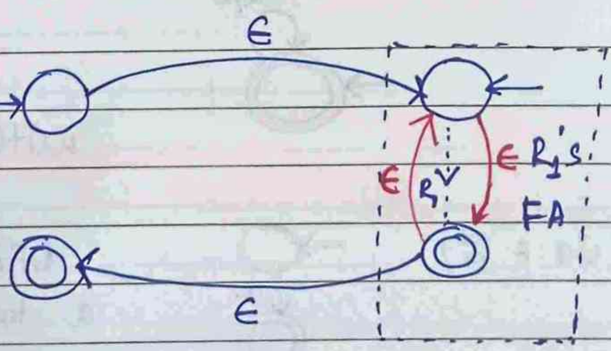
If NFA exist for R_1 and R_2 then it will exist for $R_1 + R_2$ as shown

(ii) $R_1 \cdot R_2$



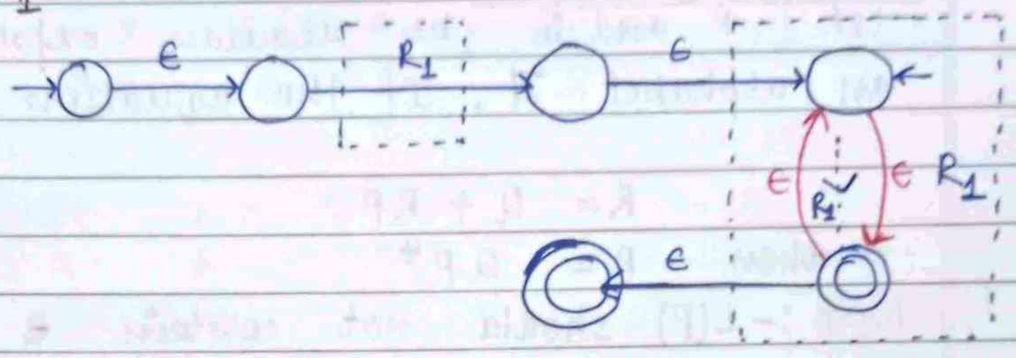
If NFA exist for R_1 and R_2 then it will exist for $R_1 \cdot R_2$ as shown

(iii) R_1^*



By adding ∈- transition from start state of R_1 to final state and back from final state to start state we can make NFA for R_1^*

(iv) R_1^+



If NFA exist for R_1 it will exist for R_1^+ by adjusting ε-transition as shown

Hence, any regular expression can be converted to finite automata.

(2) If a language is accepted by finite automaton then it can be described by regular expression

(i) Finite Automata can be converted into regular expression using state elimination method

(ii) Given $M = (Q, \Sigma, \delta, q_0, F)$ where $Q = \{q_1, q_2 .. q_n\}$

(iii) R_{ij}^{k} ~~representing~~ all string that take automaton from state q_i to q_j without passing through any intermediate state number with k or higher.

(iv) Eliminate the states using the recurrance until initial and final remains

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

3 Arden's Theorem :-

Let P and Q be regular expression over an alphabet Σ. If the equation is

$$R = Q + RP$$

then $R = QP^*$

Note :- L(P) should not contain ϵ

Proof :

As,

$$R = Q + RP$$

Substituting R again

$$\Rightarrow R = Q + (Q + RP)P$$
$$\Rightarrow R = Q + QP + RPP$$

Again substituting R

$$\Rightarrow R = Q + QP + (Q + RP)PP$$
$$\Rightarrow R = Q + QP + QPP + RPPP$$

Again substituting R

$$\Rightarrow R = Q + QP + QPP + (Q + RP)PPP$$
$$\Rightarrow R = Q + QP + QPP + QPPP + RPPP.$$

The following process will halt only if

$$R = Q + QP + QPP + QPPP \ldots\ldots$$

because $\epsilon \notin L(P)$

hence

$$R = Q(\epsilon + P + PP + PPP \cdots)$$

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

If $R = Q + PR$

$R = P^*Q$

4. Some Basic Rules while dealing with regular expression

1. ϵ based Rules :-

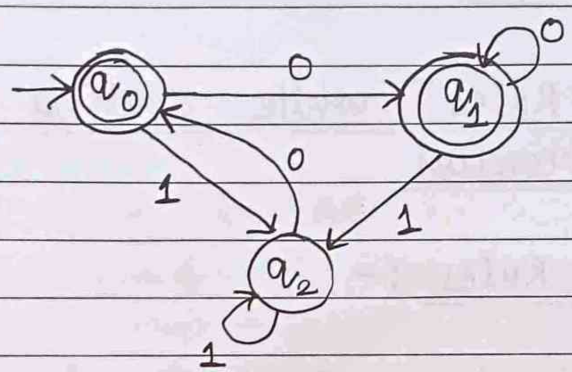
$$R + \epsilon = R + \epsilon$$
$$R \cdot \epsilon = R$$
$$\epsilon \cdot R = R$$
$$(R+\epsilon)^* = R^*$$
$$(R+\epsilon)^+ = R^*$$

2. ϕ based Rules :-

$$R + \phi = R$$
$$\phi + R = R$$
$$R \cdot \phi = \phi$$
$$\phi \cdot R = \phi$$
$$\phi^* = \epsilon$$
$$\phi^+ = \phi$$
$$(R+\phi)^* = R^*$$
$$(R+\phi)^+ = R^+$$
$$\phi + \epsilon = \epsilon$$
$$\phi \cdot \epsilon = \phi$$
$$\epsilon \cdot \phi = \phi$$

5 Applications of Arden's Theorem

2023-24 2(b) Example :1 : Compute the regular expression using Arden's Theorem for the following DFA



Let R_0 be the regular expression that represents all those strings which ends at q_0 from start state, and so is (q_1, q_2) R_1 and R_2.

$$R_0 = R_2 0 + \epsilon \quad \text{— (i)}$$

→ because R_0 is start state

$$R_1 = R_0 0 + R_1 0 \quad \text{— (ii)}$$

$$R_2 = R_0 1 + R_1 1 + R_2 1 \quad \text{— (iii)}$$

Applying Arden's Theorem in (ii)

$$\underbrace{R_1}_{R} = \underbrace{R_0 0}_{Q} + \underbrace{R_1}_{R} \underbrace{0}_{P}$$

$$\Rightarrow \underbrace{R_1}_{R} = \underbrace{R_0 0}_{Q} \underbrace{0^*}_{P^*} \quad \text{— (iv)}$$

Substituting (i) and (iv) in (iii)

$$R_2 = (R_2 0 + \epsilon)1 + (R_0 00^*)1 + R_2 1$$

$$\Rightarrow R_2 = R_2 01 + 1 + R_0 00^* 1 + R_2 1$$

$$\Rightarrow \underbrace{R_2}_{R} = \underbrace{(R_0 00^* 1 + 1 + R_0 00^* 1)}_{Q} + \underbrace{R_2}_{R} \underbrace{(01+1)}_{P} \quad \text{—(v)}$$

Applying Arden's Theorem in (v)

$$R_2 = (R_0 00^* 1 + 1 + R_0 00^* 1)(01+1)^* \quad \text{—(vi)}$$

Subtituting (vi) in (i)

$$R_0 = [R_0 00^* 1 (01+1)^* + 1(01+1)^* + R_0 00^* 1 (01+1)^*]0$$

$$R_0 = R_0 00^* 1 (01+1)^* 0 + 1(01+1)^* 0 + R_0 00^* 1 (01+1)^* 0$$

$$\underbrace{R_0}_{R} = \underbrace{1(01+1)^* 0}_{Q} + \underbrace{R_0}_{R} \underbrace{[00^* 1 (01+1)^* 0 + 00^* 1 (01+1)^* 0]}_{P} \quad \text{—(iv)}$$

Applying Arden's Theorem in (vii)

$$R_0 = 1(01+1)^* 0 (00^* 1 (01+1)^* 0 + 00^* 1 (01+1)^* 0)^*$$

$$R_0 = 1(01+1)^* 0 (00^* 1 (01+1)^* 0)^* \quad \text{—(viii)}$$

Substituting (viii) in (iv)

$$R_1 = 1(01+1)^* 0 (00^* 1 (01+1)^* 0)^* 00^*$$

hence,

Regular expression for given DFA

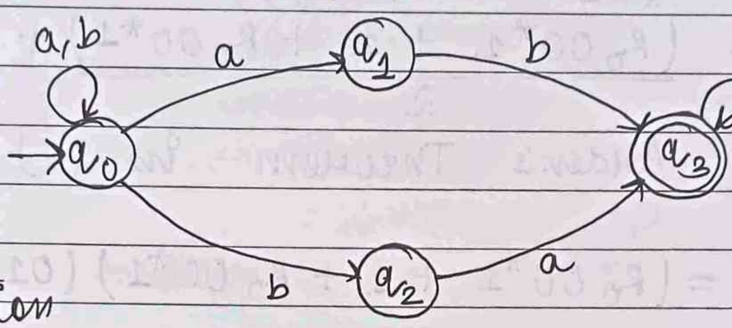
$$= R_0 + R_1 \quad \text{(All final states)}$$

$$\boxed{R_{DFA} = 1(01+1)^* 0 (00^* 1 (01+1)^* 0)^* (\epsilon + 00^*)}$$

2022-23 4(a)

Example 2 : Find the regular expression corresponding to the finite automata given below :

We are solving it using Arden's theorem but it is not recommended. Use state elimination technique instead (81 pg).



We cannot apply Arden's theorem directly on NFA due to non-determinism

		a	b	c
A	$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0, q_2\}$	ϕ
B	q_1	ϕ	$\{q_3\}$	ϕ
C	q_2	$\{q_3\}$	ϕ	ϕ
D	$* \; q_3$	ϕ	ϕ	$\{q_3\}$
E	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$	ϕ
F	$\{q_0, q_2\}$	$\{q_0, q_1, q_3\}$	$\{q_0, q_2\}$	ϕ
G	ϕ	ϕ	ϕ	ϕ
*H	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$	$\{q_0, q_2\}$	$\{q_3\}$
*I	$\{q_0, q_1, q_3\}$	$\{q_0, q_1\}$	$\{q_0, q_2, q_3\}$	$\{q_3\}$

δ_{DFA}	a	b	c
→ A	E	F	G
B	G	D	G
C	D	G	G
* D	G	G	D
E	E	H	G
F	I	F	G
G	G	G	G
* H	I	F	D
* I	E	H	D

Minimizing the given DFA :-

0- Equivalent

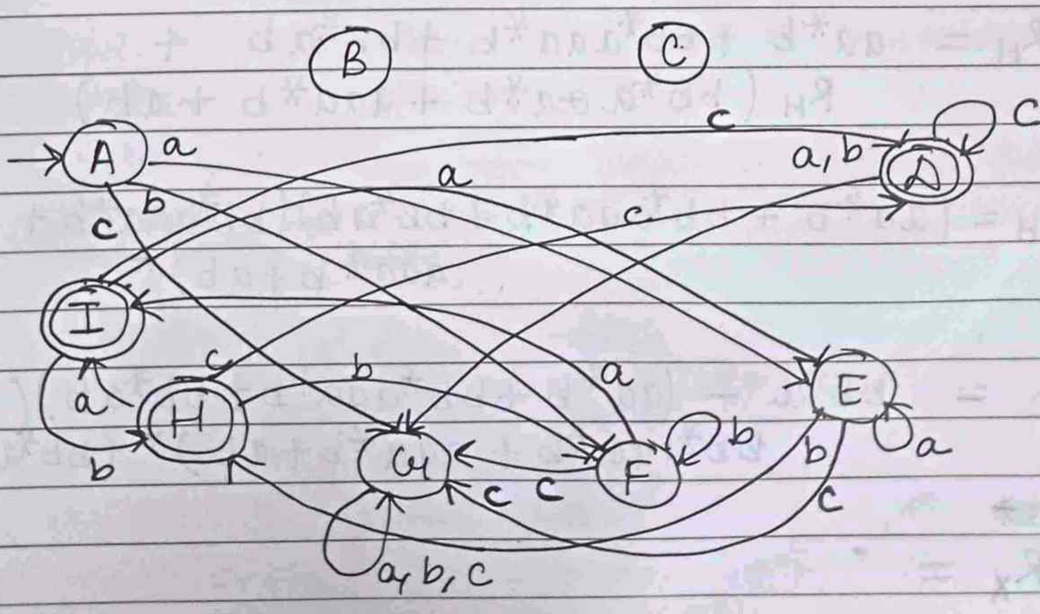
{A, B, C, E, F, G} {D, H, I}

1- Equivalent

{A, G} , {B, E} , {C, F} , {D}, {H}, {I}

2- Equivalent

{A}, {G} , {B},{E}, {C}, {F}, {D}, {H}, {I}



B and C are unreachable

$R_A = \epsilon$

$R_D = R_D c + R_I (c) + R_H (c)$

$R_E = R_E a + R_A a + R_I (a)$

$R_F = R_F b + R_A b + R_H b$

$R_G = R_G (a+b+c) + \epsilon$

$R_H = R_E b + R_I b$

$R_I = R_F a + R_H a$

$R_D = (R_I c + R_H c) c^*$

$R_E = (R_I a + a) a^*$

$R_F = (R_H b + b) b^*$

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

$R_H = R_I aa^*b + aa^*b + R_I b$

$R_I = R_H bb^*a + bb^*a + R_H a$

$$R_H = aa^*b + (R_H bb^*a + bb^*a + R_H a) aa^*b + R_H bb^*a b + bb^*a b + R_H ab$$

$$R_H = aa^*b + bb^*aaa^*b + bb^*ab + R_H (bb^*aaa^*b + aaa^*b + ab)$$

$$R_H = (aa^*b + bb^*aaa^*b + bb^*ab)(bb^*aaa^*b + aaa^*b + ab)^*$$

$$R_I = bb^*a + (aa^*b + bb^*aaa^*b + bb^*ab)(bb^*aaa^*b + aaa^*b + ab)^*(bb^*a + a)$$

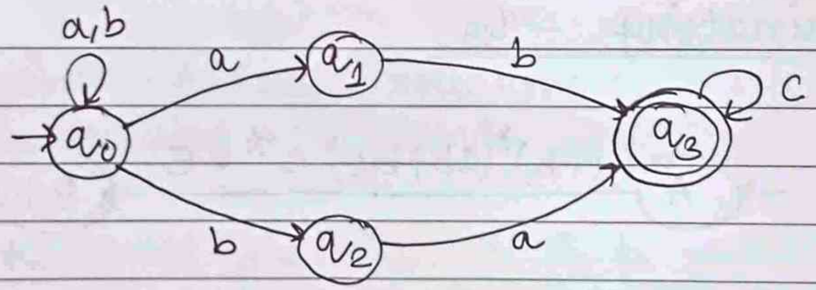
$$R_D = bb^*ac^+ + R_H(bb^*a + a)c^+ + R_H c^+$$

Regular expression for DFA

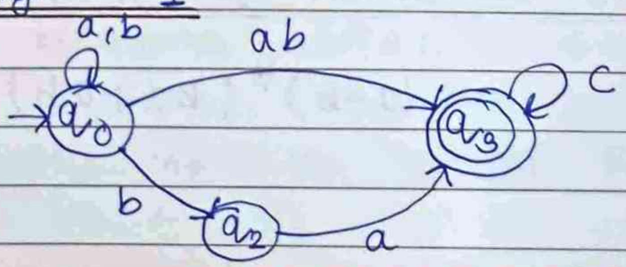
$$= R_H + R_I + R_D$$

4 <u>State elimination Technique</u> : Pg (53) Module 1

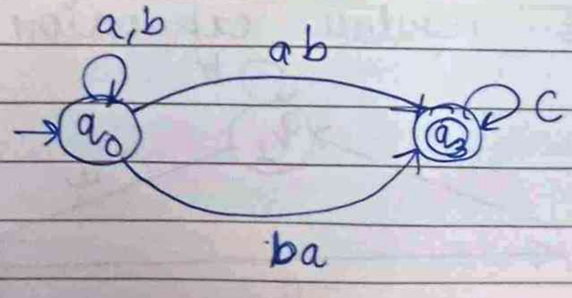
2022-23 4(a) <u>Example 1</u> : Find the regular expression corresponding to the finite automata given below:-



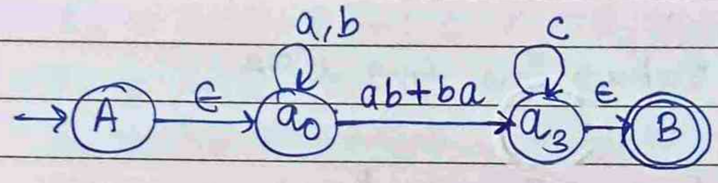
<u>Eliminating q_1</u>



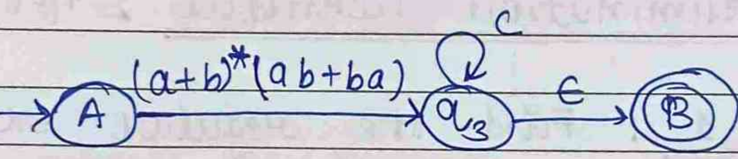
<u>Eliminating q_2</u>



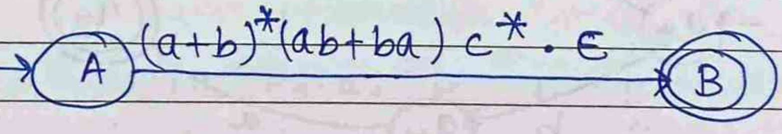
Extending start and final state



Eliminating q_0



Eliminating q_3



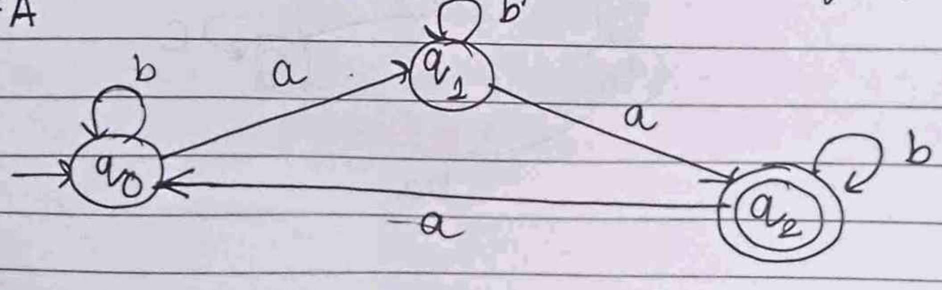
Regular expression for given NFA

$$= (a+b)^*(ba+ab)\cdot c^*$$

2020-21 4(a)

Example : 2

Find the regular expression of given FA



1. Using Arden's Theorem :-

$$R_0 = \epsilon + R_0 b + R_2 a \quad \text{—(i)}$$
$$R_1 = R_0 a + R_1 b \quad \text{—(ii)}$$
$$R_2 = R_1 a + R_2 b \quad \text{—(iii)}$$

Applying Arden's theorem in (i)

$$R_0 = (R_2 a + \epsilon) b^* \quad \text{—(iv)}$$

Applying Arden's theorem in (ii)

$$R_1 = R_0 a b^* \quad \text{—(v)}$$

Applying Arden's theorem in (iii) after substituting (v) and (iv)

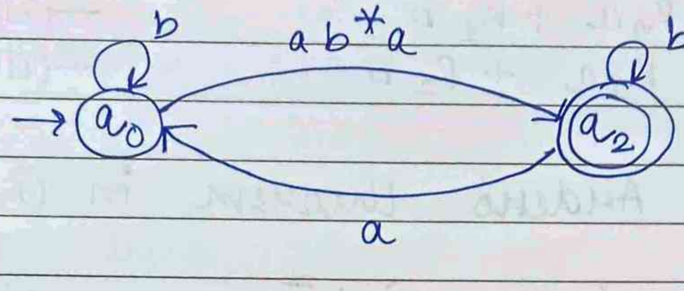
$$R_2 = (R_0 a b^*) a + R_2 b$$
$$R_2 = R_2 a b^* a b^* a + b^* a b^* a + R_2 b$$
$$R_2 = b^* a b^* a + R_2 (a b^* a b^* a + b)$$
$$R_2 = b^* a b^* a (a b^* a b^* a + b)^*$$

Regular Expression of given FA $= R_2$

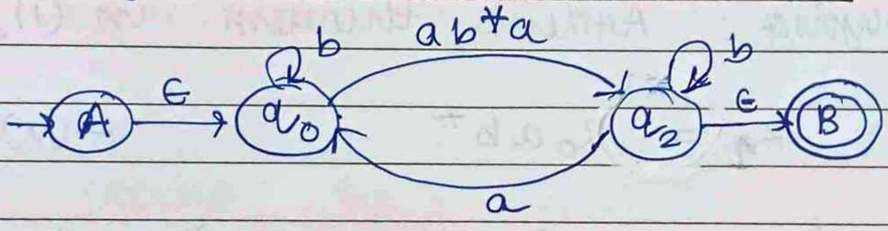
$$\boxed{R_{FA} = b^* a b^* a (a b^* a b^* a + b)^*}$$

2. Using State Elimination :

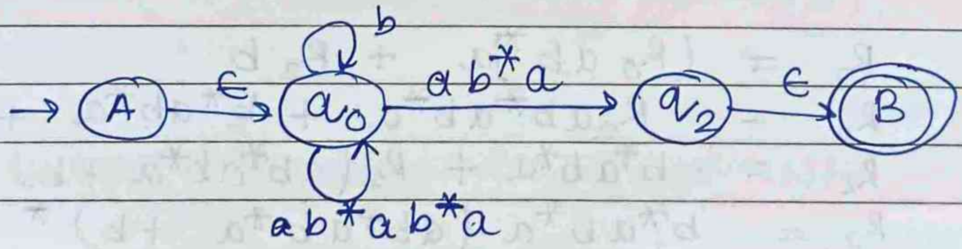
Eliminating q_1



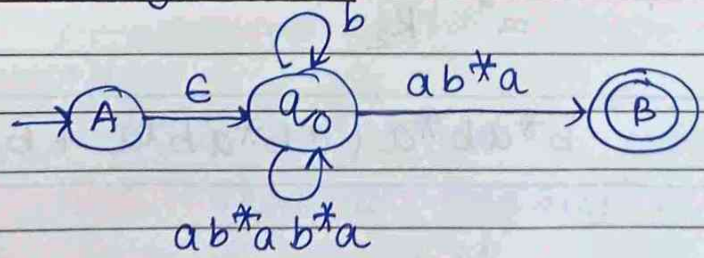
Extending start and final state



Eliminating cycle (on q_0).



Eliminating q_2



Eliminating q_0

$$\rightarrow (A) \xrightarrow{(b+ab^*ab^*a)^*ab^*a} ((B))$$

Regular expression for given FA

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

5 <u>Pumping Lemma</u>

<u>Statement</u> :-

Let L be an infinite Regular language and M be a finite automata such that $L(M) = L$ then there exist a constant 'n' such that for every string w in L where $|w| \geq n$ we can break w into three strings $w = xyz$ such that

(i) $|xy| \leq n$

(ii) $y \neq \epsilon$ i.e $|y| \geq 1$

(iii) $\forall i \geq 0, \; xy^iz \in L$

<u>Proof</u> :-

1. As L is regular, then the finite automata M such that $L(M) = L$ be defined as $M = (Q, \Sigma, \delta, q_0, F)$. Let $Q = \{q_0, q_1, q_2 \ldots q_n\}$

2. We have choosen a string w such that $|w| \geq n$ (the no of states in M)

3. While reading w, M will read symbol by symbol and do transitions as $q_0, q_i, q_j \ldots q_f$

4. Since the sequence $q_0, q_i, q_j \ldots q_f$ has $|w|+1$ entries which is greater than n, By using pigeonhole principle at least one state must be repeated as

$$q_0, q_i, q_j \ldots q_r \ldots q_r \ldots q_f$$

5 Let $w = xyz$.
Upon reading x we are in q_r
Upon reading y we are in q_r again
Upon reading z we are at q_f

6 We can apply constraints as
(i) $|xy| \leq n$: becoz xy refers to the part where repetition has occured and repetition will occur only on or before reading n symbols
(ii) $y \neq \epsilon$: becoz at least one symbol must be read to come back to same state
(iii) $xy^iz \in L \ \forall i \geq 0$: becoz every repetion of i leads back to same state

Points to Ponder

1. If language is regular it must satisfy the pumping lemma
2. If language does not satisfy the pumping lemma it is definitely not regular.
3. If a language satisfies pumping lemma it may or maynot be regular

6. Applications of Pumping Lemma

Working Procedure

Step 1 : Choose any constant n
($n \geq$ no of states in DFA for L)

Step 2 : Select any string w that belongs to L such that $|w| \geq n$

Step 3 : Divide the string w into 3 parts
$w = xyz$ such that
$|xy| \leq n$
$|y| \geq 1$

Step 4 : If

$\forall i \geq 0$
$xy^iz \in L$

This will happen only if language was regular
(But may happen for some non-regular as well)

$\exists i \geq 0$
$xy^iz \notin L$

Be sure language is non regular
(This will never happen for any regular language)

Answer 4a 2023-24, 4b 2022-23

State Pumping Lemma for Regular Language. Show that the given language $L = \{a^p \mid$ where p is a prime$\}$ is not regular

Statement :-

Let L be an infinite regular language.
" For all regular languages L there exists an integer 'n' such that : for all strings w in L, where $|w| \geq n$ there exist a partition of w as $w = xyz$ such that

(i) $|xy| \leq n$
(ii) $|y| \geq 1$
(iii) $\forall i \geq 0 \; xy^iz \in L$ "

Given :

$L = \{a^p \mid p \text{ is prime}\}$

Let $n = 5$

Let $w = a^7$ $(a^7 \in L)$ $|w| \geq n$

Let $w = xyz$
where

$x = aa$
$y = a$ $|xy| \leq n$
$z = aaaa$ $|y| \geq 1$

Pumping the strings : xy^iz $\forall i \geq 0$

$i = 0$: $xy^0z = aaaaaa \notin L$

$i = 1$: $xy^1z = a^7 \in L$

$i = 2$: $xy^2z = a^8 \notin L$

Hence there exist $i = 0, 2$ such that $xy^iz \notin L$

This language is not regular.

2021-22

Question 5(b) : Prove that the following language $L = \{a^n b^n : n \geq 0\}$ is not a regular language

Let $n = 4$

Let $w = aaabbb$, $|w| \geq 4$, $w \in L$

Let $w = xyz$
where $x = aa$
$y = ab$ $\quad |xy| \leq 4$
$z = bb$ $\quad |y| \geq 1$

Pumping the strings : $xy^iz \ \forall i \geq 0$

$i = 0$: $xy^0z = aabb \in L$
$i = 1$: $xy^1z = aaabbb \in L$
$i = 2$: $xy^2z = aaababbb \notin L$
$i = 3$: $xy^3z = aaabababbb \notin L$

Hence there exist $i = 2, 3 \ldots$ such that $xy^iz \notin L$

This language is not regular.

2020-21 4(b)

Using pumping lemma for regular languages prove that language $L = 0^{n^2}, n \geq 1$ is not regular

Let $p = 4$

let $w = 0^9$ $(0^9 = 0^{3^2} \in L)$ $|w| \geq p$

let $w = xyz$
$x = 000$
$y = 0$ $\quad |xy| \leq p$
$z = 00000$ $\quad |y| \geq 1$

Pumping the strings : xy^iz $\forall i \geq 0$

$i=0$: $xy^0z = 000\,00000 = 0^8 \notin L$
$i=1$: $xy^1z = 000000000 = 0^9 \in L$
$i=2$: $xy^2z = 000\ 00\ 00000 = 0^{10} \notin L$

hence there exist $i = 0, 2$ where $xy^iz \notin L$
∴ This language is not regular.