



#TAFL UNIT 2 - #TAFL UNIT 2

Income Tax Law & Practices (SS Jain Subod PG(Autonomous) College)



Scan to open on Studocu

AKTU RKC STUDY MATERIAL



WEBSITE – www.akturkcstudymaterial.com

Main discussion group (AT TELEGRAM)

Other group at telegram

First year all notes

ALL AKTU NOTES

1- FIRST YEAR DISCUSSION GROUP

2- CSE & IT STUDENT COMMUNITY

3- ME STUDENT COMMUNITY

4- EE & EEE STUDENT COMMUNITY

5- ECE STUDENT COMMUNITY

6- CIVIL ENGINEERING COMMUNITY

WEBSITE – www.akturkcstudymaterial.com

Thanks for join "[AKTU RKC TECHNO FAMILY](http://www.akturkcstudymaterial.com)"

Join us by touching blue text and social media logo

This document is available free of charge on



Downloaded by Amit Sharma (amitmudgal69@gmail.com)

Unit-2

①
Regular Expression: are used to represent the strings accepted by Finite Automata (Regular languages) in an algebraic manner

Features of Regular expression:

- ① Any terminal symbol, λ and ϕ are regular expression
- ② The union of two regular expression ~~is also~~ R_1 and R_2 written as $R_1 + R_2$ is also regular expression
- ③ The Concatenation of two regular expression R_1 and R_2 , is written as $R_1 R_2$ is also regular expression.
- ④ The Closure of a regular expression R , written as R^* is also regular expression.
- ⑤ If R is a regular expression, then (R) is also a regular expression.

Regular Set: Any set represented by a regular expression is called "Regular Set".

for $a, b \in \Sigma$

- (1) Regular expression a is represented by regular set $\{a\}$.
- (2) Regular expression $a+b$ is represented by regular set $\{a, b\}$.
- (3) Regular expression ab is represented by regular set $\{ab\}$.
- (4) Regular expression a^* is represented by regular set $\{\lambda, a, aa, aaa, aaaa, \dots\}$
- (5) Regular expression $(a+b)^*$ is represented by regular set $\{a, b\}^*$

Represent the following regular set by regular expressions: (2)

S.No.	Regular Set	Regular expression
1.	$\{101\}$	101
2.	$\{abba\}$	$abba$
3.	$\{01, 10\}$	$01+10$
4.	$\{\lambda, ab\}$	$\lambda+ab$
5.	$\{abb, a, b, bba\}$	$abb+a+b+bb a$
6.	$\{\lambda, 0, 00, 000, \dots\}$	0^*

Describe the following Set by regular expressions

(a) The set of all strings of 0's and 1's ending in 00.

Ans $(0+1)^*00$

(b) The set of all strings of 0's and 1's beginning with 0 and ending with 1.

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

(c) $L = \{\lambda, 11, 1111, 111111, \dots\}$

Ans $L = \{\lambda, 11, (11)^2, (11)^3, \dots\}$
 $= (11)^*$

(d) The set of all strings in which every 0 is immediately followed by at least two 1's. $\Sigma = \{0, 1\}$

Ans $(1+011)^*$

Find the regular expression corresponding to each of the following subset of $\{a, b\}^*$.

(a) The set of all strings containing exactly two a's.

Ans (a) $b^* a b^* a b^*$

(b) The set of all strings containing at least 2 a's.

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

(c) The set of all strings containing at most 2 a's.

$b^* + b^* a b^* + b^* a b^* a b^*$
 $\downarrow \quad \downarrow \quad \downarrow$
 (zero a) (one a) (two a)

(d) The set of all strings containing the substring 'aa'.

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

(e) $L = \{ a, aa, aaa, aaaa, \dots \}$
 $= a \{ a, a, aa, aaa, \dots \}$

$= a(a)^*$

Arden's theorem to Convert NFA to regular expression

Arden's theorem is used to convert NFA into regular expression.

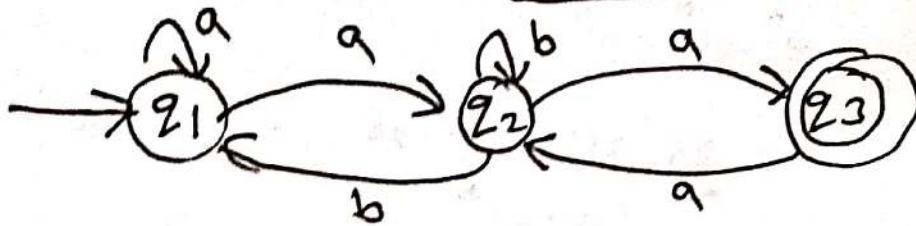
Arden's theorem: If P and Q are two regular expressions over Σ . If P does not λ , then

the following equation in R ,

$R = Q + RP$ has a unique solution

given by $R = QP^*$.

Example Show using Arden's theorem that (4)
 Strings recognized by given transition system is $(a+ a(b+aa)^*b)^*a(b+aa)^*a$



Ans

(i) for each state, we need to find the inputs that are coming to that state.

(ii) kindly ensure that NFA does not have λ -moves for using Arden's theorem.

$$q_1 = q_1 a + q_2 b + 1 \quad \text{--- (1)}$$

$$q_2 = q_1 a + q_2 b + q_3 a \quad \text{--- (2)}$$

$$q_3 = q_2 a \quad \text{--- (3)}$$

putting the value of q_3 from eqⁿ (3) to eqⁿ (2)

$$q_2 = q_1 a + q_2 b + q_2 a a$$

$$\frac{q_2}{R} = \frac{q_1 a}{Q} + \frac{q_2 (b+aa)}{R} \quad \text{P}$$

$$R = Q P^*$$

$$q_2 = q_1 a (b+aa)^* \quad \text{--- (4)}$$

putting the value of q_2 from eqⁿ (4) to eqⁿ (1)

$$q_1 = q_1 a + q_1 a (b+aa)^* b + 1$$

$$\frac{q_1}{R} = \frac{q_1 (a + a(b+aa)^*b)}{R} + \frac{1}{Q}$$

$$q_1 = d \cdot (a + a(b+aa)^*b)^*$$

$$q_1 = (a + a(b+aa)^*b)^* \quad \text{--- (5)}$$

(using Arden's theorem)

$R = Q + RP$ has solution, $R = QP^*$

(Use Arden's theorem)

$R = Q + RP$ has solution

$R = QP^*$

Putting the value of z_1 from eqn (5) to eqn (4)

$$z_2 = z_1 a (b + aa)^* \quad (4)$$

$$z_2 = (a + a(b + aa)^* b)^* a (b + aa)^*$$

Putting the value of z_2 in eqn (3)

$$z_3 = z_2 a$$

$$z_3 = (a + a(b + aa)^* b)^* a (b + aa)^* a$$

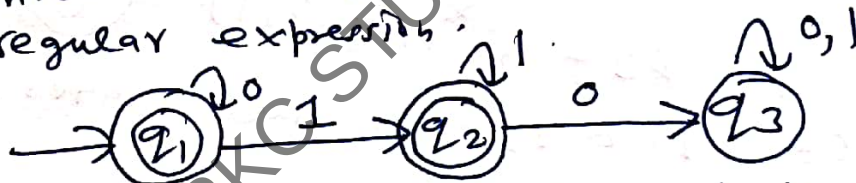
Since z_3 is the final state, so it shows the set of strings accept in regular expression form

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

Ans

ex 2

Convert the given transition system (NFA) into regular expression.



Ans

Since it does not have λ -transition, so we can apply Arden's theorem.

$$z_1 = z_1 0 + \lambda \quad (1)$$

$$z_2 = z_1 1 + z_2 1 \quad (2)$$

$$z_3 = z_2 0 + z_3 (0 + 1) \quad (3)$$

We can directly apply Arden's theorem in this $R = Q + RP$ has solution $R = QP^*$

$$\frac{z_1}{R} = \frac{z_1 0 + \lambda}{RP + Q}$$

$$z_1 = \lambda \cdot 0^* = 0^* \quad (4)$$

So $z_1 = 0^*$

(6)

Putting value of z_1 in eqⁿ (2)

$$z_2 = z_1 + z_2$$

$$z_2 = \frac{0^*}{R} + \frac{z_2}{R}$$

$$R = Q + RP$$

with solution
 $R = QP^*$

$$z_2 = 0^* 1 1^* \text{ --- (5)}$$

Putting value of z_2 in eqⁿ (3)

$$z_3 = z_2 0 + z_3 (0 + 1)$$

~~$z_3 = z_2$~~

$$z_3 = \frac{0^* 1 1^* 0 + z_3 (0 + 1)}{R}$$

$$R = Q + RP$$

$$R = QP^*$$

$$z_3 = 0^* 1 1^* 0 (0 + 1)^* \text{ --- (6)}$$

Since z_1 & z_2 are final states, so solution is given by

$$\begin{matrix} z_1 + z_2 \\ \downarrow \downarrow \\ 0^* + 0^* 1 1^* \end{matrix}$$

Identities for regular expression

- ① $\phi + R = R$
- ② $\phi R = R\phi = \phi$
- ③ $\lambda R = R\lambda = R$
- ④ $\lambda^* = \lambda$ and $\phi^* = \lambda$
- ⑤ $R + R = R$
- ⑥ $R^* R^* = R^*$
- ⑦ $RR^* = R^* R$

- ⑧ $(R^*)^* = R^*$
- ⑨ $\lambda + RR^* = R = \lambda + R^* R$
- ⑩ $(PQ)^* P = P(QP)^*$
- ⑪ $(P+Q)^* = (P^* + Q^*)^* = (P^* Q^*)^*$
- ⑫ $(P+Q)R = PR + QR$
and
 $R(P+Q) = RP + RQ$

Represent the following sets by regular expressions

(a) $\{0, 1, 2\}$
 $0 + 1 + 2$

(b) $\{1^{2n+1} \mid n \geq 0\}$
 $= \{1^{2 \times 1 + 1}, 1^{2 \times 2 + 1}, 1^{2 \times 3 + 1}, 1^{2 \times 4 + 1}, 1^{2 \times 5 + 1}, \dots\}$
 $= \{111, 11111, 1111111, 111111111, 11111111111, \dots\}$
 $= 111 \{11\}^*$

(c) $\{w \in \{a, b\}^* \mid w \text{ has only one } a\}$
 $= b^* a b^*$

(d) The set of all strings over $\{0, 1\}$ which has at most two zeros.
 $1^* + 1^* 0 1^* + 1^* 0 1^* 0 1^*$

(e) $\{a^2, a^5, a^8, \dots\}$
 $= a^2 \{1, a^3, a^6, \dots\}$
 ~~$= a^2 (a^3)^*$~~
 $= a^2 (a^3)^*$
 $= a^2 (aaa)^*$

(f) $\{a^n \mid n \text{ is divisible by } 2 \text{ or } 3 \text{ or } n = 5\}$
 $= (aa)^* + (aaa)^* + aaaaa$

(g) The set of all strings over $\{a, b\}$ beginning & ending with a .
 $a(a+b)^*a$

Proof of Arden's theorem

(8)

Arden's theorem: "Let P and Q be two regular expressions over Σ . If P does not contain λ , then the following equation in R

$$R = Q + RP$$

has a unique solution given by $R = QP^*$.

Proof:

Since $R = Q + RP$

$$= Q + (Q + RP)P$$

(Putting $R = Q + RP$)

$$= Q + QP + RP^2$$

($R = Q + RP$)

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

$$= Q + QP + QP^2 + RP^3$$

$$= Q + QP + QP^2 + QP^3 + QP^4 + \dots$$

$$= Q(1 + P + P^2 + P^3 + P^4 + \dots)$$

$$R = QP^*$$

~~Conversion from Finite Automata (FA) to Regular Expression (RE)~~

~~Basic Rules for Conversion:~~

~~(i) $\lambda \rightarrow \emptyset$~~

Conversion from Regular Expression (RE) to Finite Automata (FA)

9

Rules for Conversion:

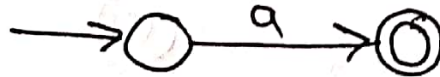
(i) $R = \Lambda$



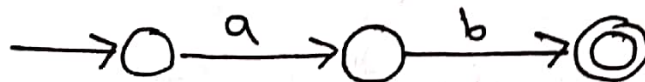
(ii) $R = \phi$



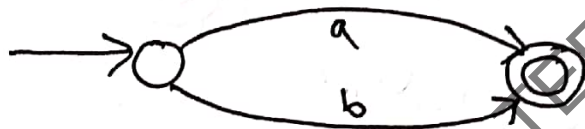
(iii) $R = a$



(iv) $R = ab$



(v) $R = a+b$



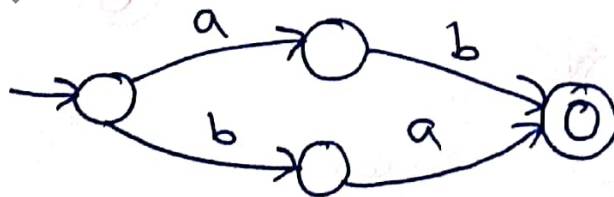
(vi) $R = a^*$



(vii) $R = (a+b)^*$



(viii) $R = \underline{a}b + b\underline{a}$

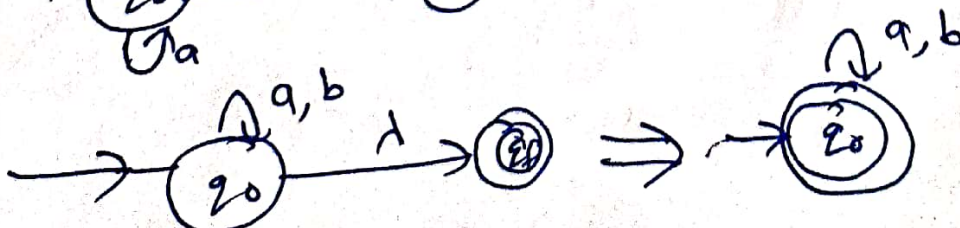
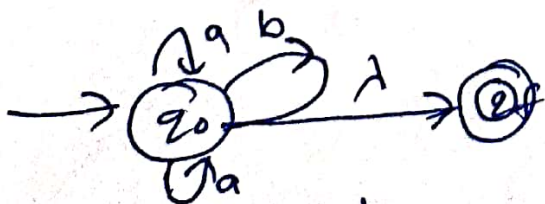
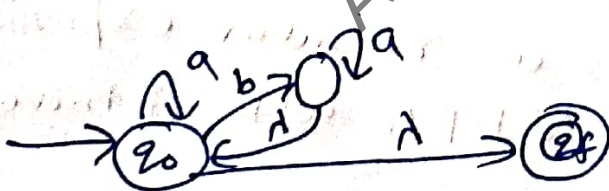
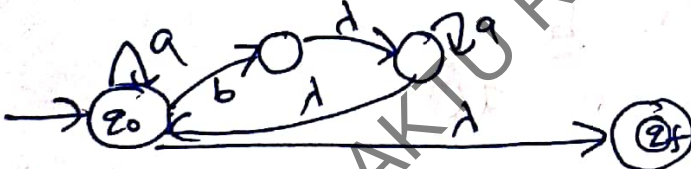
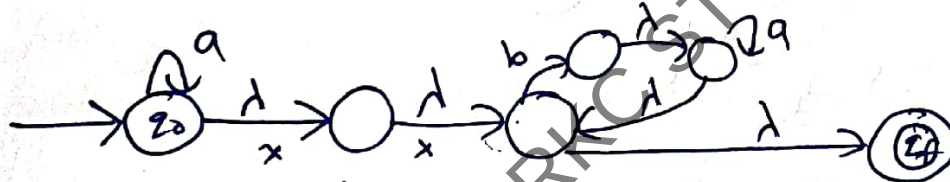
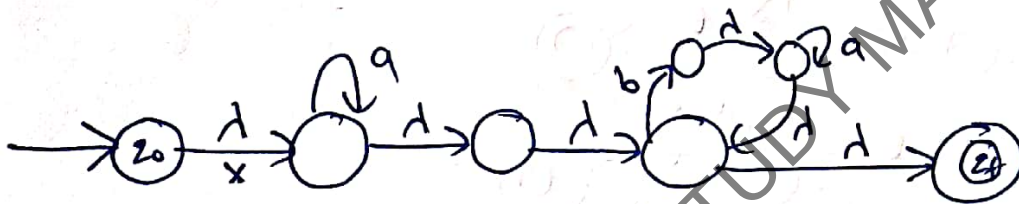
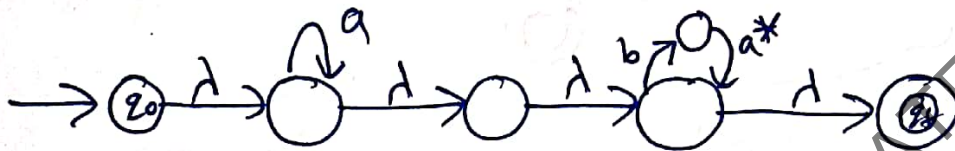
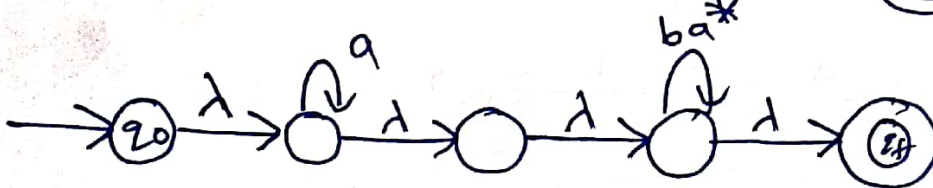
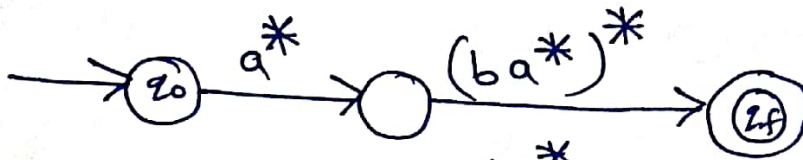
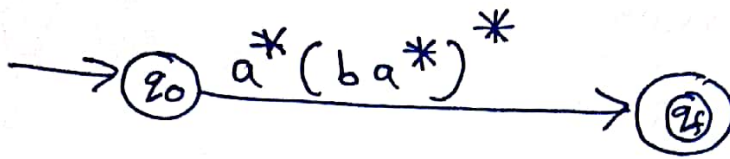


Kleene's theorem: If R is a regular expression then there exist an NFA with Λ -moves

Example 1:

10

Convert $a^*(ba^*)^*$ into Finite Automaton (FA)

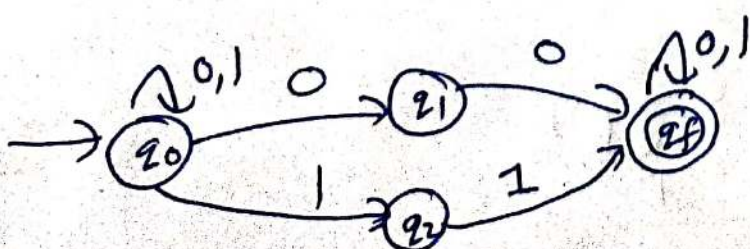
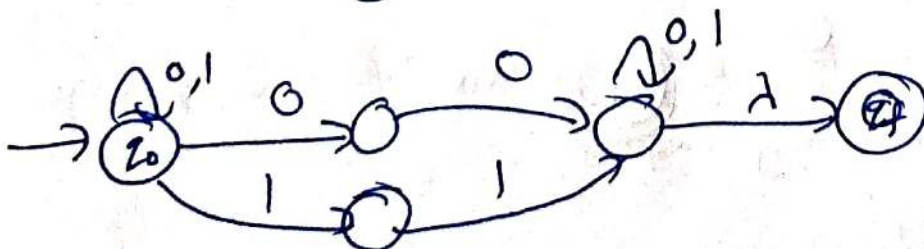
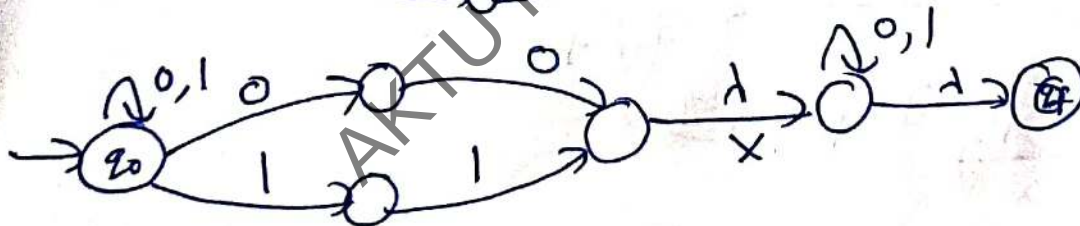
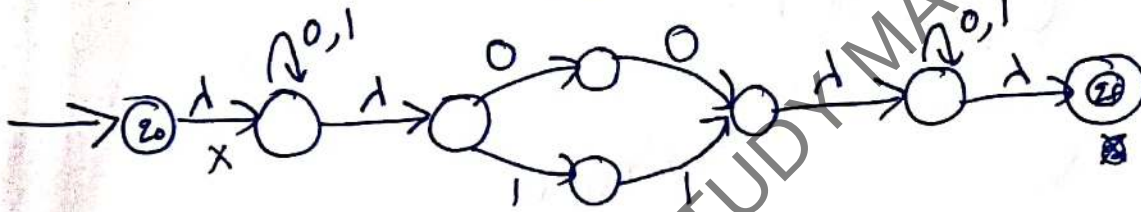
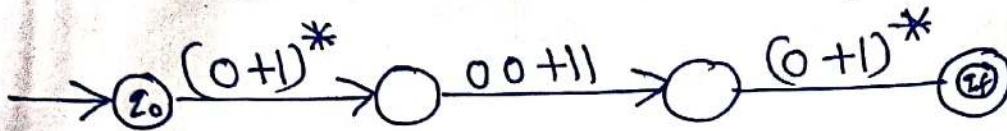
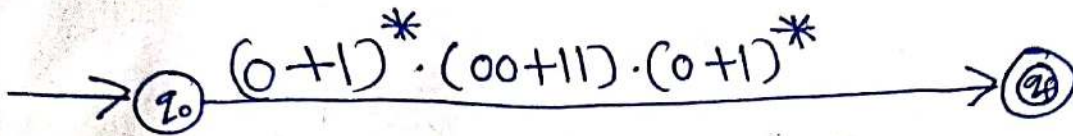


Example 2

Construct the finite automaton equivalent to regular expression

$$(0+1)^* (00+11) (0+1)^*$$

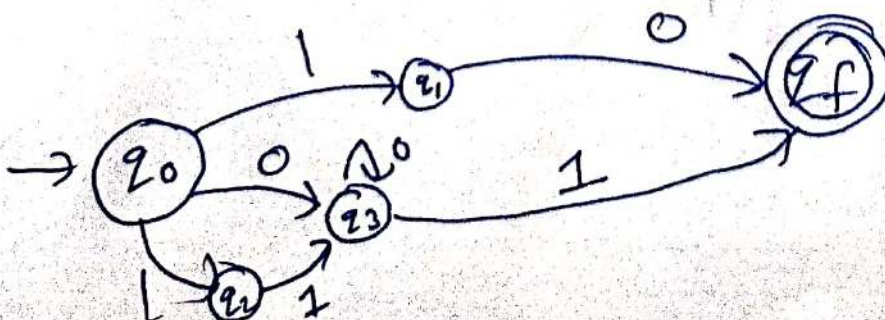
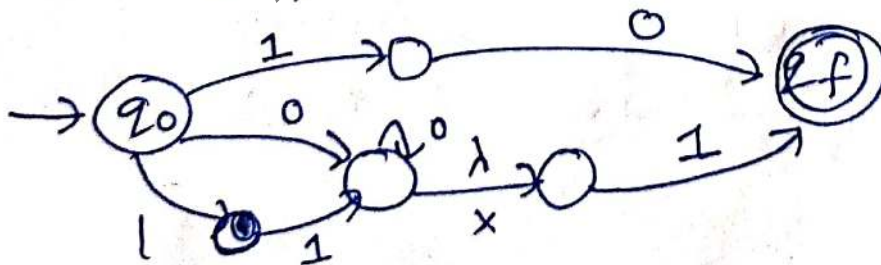
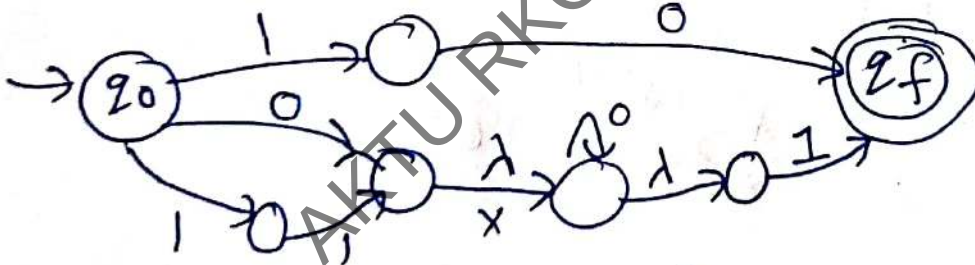
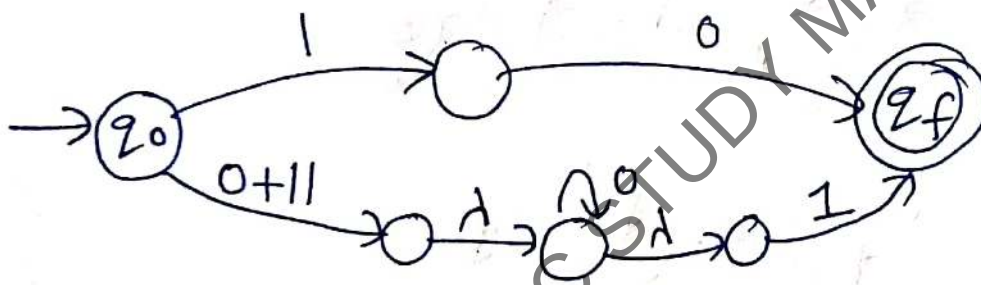
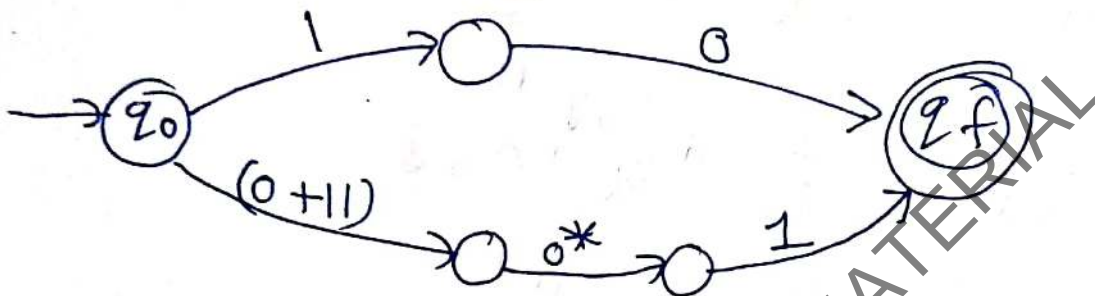
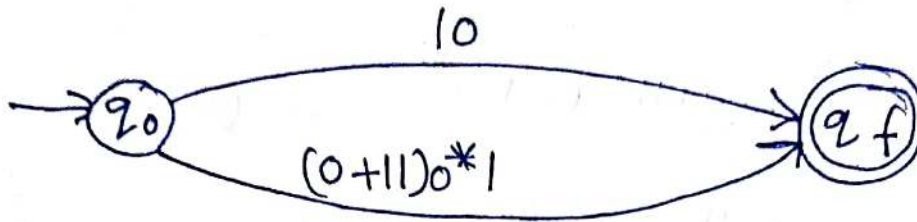
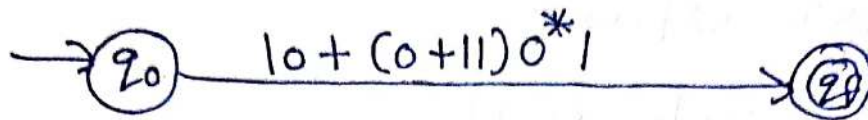
11



Ex:2

Construct DFA for the following regular expression $10 + (0+11)0^*1$

(12)

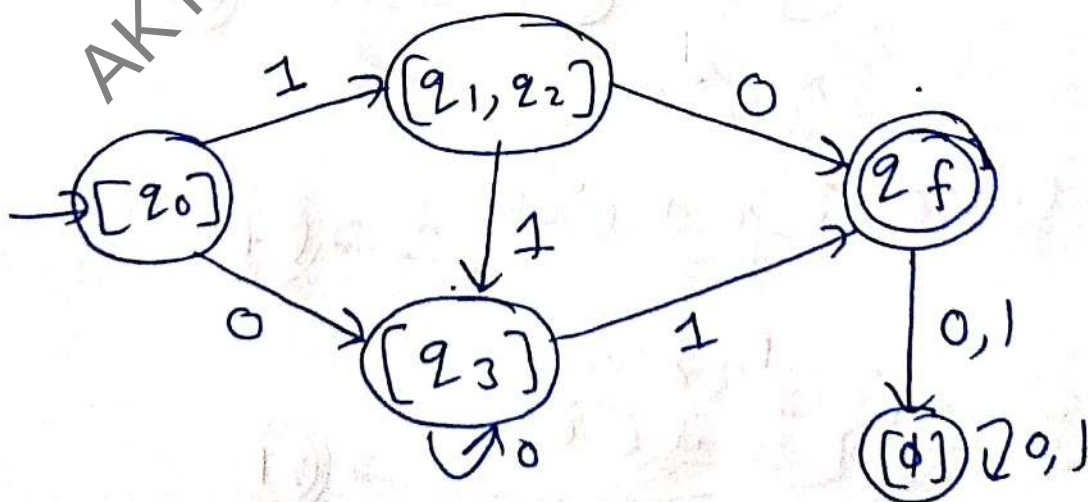


Transition Table for given N DFA

Q \ Σ	0	1
$\rightarrow q_0$	q_3	q_1, q_2
q_1	q_f	
q_2		q_3
q_3	q_3	q_f
(q_f)		

NDFA to DFA

Q \ Σ	0	1
$\rightarrow [q_0]$	$[q_3]$	$[q_1, q_2]$
$[q_3]$	$[q_3]$	$[q_f]$
$[q_1, q_2]$	$[q_f]$	$[q_3]$
$[q_f]$	$[\phi]$	$[\phi]$
$[\phi]$	$[\phi]$	$[\phi]$



DFA

Prove that

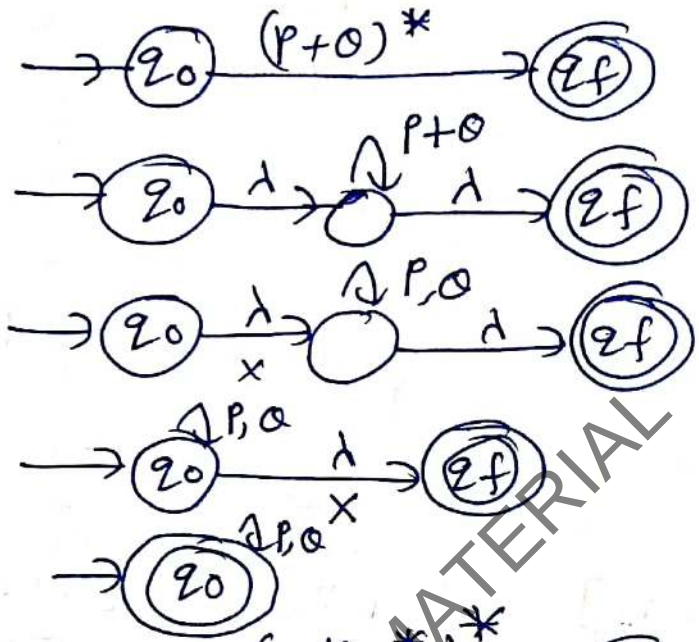
$$(P+Q)^* = (P^*Q^*)^* = (P^*+Q^*)^*$$

(14)

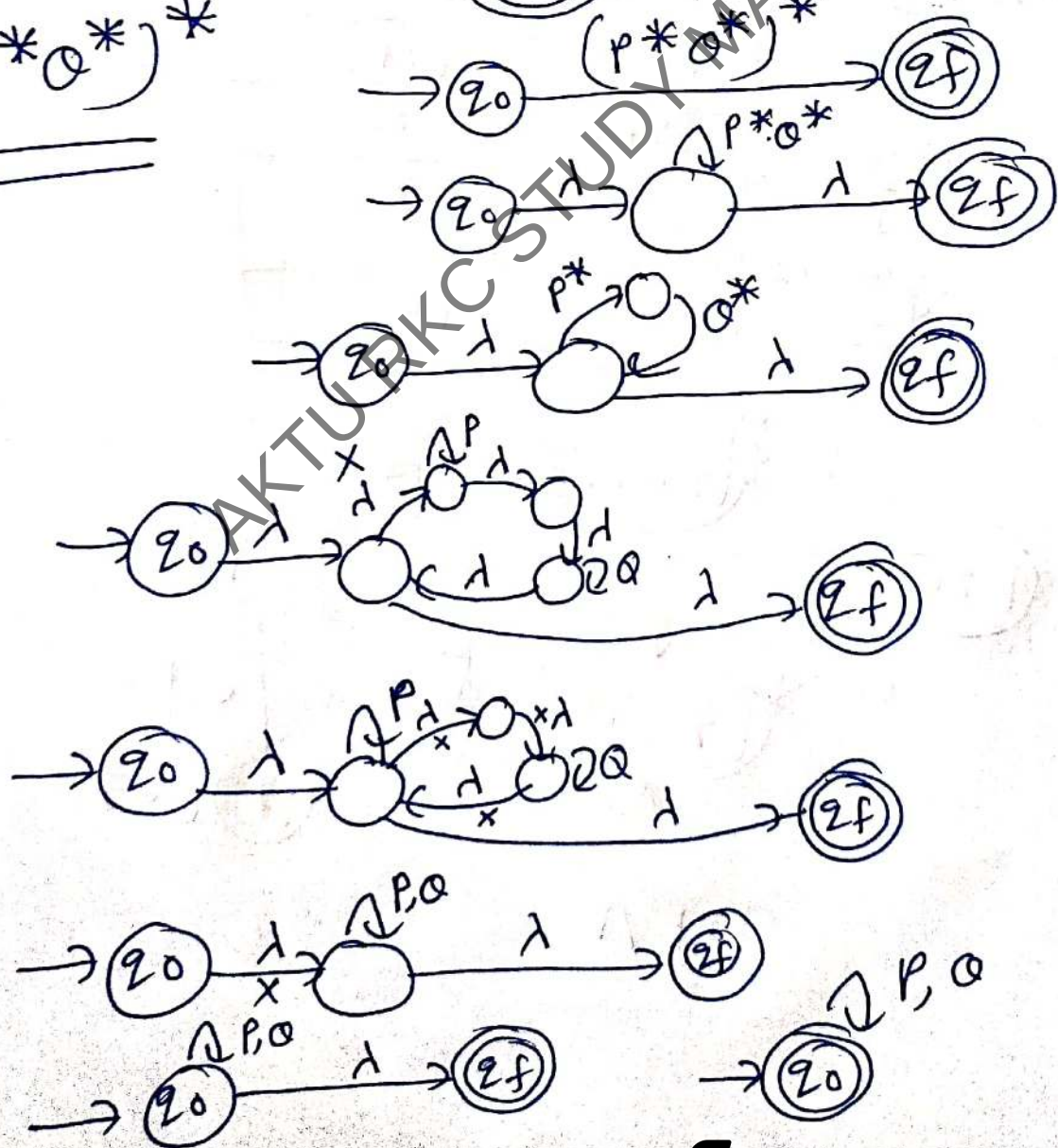
Ans

Draw FA for all three Regular expressions

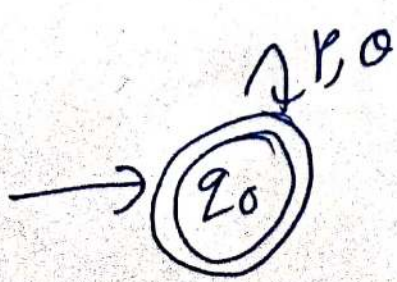
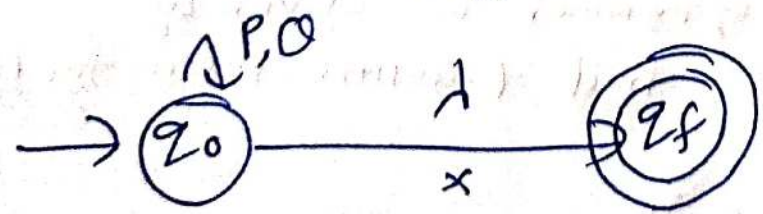
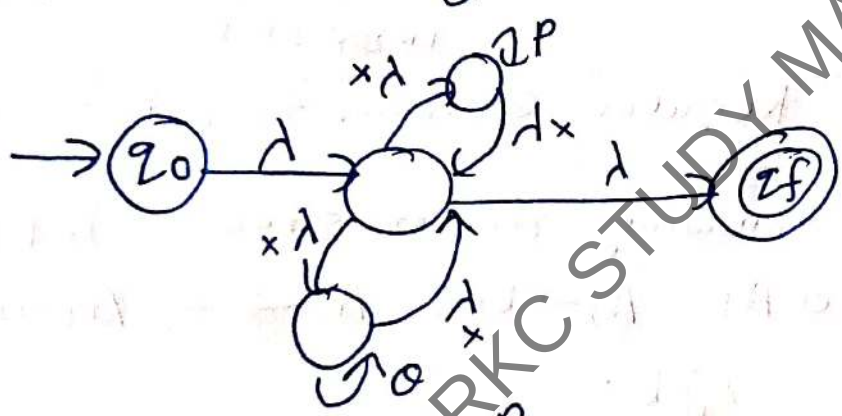
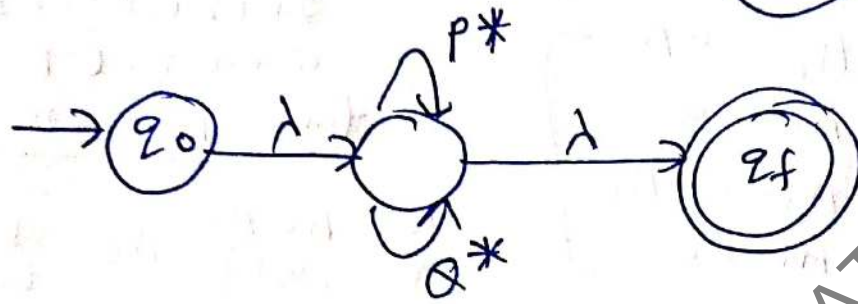
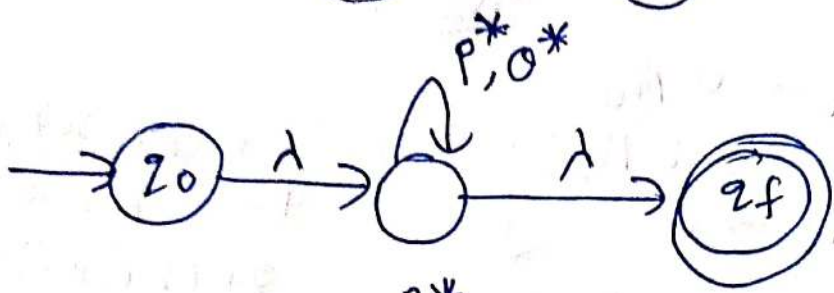
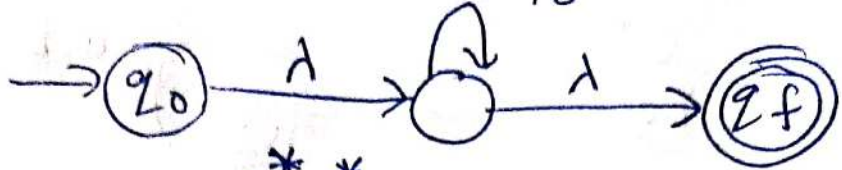
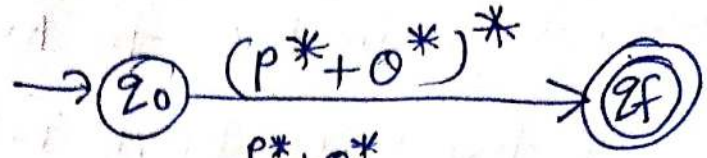
$(P+Q)^*$



$(P^*Q^*)^*$



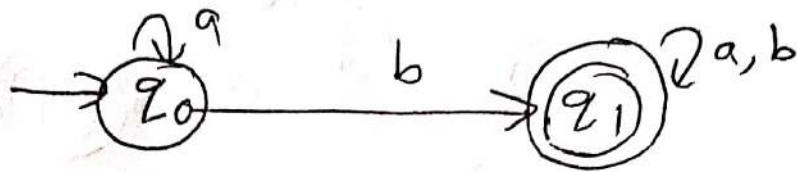
$$(p^* + q^*)^*$$



Conversion from FA to Regular Grammar

(16)

ex:1 Convert the given DFA into Regular Grammar



$$G = \left[\begin{array}{l} A_0 \rightarrow a A_0 \\ A_0 \rightarrow b A_1 \\ A_0 \rightarrow b \\ A_1 \rightarrow a A_1 \\ A_1 \rightarrow a \\ A_1 \rightarrow b A_1 \\ A_1 \rightarrow b \end{array} \right]$$

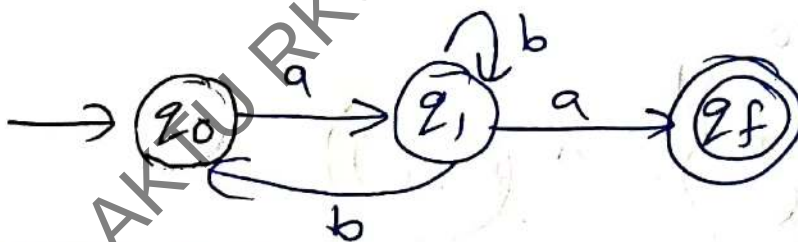
If $\delta(q_i, x) \rightarrow q_j$
 then $A_i \rightarrow x q_j$
 If $\delta(q_i, x) \rightarrow q_j$
 where $q_j \in F$
 then $A_i \rightarrow x$

$N = A_0, A_1, A_0 = \text{start}$
 $T = \{a, b\}, A_0 = \text{variable}$
~~initial state~~

Conversion from Regular Grammar to FA

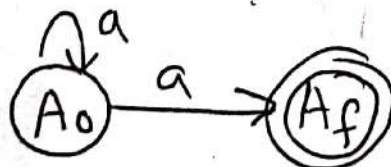
ex:2

Convert the following regular grammar to FA
 $A_0 \rightarrow a A_1, A_1 \rightarrow b A_1, A_1 \rightarrow a, A_1 \rightarrow b A_0$



ex:3

If a regular grammar is given by
 $S \rightarrow aS | a$, find transition system for it.



Numericals on Regular expressions (17)

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

Ans

$$\begin{aligned} \text{L.H.S } & (1+00^*1) + (1+00^*1)(0+10^*1)^*(0+10^*1) \\ \Rightarrow & (1+00^*1) \left[\underbrace{1}_{R^*} + \underbrace{(0+10^*1)^*}_{R} \underbrace{(0+10^*1)}_R \right] \\ \Rightarrow & (1+00^*1)(0+10^*1)^* \\ \Rightarrow & 1 \left[1 + \underbrace{00^*}_{R R^*} \right] (0+10^*1)^* \\ \Rightarrow & 10^*(0+10^*1)^* \\ \Rightarrow & \underline{\underline{0^*1(0+10^*1)^*}} \end{aligned}$$

$[1+R^*R=R^*]$
 from I9
 $[1+RR^*=R^*]$

② Prove that $P+PQ^*Q = a^*bQ^*$, where
 $P = b+aa^*b$ and Q is any regular expression.

Qs L.H.S

$$\begin{aligned} & P+PQ^*Q \\ \Rightarrow & P(1+Q^*Q) \\ \Rightarrow & \underline{P} \cdot Q^* \\ \Rightarrow & (b+aa^*b)Q^* \\ \Rightarrow & b(\underline{1+aa^*})Q^* \\ \Rightarrow & b \cdot a^*Q^* \\ \Rightarrow & \underline{\underline{a^*bQ^*}} \end{aligned}$$

$[1+R^*R=R^*]$
 from I9
 $[1+aa^*=a^*]$
 from I9

Pumping Lemma for Regular Languages (18)

- ① Pumping Lemma for regular languages are used to prove that certain languages are not regular.
- ② In this, from a given string, we create new strings by pumping (generating) a specific part of a string.
- ③ Lemma are parts of a theorem, that helps in the proof of a theorem. They are the stepping stones in the proof of a theorem.
- ④ "If L is a regular language and w is a string of L . If w is divided into 3 parts x, y and z , then it must satisfy the following conditions

① $xyz \in L$ for $i \geq 0$

② $|y| > 0$

③ $|xy| \leq p$ (where p is the pumping length)

example: Prove that $\{L = a^n b^n \mid n \geq 1\}$ is not regular

Ans: Assume $L = a^n b^n$ is regular and $p =$ pumping length
then $w = a^p b^p$, $L = a^n b^n = \{ab, aabb, aaabbb, \dots\}$
let $w = a^3 b^3$ ($p=3$)
 $w = \underline{aaa bbb}$

divide into 3 parts such that $\begin{cases} |y| > 0 \\ |xy| \leq 3 \end{cases}$
since $p=3$ is pumping length

Case 1: $w = \underline{aaa} \underline{bbb}$ $\begin{cases} |y| > 0 \\ |xy| \leq 3 \end{cases}$
 $\begin{cases} x=aa \\ y=aa \\ z=bbb \end{cases}$
 $xy^i z \in L$ for $i \geq 0$
for $i=0$, $xy^0 z = a(aa)^0 bbb = a.1.bbb = \underline{abbb} \notin L$
for $i=1$, $xy^1 z = a(aa) bbb = \underline{aaabbb} \in L$
for $i=2$, $xy^2 z = a(aa)^2 bbb = \underline{aaaaabbb} \notin L$

If rules ~~violates~~ violater for any value of i , then we ~~can~~ can say that Language is not regular

Case 2:

$$w = \frac{aaa}{x} \frac{bbb}{y} \frac{bbb}{z}$$

$$\begin{cases} |xy| > 0 \\ |xyz| \leq 3 \end{cases}$$

(19)

$$xy^iz \in L \text{ for } i \geq 0$$

for $i=0$, $xy^0z \in L$

$$= (aa)(a)^0(bbb) = aa \cdot 1 \cdot bbb = aabbb \notin L$$

for $i=1$

$$xy^1z$$

$$(aa)(a)bbb = aaa \cdot bbb \in L$$

for $i=2$

$$xy^2z$$

$$(aa)(a)^2bbb = aaaa \cdot bbb \notin L$$

for $i=3$

$$xy^3z$$

$$(aa)(a)^3bbb = aaaaa \cdot bbb \notin L$$

for $i=4$

$$xy^4z$$

$$(aa)(a)^4bbb = aaaaaa \cdot bbb \in L$$

only for $i=1$, it belongs to regular language.

Since $w = a^4b^4$

$$\boxed{\text{for } b=4}$$

$$w = a^4b^4$$

~~$$w = aaaaa \cdot bbb$$~~

let $x=a$

$$w = a^4b^4$$

$$w = \frac{aaaa}{x} \frac{bbb}{y} \frac{bbb}{z}$$

$$\begin{cases} |xy| > 0 \\ |xyz| \leq 3 \end{cases}$$

$$xy^iz \in L \text{ for } i \geq 0$$

$$\begin{cases} x=a \\ y=aa \\ z=abbbb \end{cases}$$

for $i=0$

$$xy^0z = a(aa)^0abbbb = aabbbb \notin L$$

for $i=1$

$$xy^1z = a(aa)^1abbbb = aaaaa \cdot bbb \in L$$

for $i=2$

$$xy^2z = a(aa)^2abbbb = aaaaaa \cdot bbb \notin L$$

Since some of the strings does not belongs to language L , So $\{L = a^nb^4 | n \geq 1\}$ is not regular

Ex: 2

Show that $\{L = a^p \mid p \text{ is a prime no.}\}$ is not regular

(20)

Ans

$L = a^p \mid p \text{ is a prime no.}$

$= \{aa, aaa, aaaaa, aaaaaaa, \dots\}$
since $p = 2, 3, 5, 7, 11, 13, \dots$
are prime no.

Using pumping lemma

① $xy^iz \in L$ for $i \geq 0$

② $|y| > 0$

③ $|xy| \leq p$ (where p is the pumping length)

Case 1:

Let $w = a^p$ ($p=3$)
 $w = a^3$
 $w = \frac{aaa}{x \ y \ z}$

$\begin{cases} |y| > 0 \\ |xy| \leq p \\ p \leq 3 \end{cases}$

$\begin{cases} x=a \\ y=a \\ z=a \end{cases}$

divide into 3 parts
for $i=0$, $xy^iz = xy^0z = a \cdot 1 \cdot a = aa \notin L$

for $i=1$, $xy^iz = xy^1z = a(a)1a = aaa \in L$

for $i=2$, $xy^iz = xy^2z = a(a^2)a = aaaaa \notin L$

for $i=3$, $xy^iz = xy^3z = a(a^3)a = aaaaaaa \in L$

for $i=4$, $xy^iz = xy^4z = a(a^4)a = aaaaaaaaa \notin L$

Since some of the values are not belonging to Language, so for Case 1, it is not regular

~~Case 2:~~

~~Let $w = a^p$ ($p=5$)
 $w = a^5$
 $w = \frac{aaaaa}{x \ y \ z}$~~

$w = \frac{aaaa}{x \ y \ z} = a^3 = a^2$

Let $x=aa, y=a, z=\epsilon$

$\begin{cases} |y| > 0 \\ |xy| \leq p \end{cases}$

Case 2:

$$L = xy^iz$$

(21)

$$\underline{i=0}, xy^0z = aa(a)^0 \cdot \epsilon \\ = aa \cdot 1 \cdot \epsilon = aa \in L$$

$$i=1, xy^1z = aa(a)^1 \cdot \epsilon \\ = aaa \cdot \epsilon = aaa \in L$$

$$\underline{i=2}, xy^2z = aa(a)^2 \cdot \epsilon \\ = aaaa \notin L$$

$$\underline{i=3}, xy^3z = aa(a)^3 \cdot \epsilon \\ = aaaaaa \in L$$

for, $i=2$, string does not belong to language.

Since we got some strings that does not belong to language L , so L is not regular
i.e. $\{L = a^p \mid p \text{ is a prime no.}\}$ is not regular

Question for Practice

(a) $L = \{a^n b^{2n} \mid n \geq 0\}$

(b) $L = \{a^n b^m \mid 0 \leq n \leq m\}$

(c) $L = \{a^{2n} \mid n \geq 1\}$

Using pumping Lemma, prove that following languages are not regular.

More examples on Regular grammar to

DFA

ex: Construct a DFA equivalent to the grammar

$$S \rightarrow aS | bS | aA$$

$$A \rightarrow bB$$

$$B \rightarrow aC$$

$$C \rightarrow \lambda$$

Q9

First remove λ by placing the value of $C \rightarrow \lambda$ in $B \rightarrow aC$,

the modified grammar is

$$S \rightarrow aS$$

$$S \rightarrow bS$$

$$S \rightarrow aA$$

$$A \rightarrow bB$$

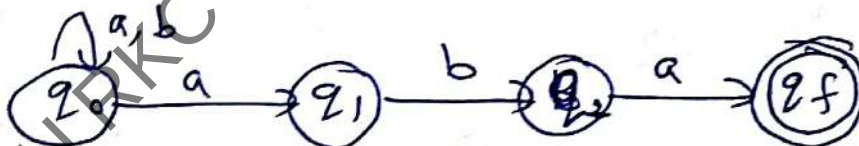
$$B \rightarrow a$$

(putting $C \rightarrow \lambda$ in $B \rightarrow aC$)

$G =$

So transition system (NFA) is

Let q_0, q_1, q_2 are states corresponding to S, A, B non-terminals



NFA table

Q \ ε	a	b
→ q ₀	q ₀ , q ₁	q ₀
q ₁	-	q ₂
q ₂	q _f	-
(q _f)	-	-

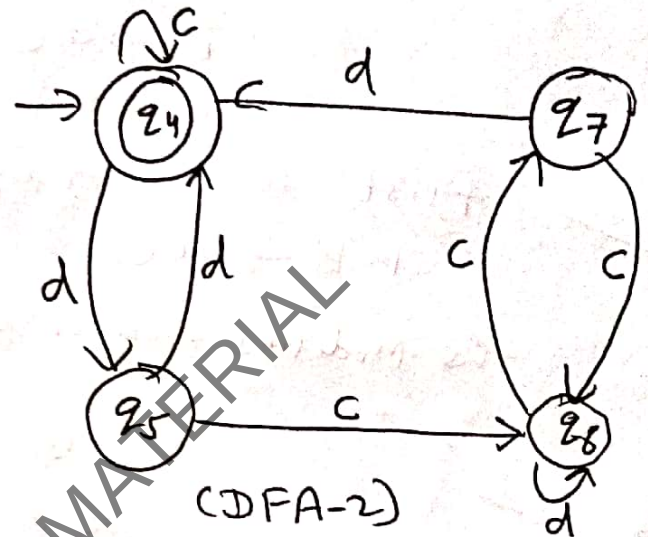
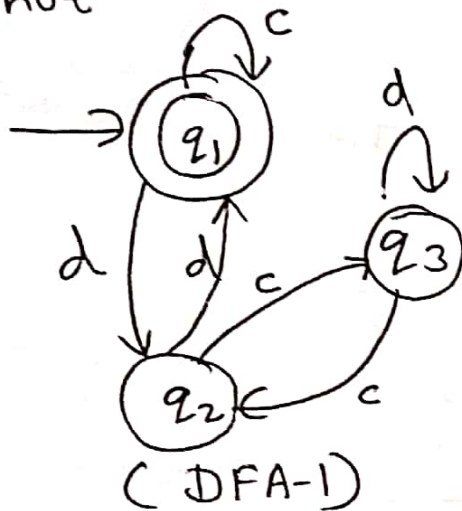
NDFA to DFA

Q \ ε	a	b
→ [q ₀]	[q ₀ , q ₁]	[q ₀]
[q ₀ , q ₁]	[q ₀ , q ₁]	[q ₀ , q ₂]
[q ₀ , q ₂]	[q ₀ , q ₁ , q _f]	[q ₀]
[q ₀ , q ₁ , q _f]	[q ₀ , q ₁]	[q ₀ , q ₂]

Equivalence of two Finite Automata using 23

Comparison Method

ex: Consider the following two DFA shown in the given figure. Using Comparison method, check whether the two DFA are equivalent or not.



Ans Comparison Table Use starting state of each DFA to initiate table.

inputs \ states	c	d
(q1, q4)	(q1, q4)	(q2, q5)
(q2, q5)	(q3, q6)	(q1, q4)
(q3, q6)	(q2, q7)	(q3, q6)
(q2, q7)	(q3, q6)	(q1, q4)

As we did not get a single pair of states, in which for a given input (c or d), one next state is final & other is non-final (or vice versa) at every row.

So both DFA are equivalent

Right Linear Grammar

(24)

A grammar is said to be right linear grammar, if the productions are of the form

$$A \rightarrow xB$$

$$A \rightarrow x$$

Where $A, B \rightarrow$ Non terminal

$x \rightarrow$ terminal including null string

Left Linear Grammar:

A grammar is said to be left linear grammar, if the productions are of the form

$$A \rightarrow Bx$$

$$A \rightarrow x$$

Where $A, B \rightarrow$ Non terminal

$x \rightarrow$ Terminal including null string.

Regular Grammar:

A regular grammar is either right linear grammar or left linear grammar

Closure properties of regular languages (25)

① If L_1 and L_2 are regular languages, then $L_1 \cup L_2$, $L_1 \cdot L_2$ and L_1^* are also regular languages.

Proof

If L_1 and L_2 are regular languages, then there exist regular expressions γ_1 and γ_2 such that $L_1 = L(\gamma_1)$

and

$$L_2 = L(\gamma_2)$$

As by definition we know that

(i) $\gamma_1 + \gamma_2$ is regular expression, so correspondingly

$L_1 \cup L_2$ is also regular language

(ii) $\gamma_1 \cdot \gamma_2$ is regular expression, so correspondingly

$L_1 \cdot L_2$ is also regular language.

(iii) γ_1^* is regular expression, so correspondingly

L_1^* is also regular language.

② If L is a regular language, then its complement $\bar{L}$ is also regular language.

Proof

Since L is a regular language, so there exist a DFA for it.

Let



Let's name it M

$M = \text{DFA for regular language } L$

Now if we complement M , then all non-final states will become final states and all final states will be non-final states.



Since $\bar{L}$ is still DPA, So its Language $\bar{L}$ is also regular language.

So we can say that if L is a regular language, then its Complement $\bar{L}$ is also regular language.

③ If L_1 and L_2 are two regular languages, then $L_1 \cap L_2$ is also regular.

Proof Using De-Morgan's law, we know that,

$$L_1 \cap L_2 = \overline{(\bar{L}_1 \cup \bar{L}_2)} \quad \text{--- ①}$$

from R.H.S

$$\begin{aligned} \text{(Since } A \cap B &= \overline{(\bar{A} \cup \bar{B})}) \\ &= \overline{(\bar{A} + \bar{B})} \end{aligned}$$

Since L_1 is regular language,
So its Complement $\bar{L}_1$ is also regular language
Since L_2 is regular language,
So its Complement $\bar{L}_2$ is also regular language
Since $\bar{L}_1$ and $\bar{L}_2$ are regular languages, so
 $\bar{L}_1 + \bar{L}_2$ is also regular language.
as we know that the Complement of
a regular language is also regular language,
so $\overline{(\bar{L}_1 + \bar{L}_2)}$ is also regular

from eqn ① R.H.S = L.H.S
i.e. $L_1 \cap L_2$ is also regular.

④ If L_1 and L_2 are regular languages, then $L_1 - L_2$ is also regular.

Proof from set theory, we know that

$$A - B = A \cap \bar{B} \text{ so } L_1 - L_2 = L_1 \cap \bar{L}_2$$

We also know that if L_2 is regular, then $\bar{L}_2$ is also regular.

Since L_1 and $\bar{L}_2$ are both regular, so their Intersection $L_1 \cap \bar{L}_2$ is also regular language.

Homomorphism and inverse homomorphism in regular language!

Homomorphism(h): A homomorphism is a substitution in which a single letter is replaced with a string.

Let Σ and τ are two alphabets

$$h: \Sigma \rightarrow \tau^*$$

$$\text{Let } \Sigma = \{0,1\}, \tau = \{a,b\}$$

$$h(0) = aa, h(1) = bb$$

If L is a regular language

$$L = \{00, 101\}$$

$$\begin{aligned} \text{then } h(L) &= \{h(00), h(101)\} \\ &= \{aaaa, bbaabb\} \end{aligned}$$

If γ is a regular expression

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

$$h(\gamma) = (aa+bb)^* (bb)^*$$

Inverse homomorphism (h^{-1})

The inverse process of homomorphism is called inverse homomorphism. It is represented by h^{-1} .

$$\text{Let } \Sigma = \{0, 1, 2\}, \quad \tau = \{a, b\}$$

$$h(0) = a, \quad h(1) = b, \quad h(2) = ab$$

Given regular language

$$L = \{ \underline{a} \underline{b} \underline{a} \underline{b} \}$$

Case-1

$$h^{-1}(L) = \{ 22 \}$$

Case-2

$$L = \{ \underline{a} \underline{b} \underline{a} \underline{b} \}$$

$$h^{-1}(L) = \{ 0101 \}$$

Case-3

$$L = \{ \underline{a} \underline{b} \underline{a} \underline{b} \}$$

$$h^{-1}(L) = \{ 201 \}$$

Case-4

$$L = \{ \underline{a} \underline{b} \underline{a} \underline{b} \}$$

$$h^{-1}(L) = \{ 012 \}$$

$$\text{So } h^{-1}(L) = \{ 22, 0101, 201, 012 \}$$

$$\text{Let } \Sigma = \{0, 1\}, \quad \tau = \{a, b\}$$

$$h(0) = aa, \quad h(1) = bb$$

$$\text{and } h(r) = (aa+bb)^* (bb)^*$$

$$\text{then } h^{-1}(h(r)) = (0+1)^* 1^*$$

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

Describe the decision properties of regular languages:

There are three decision properties of regular languages:

- ① Membership property
- ② Emptiness property
- ③ Equivalence property
- ④ Finiteness Property

① Membership property:

This property specifies that whether a particular string w belongs to some language (Regular language).

Proof Let L is a regular language.

i.e. there exist a Finite Automata (DFA/NFA) for it. If there exist a string w then if w reaches to an accept state (final state), ~~after~~ then we can say that string w belongs to regular language.

② Emptiness property:

Proof The emptiness property specifies that whether a regular language is empty or not.

If L is a regular language then there exist a FA (DFA/NFA) for it.

If there exist some path from start state to final state in FA, then the regular language is not empty.

However, if there is no path reaching to a final state from the start state, then the language is empty.

③ Equivalence property:

Equivalence property specify that whether two regular language are equivalent.

Proof

Since we know that any regular language can be specified by FA (DFA/NDFA).
So if there are two regular language L_1 and L_2 , and they can be represented by DFA₁ and DFA₂. Then by using Comparison method, we can say that whether DFA₁ and DFA₂ are equivalent or not. If $DFA_1 = DFA_2$ then, these regular language are also equivalent.

④ Finiteness Property:

Finiteness property specify that whether the strings generated by regular languages are finite or not.

Proof

As we know that any regular language can be represented using DFA/NDFA.

- (i) So for the DFA drawn from the regular language, check the states that can not be reached from the initial states & delete them. This is called Elimination of unreachable states.
- (ii) Check the states from which we can not reach the final states & delete them. This is called elimination of dead states.
- (iii) If the resulting DFA contains loops then the finite automaton accept infinite strings of a regular language.
- (iv) If the resulting DFA does not contain loops, then the finite Automaton accept finite strings of a language.

Some other closure properties of regular languages

If L is a regular language and h is a homomorphism on its alphabet, then $h(L)$ is also a regular language.

Proof If L is a regular language then there exist a regular expression R for it. When we will apply homomorphism h on each symbol of regular expression R , then still we got another regular expression after homomorphism. i.e. the $h(L)$ or homomorphism of a regular language is also a regular language.

If h^{-1} is a homomorphism and L is a regular language, then on its alphabet, then $h^{-1}(L)$ is also a regular language.

Proof If L is a regular language and $h(L)$ is also regular language due to closure property, i.e. for $h(L)$, there exist a ~~regular~~ regular expression $h(r)$. If we take the inverse of regular expression $[h^{-1}(h(r))]$, we still got regular expression r . So $h^{-1}(L)$ is also a regular language.