Date:
Page No.:

Theory of Computation.

language → set of strings.

→ We can't design ~~a~~ machine which can detect infinite loops (e.g. It's not possible to design a machine which Accepts all valid Java codes as well as never goes in infinite loops).

Undecidable.

Turning Machine.

Context Free language

Finite state Machine

[TOC_02]

Symbol : a, b, c ... 0, 1, 2, 3 ...

Alphabet (Σ) : collection of symbols : {a, b}, {d, e, f}

string : sequence of symbols : a, b, 0, 1, aa, bb, ab

language : set of strings

eg $\Sigma = \{0, 1\}$

Finite $\begin{cases} L_1 : \text{set of all strings of length 2} \\ \quad = \{00, 01, 10, 11\} \\ L_2 : \text{——— length 3} \end{cases}$

{000, 001, 010, 011, 100 .... 111}

Infinite { L_3 : set of all strings that begin with 0.

Powers of Σ : eg $\Sigma = \{0, 1\}$

Σ^0 : set of all strings of length 0 : $\Sigma^0 = \{\epsilon\}$

Σ^1 : ——— 1 : $\Sigma^1 = \{0, 1\}$

Σ^2 : ——— 2

.

.

.

Σ^n : set of all strings of length n

Scanned with CamScanner

Date:
Page No.:

Cardinality : numbers of elements in set.

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \ldots \Sigma^n$

$= \{\epsilon\} \cup \{0,1\} \cup \{00, 01, 10, 11\} \cup \ldots$

= Set of all possible strings of all lengths over $\{0,1\}$

↳ infinite set

TOC_03

Finite state Machine

Finite Automata (FA)

- FA with output
  - Moore
  - Mealy
- Without output
  - DFA
  - NFA
  - ε-NFA

FSM : (i) simplest model of computation
(ii) limited memory.

DFA (Deterministic —)

$DFA = \{Q, \Sigma, q_0, F, \delta\}$

$Q \rightarrow$ set of all state

$\Sigma =$ input

$q_0 \rightarrow$ start state / initial state

$F \rightarrow$ set of final states (can be more than one)

$\delta \rightarrow$ Transition function from $Q \times \Sigma \rightarrow Q$

Q (states) \ Σ (Input →)	0	1
A	C	B
B	D	A
C	A	D
D	B	C

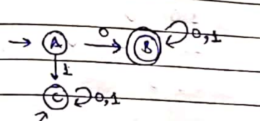
← Transition state.

Date:
Page No.:

TOC_04

L1 : Set of all strings that start with '0'

$\{0, 00, 01, 000, 001, 010 \ldots \infty\}$



Trap / Dead state

Eg :- 001

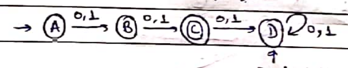
$\rightarrow A \xrightarrow{0} B \xrightarrow{0} B \xrightarrow{1} B \rightarrow$ Final state

TOC_05

L2 : set of all strings over $\{0,1\}$ of length 2.

$\Sigma : \{0,1\}$

L2 : $\{0,0,01,10,11\}$



Dead state

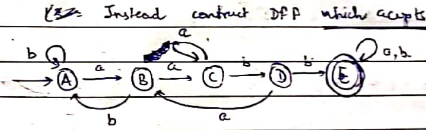
length	0	1	2	3/4/5 ...

TOC_06

L3 : accept any sequence of $\{a,b\}$ but doesn't contain aabb in

$\Sigma = \{a,b\}$

Instead construct DFA which accepts aabb

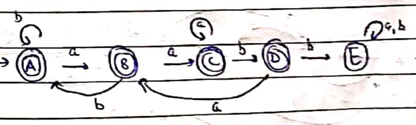


→ Flip the states.

↳ make final → non final

nf → final state

L3 :

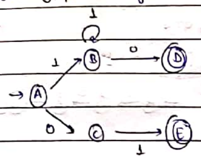


Date:
Page No.:

TOC_07

What DFA recognizes.

Ex:



10 ✓

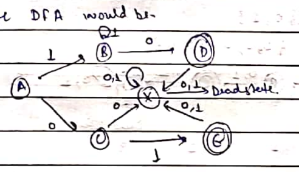
11 . . . . 1, 0 ✓

01 ✓

L : { Accepts the string 01 or string of atleast one '1' followed by a '0' }

strings like 001, 010, 011, 1101, 1100 goes to dead state.

So complete DFA would be.



Class.
8/01/2021

$L : \{0^n 1^n \mid n > 1\} \Rightarrow 0^3 1^3 = 000111$

$L^2 = \{0^n 1^n 0^m 1^m \mid n, m \geq 0\} \qquad \neq 01001$

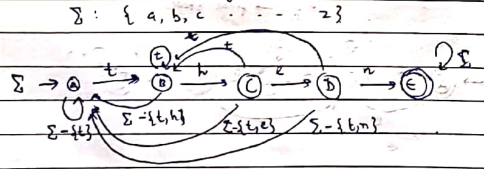
$L^R =$ Reverse of L $\{1^n 0^n \mid n \geq 0\}$

$\hat{\delta}(q, \epsilon) = q \qquad \hat{\delta}(q, w) = \delta(\hat{\delta}(q, x), a) = p$

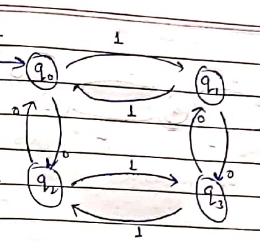
$= \delta(s, a) = p$

Ex: DFA to recognize 'then'

$\Sigma : \{a, b, c \ldots z\}$



Even 1
Even 0



If final state is q_0 Machine (DFA) recognizes Even 1, Even 0

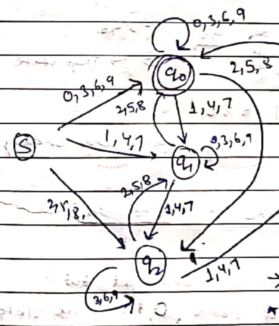
$q_1 \rightarrow$ odd 1, Even 0

$q_2 \rightarrow$ Even 1, odd 0

$q_3 \rightarrow$ odd 1, odd 0.

11/01/2020
class.

Divisibility by 3.



$Q = \{S, q_0, q_1, q_2\}$

$F = \{q_0\}$

$S = S$

$M = \{Q, \Sigma, \delta, S, F\}$

δ	0,3,6,9	1,4,7	2,5,8
$\rightarrow S$	q_0	q_1	q_2
$* q_0$	q_0	q_1	q_2
q_1	q_1	q_2	q_0
q_2	q_2	q_0	q_1

W = 5678923124б

→ $\hat{\delta}(S, 5) = q_2$

$\hat{\delta}(S, 56) = q_2$

$\hat{\delta}(567) = q_0$

$\hat{\delta}(S, 5678) = q_2$

~~$\hat{\delta}(S, 56789) = q_2$~~

$\hat{\delta}(S, 567892) = q_1$

$\hat{\delta}(S, 5678923) = q_1$

$\hat{\delta}(S, 56789231) = q_2$

$\hat{\delta}(S, \ldots 312) = q_1$

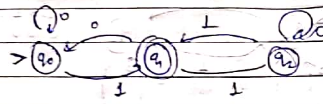
$\hat{\delta}(S, \ldots 3121) = q_1$

$\hat{\delta}(S, \ldots 31246) = q_1$

Scanned with CamScanner

Date:
Page No.:

Example:



Find the language of machine.

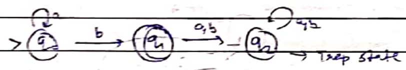
It comes to q_1 when
- any zero (1)
- any No. of zero (1) (any zero) 1
- 1 (any no of zero) 1

so → we get q_1 when last symbol is 1 from q_0

L: consecutive odd no of 1's in the end.

eg 1101, 110111, 111

Example:

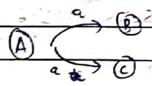


$\Rightarrow L = \{a^n b \mid n \geq 0\}$

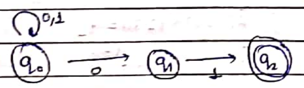
All strings having one b and ending in b.

* **NFA (Non-deterministic Finite Automata)**

δ: that takes some state and input but it returns a set of states {0, 1 or more states}



Eg:

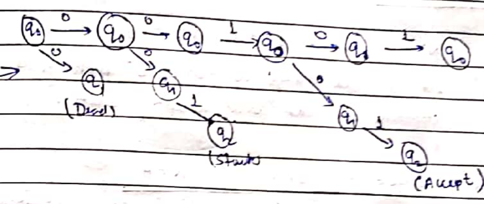


Accept all string ending in 0, 1,
... ... — 01

12/01/2024

Date:
Page No.:

Assume w = 00101



NFA = { Q, Σ, S, F, δ } $\delta = Q \times \Sigma \to 2^{Q}$ (Power set of Q)

Q = { q_0, q_1, q_2
Σ = {0, 1}
S = q_0
F = { q_2 }

δ	0	1
→ q_0	$\{q_0, q_1\}$	$\{q_0\}$
q_1	φ	$\{q_2\}$
* q_2	φ	{φ}

$\hat{\delta}$ for NFA:

Basis $\hat{\delta}(q, \epsilon) = \{q\}$

let $w = xa$ $a \in \Sigma$

$\hat{\delta}(q, x) = \{p_1, p_2 \ldots p_n\}$ → set of states.

$$\bigcup_{i=1}^{k} \delta(p_i, a) = \delta(p_1, a) \cup \delta(p_2, a) \ldots$$

$$= \{r_1, r_2 \ldots r_m\}$$

$$\hat{\delta}(q, w) = \{r_1, r_2, r_3 \ldots r_m\}$$

Example: w = 00101

δ	0	1
q_0	$\{q_0 q_1\}$	$\{q_0\}$
q_1	φ	$\{q_2\}$
q_2	φ	φ

can be verified

$\hat{\delta}(q_0, \epsilon) = \{q_0\}$

$\hat{\delta}(q_0, 0) = \delta(q_0, 0) = \{q_0, q_1\}$

$\hat{\delta}(q_0, 00) = \delta(q_0, 0) \cup \delta(q_1, 0)$
$= \{q_0, q_1\} \cup \phi = \{q_0, q_1\}$.

$\hat{\delta}(q_0, 001) = \delta(q_0, 1) \cup \delta(q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

$\hat{\delta}(q_0, 0010) = \delta(q_0, 0) \cup (q_2, 0) = \{q_0, q_1\} \cup \phi = \{q_0, q_1\}$

$\hat{\delta}(q_0, 00101) = \delta(q_0, 1) \cup (q_1, 1) = \{q_0\} \cup \{q_2\} = \{q_0, q_2\}$

Scanned with CamScanner

Date:
Page No.:

Language of NFA:

$L(N) = \{ w \mid \hat{\delta}(q_0, w) \cap F \neq \phi \}$

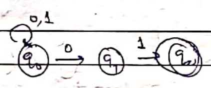
set of all strings w from Σ^* such that $\hat{\delta}(q_0, w)$ contains at least one accepting state.

EQUIVALENCE OF DFA and NFA

NFA $N = (Q_N, \Sigma, \delta_N, q_0, F_N)$ then construct a DFA

$D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ st $L(D) = L(N)$

- → Σ is same for both m/c.
- → starting state for DFA is $\{q_0\}$
- → Q_D is a subset of Q_N, for n states in NFA Q_D 2^n
- → then F_D is set of subsets S of N such that $S \cap F_N \neq \phi$
  so F_D is set of all subsets containing an state of F_N
- → For each set $S \subseteq Q_N$ and $a \in \Sigma$ $\delta_D(S, a) = \bigcup_{p \in S} \delta_N(p, a)$



δ_D	0	1
ϕ	ϕ	ϕ
$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	ϕ	$\{q_2\}$
$\{q_2\}$	ϕ	ϕ
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1, q_2\}$	ϕ	$\{q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

→ Table of NFA

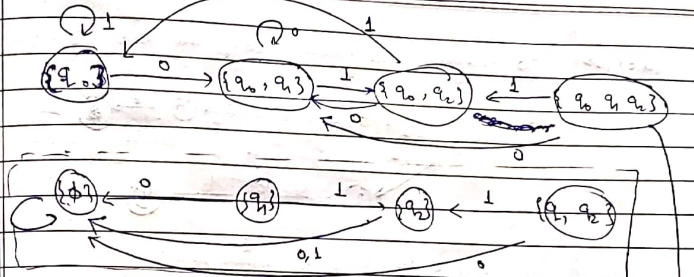
Or we take union of

→ $\delta(q_0, 0) \cup \delta(q_1, 0)$

$\{q_0, q_1\} \cup \phi$

$\Rightarrow \{q_0, q_1\}$

Date:
Page No.:

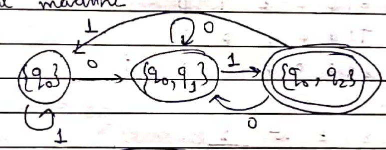


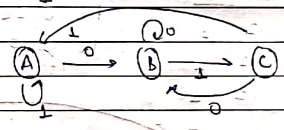
Jaha bhi q_2, is the final state.

Since machine starts from q_0, machine can never reach these state

Also no ~~out~~ incoming arcs

So final machine





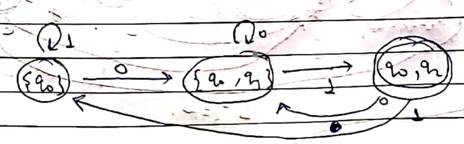
$A = \{q_0\}$

$B = \{q_0, q_1\}$

$C = \{q_0, q_2\}$

This was I method, we explored all states and then reduced.

Other way is start from starting state



$q_0(0) \rightarrow q_0 q_1$

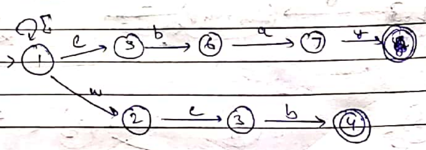
$q_2(0) = \phi$

$q_0(1) = q_0$

$q_2(1)$ ϕ

Date:
Page No.:

14.01.2021 To recognize the tokens 'ebay' and 'web', $\Sigma = \{a \ldots z\}$



δ	e	w	b	a	y	Σ - remaining symbols
1	1,5	12	1	1	1	1
2	3	ϕ	ϕ	ϕ	ϕ	ϕ
3	ϕ	ϕ	4	ϕ	ϕ	
4	ϕ	ϕ	ϕ	ϕ	ϕ	
5	ϕ	ϕ	6	ϕ	ϕ	
6	ϕ	ϕ	ϕ	7	ϕ	
7	ϕ	ϕ	ϕ	ϕ	8	
8	ϕ	ϕ	ϕ	ϕ	ϕ	ϕ

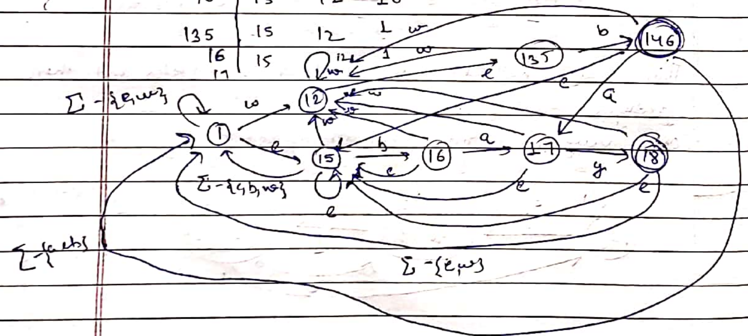
DFA:

δ	e	w	b
1	15	12	1
12	135	12	1
15	15	12	16
135	15	12	1
16	15		
17			

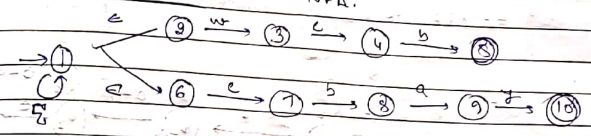
1e ∪ 2e

15 ∪ 3

135



NFA with ∈-Moves ∈-NFA.



$\in$-NFA $= \{Q, \Sigma, \delta, q_0, F)$ $\Sigma = \{a \ldots z\}$

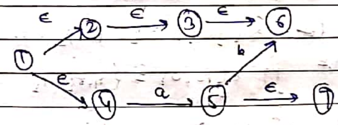
$F = \{5, 10\}$

$\delta = Q \times \{\Sigma \cup \in\} \rightarrow 2^Q$ (Power set of Q)

δ	∈	w	e	b	a	y
1	2,6	1	1	1	1	1
2		3				
3			4			

15/1/21 $\in$-closure:

1) A state q is in $\in$-closure (q)
2) if $p \in$ $\in$-closure then $\in$-closure (q) contains all elements in $\delta(p, \in)$



$\in$ closure (7) = {7}

— (6) = {6}

— (5) = {5, 7}

— (4) = {4}

— (3) = {3, 6}

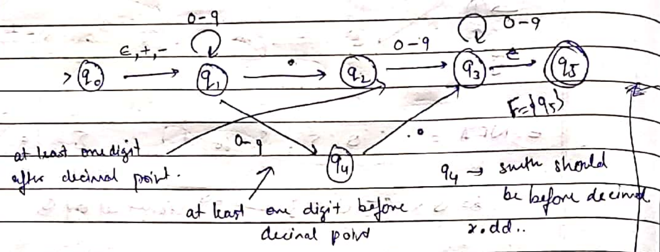
(2) = {2, 3, 6}

(1) = {1, 2, 3, 6}

Scanned with CamScanner

Date:
Page No.:

ϵ- NFA to recognize number in input.:



	δ	ϵ	+,-	.	0-9
>	q_0	$\{q_1\}$	$\{q_1\}$	ϕ	ϕ
	q_1	ϕ	ϕ	q_2	$\{q_3, q_4\}$
	q_2	ϕ	ϕ	ϕ	$\{q_3\}$
	q_3	$\{q_5\}$	ϕ	ϕ	$\{q_3\}$
	q_4	ϕ	ϕ	$\{q_3\}$	ϕ
*	q_5	ϕ	ϕ	ϕ	ϕ

Defination of $\hat{\delta}$ for ϵ-NFA

1. $\hat{\delta}(q, \epsilon) = \epsilon$ closure of q
2. Let $w = xa$ $a \in \Sigma$

(a) Let $\hat{\delta}(q, x) = \{p_1, p_2 \ldots p_k\}$

till NFA → (b) Let $\bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, r_3 \ldots r_m\}$

additional step for ϵ-NFA ← (c) then $\hat{\delta}(q, w) = \bigcup_{j=1}^{m} \epsilon \text{ closure}(r_j)$

Date:
Page No.:

Example Input = 5.6 $\hat{\delta}(q_0, 5.6) = ?$

1. Since $\hat{\delta}(q_0, \epsilon) = \epsilon$-closure of $q_0 = \{q_0, q_1\}$

refer

2. $\hat{\delta}(q_0, 5) \overset{\epsilon\text{-closure}}{=} (\delta(q_0, 5) \cup \delta(q_1, 5))$

$= \{q_1, q_4\}$

closure of $\{q_1, q_4\} = \{q_1, q_4\}$

State	ϵ closure
q_0	$\{q_0, q_1\}$
q_1	$\{q_1\}$
q_2	$\{q_2\}$
q_3	$\{q_3, q_5\}$
q_4	$\{q_4\}$
q_5	$\{q_5\}$

$\delta(q_0, 5.) = \{q_2, q_3, q_5\}$

$\delta(q_0, 5.6) \overset{\epsilon\text{-closure}}{=} (\delta(q_2, 6) \cup \delta(q_3, 6) \cup \delta(q_5, 6))$

$= \{q_3\} \cup \{q_3\} \cup \phi$

$= \epsilon\text{-closure } \{q_3\} = \{q_3, q_5\}$

↓ final state.

Hence 5.6 is accepted by machine.

<u>Language of ϵ-NFA</u>

$E = \{Q, \Sigma, \delta, q_0, F\}$

$L(E) = \{w \mid \hat{\delta}(q_0, w) \cap F \neq \phi\}$

<u>Eliminating ϵ- Transitions:</u> Machine to equivalent DFA

ϵ- NFA $= E = (Q_E, \Sigma, \delta_E, q_0, F_E)$

$= D = (Q_D, \Sigma, \delta_D, q_0, F_D)$

1. Q_D is set of subsets of Q_E
2. $q_D = \epsilon$ - closure (q_0)
3. $F_D = \{S \mid S$ is in Q_D and $S \cap F_E \neq \phi\}$
4. $\delta_D(S, a)$ is computed for $a \in \Sigma$ and $S \in Q_D$ by

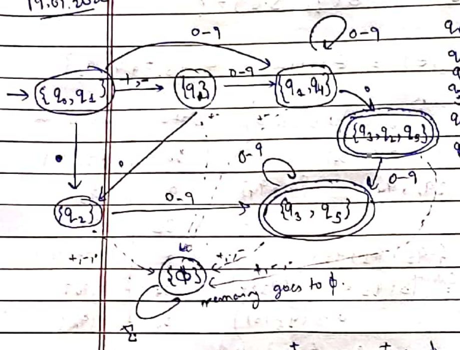
(a) let $S = \{p_1 \ldots p_k\}$.

(b) Compute $\bigcup_{i=1}^{k} \delta_E(p_i, a) = \{r_1, r_2, \ldots r_m\}$

(c) then $\delta_D(S, a) = \bigcup_{j=1}^{m} \epsilon\text{-closure}(r_j)$.

Scanned with CamScanner

19.01.2020



	∈ - closure
q_0	$\{q_0, q_1\}$
q_1	$\{q_1\}$
q_2	$\{q_2\}$
q_3	$\{q_3, q_5\}$
q_4	$\{q_4\}$

$q_0 q_1 \xrightarrow{+} q_1 \xrightarrow{+} \phi \Rightarrow$ ++23 is invalid token.

Regular Expressions: (RE)

1. The constants ϵ and ϕ are RE $\{\epsilon\}$ ϕ $L(\epsilon) = \{\epsilon\}$ $L(\phi) = \phi$
2. If $a \in \Sigma$ then a is RE : $L(a) = \{a\}$
3. If E and F are RE, then E+F is also RE which denotes
   $L(E+F) = L(E) \cup L(F)$
4. If E and F are RE, then E.F is RE. denoting concatenation L(E) and L(F).
   $L(E.F) = L(E).L(F)$
5. If E is RE, the E^* is a RE
   $L(E^*) = (L(E))^*$
6. If E is a RE then Parenthesis E ie (E) is also RE denoting same language as E.
   $L((E)) = L(E)$

Example:

RE	Set
a	$\{a\}$
b	$\{b\}$
$a+b$	$\{a, b\}$
ab	$\{ab\}$
a^*	$\rightarrow a^0 \cup a^1 \cup a^2 \cup a^3 \ldots$ $\{\epsilon, a, aa, aaa, aaaa, \ldots\}$

string of a with all possible length

$(a+b)^* = \{a,b\}^0 \cup \{a,b\}^1 \cup \{a,b\}^2 \ldots$
$= \{\epsilon, a, b, aa, ab, ba, bb \ldots\}$

Reverse

$\{101\} \rightarrow 101$ $\qquad$ $\{a, b, aab\} \rightarrow a+b+aab$

$\{abab\} \rightarrow abab.$ $\qquad$ $\{\epsilon, 0, 00, 000\} \rightarrow 0^*$

$\{01, 10\} \rightarrow 01+10$

$\{\epsilon, ab\} \rightarrow \epsilon + ab$

$\{1, 11, 111 \ldots\} = \{1\}.\{\epsilon, 1, 11, \ldots\} = 1.1^*$

$RE = \underbrace{(aa)^*}_{\text{even a}} \underbrace{\underbrace{(bb)^*}_{\text{even no of b}} b}_{\text{odd no of b.}}$ $\qquad$ $(aa)^* = \{\epsilon, aa, aaaa, aaaaaa$ even length of a.

even 'a' followed by odd 'b'

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

Scanned with CamScanner

Date:
Page No.:

+ is treated as union

20.01.2020

Identity for Regular Expression

1) $\phi + R = R$
2) $\phi R + R\phi = \phi$
3) $\epsilon R = R\epsilon = R$
4) $\epsilon^* = \epsilon$ and $\phi^* = \epsilon$
5) $R + R = R$
6) $R^* R^* = R^*$
7) $R^* R = RR^*$
8) $(R^*)^* = R^*$
9) $\epsilon + RR^* = \epsilon + R^* R = R^*$
10) $(PQ)^* P = P(QP)^*$
11) $(P+Q)^* = (P^* Q^*)^* = (P^* + Q^*)^*$
12) $(P+Q)R = PR + QR$
13) $R(P+Q) = RP + RQ$

Proof

- $RR^* = \{R\}\{\epsilon, R, RR \ldots\}$
  $= \{R, RR, RRR \ldots\}$

$\epsilon + RR^* = \{\epsilon\} \cup \{R, RR, RRR \ldots\}$
$= \{\epsilon, R, RR, RRR \ldots\} = R^*$

- $(PQ)^* P = \{\epsilon, PQ, PQPQ, PQPQPQ \ldots\} \cup \{P\}$
  $= \{\epsilon P, PQP, PQPQP \ldots\}$
  $= \{P, PQP, PQPQP \ldots\}$
  $= \{P\} \cup \{\epsilon, QP, QPQP \ldots\}$
  $= ~~\{P\}~~ P(QP)^*$

Prove that

$\epsilon + 1^*(011)^*(1^*(001)^*)^* = (1+011)^*$

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

$\Rightarrow (1^*(011)^*)^*$

$(P^* Q^*)^* = \cancel{(P+Q)} \quad (P+Q)^*$

$= (1^*(011)^*)^* = (1+011)^*$

Book: Ullman

Theorem: Let r be RE then there exists ϵ-NFA which accepts $L(r)$

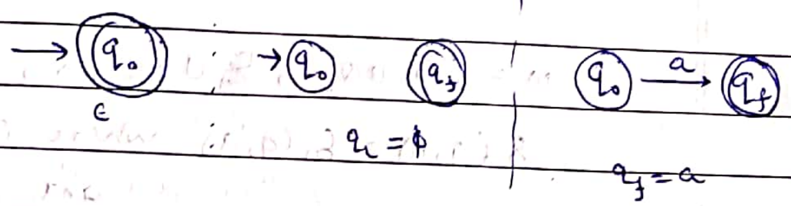
$*, \cdot, +$

Proof

Scanned with CamScanner

Basis (zero operators)

r must be ϵ , ϕ , a ← symbol from Σ



Induction: Let the thm is true for fewer than i operator $i \geq 1$

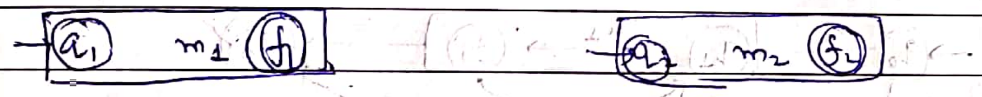
Now, Let RE have i operators, 3 cases

Case 1 $\quad r = r_1 + r_2 \qquad \because r_1, r_2$ must have $< i$ operator

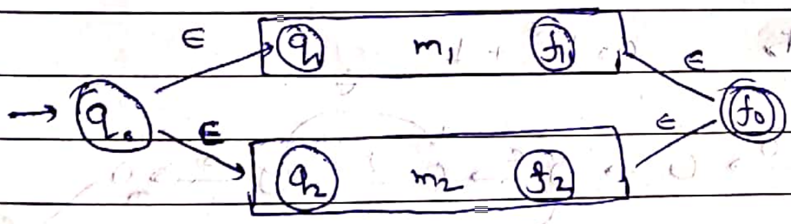
$M_1 = (Q_1, \Sigma_1, \delta_1, q_1, \{f_1\}) \qquad L(M_1) = L(r_1)$

$M_2 = (Q_2, \Sigma_2, \delta_2, q_2, \{f_2\}) \qquad L(M_2) = L(r_2)$

$L(r_1) \qquad\qquad L(r_2)$



$L(r_1 + r_2)$



$M_0 = \{Q_1 \cup Q_2 \cup \{q_0, f_0\}, \Sigma_1 \cup \Sigma_2, \delta_0, q_0, f_0\}$

$\delta_0(q_0, \epsilon) = \{q_1, q_2\}$

$\delta(f_1, \epsilon) = \delta(f_2 \epsilon) = \{f_0\}$

$\delta(q, a) = \delta_1(q, a)$ for q in $Q_1 - \{f_1\}$ and a in $\Sigma_1 \cup \{\epsilon\}$

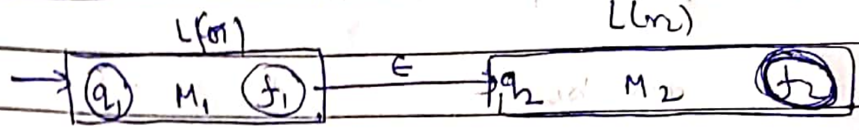
$= \delta_2(q, a)$ for q in $Q_2 - \{f_2\}$ and a in $\Sigma_2 \cup (\epsilon)$

Scanned with CamScanner

Date:

Page No.:

$\boxed{\text{Case 2}}$ $r = r_1 \cdot r_2$



$M = (Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta, \{q_1\}, \{f_2\})$

$\delta(q, a) = \delta_1(q, a)$ where $Q \in Q_1 - \{f_1\}$

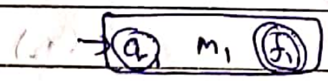
and $a = \Sigma_1 \cup \{\epsilon\}$

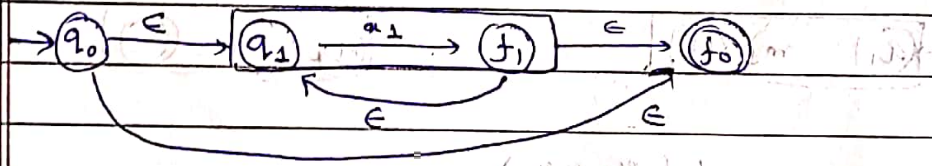
$\delta(f_1, \epsilon) = \{q_2\}$

$\delta(q, a) = \delta_2(q, a)$ for q in q_2 and a in $\Sigma_2 \cup \epsilon$

$\boxed{\text{Case 3}}$ $r = (r_1)^*$

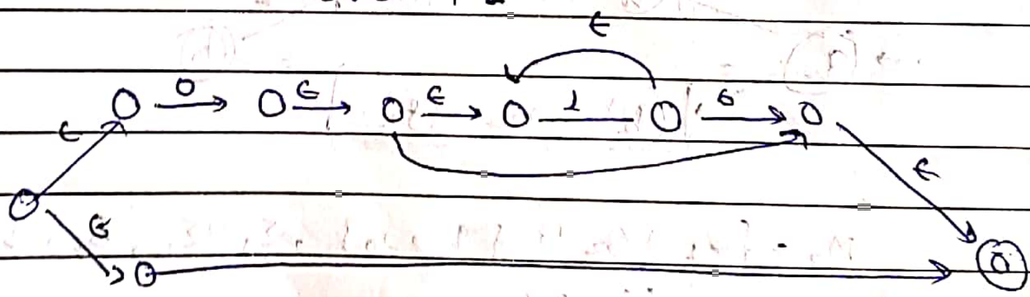
$M_1 = (Q_1, \Sigma, \delta, q_1, \{f_1\})$ $L(M_1) = r$



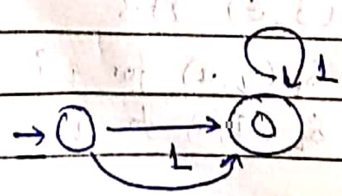


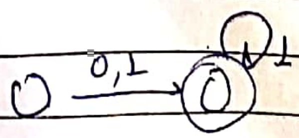
Ulmann

Example Q1. $0 \cdot 1^* + 1$



OR





Scanned with CamScanner

Date:
Page No.:

Now we are given machine and we need to find RE

22.01.2021

If L is accepted by a DFA the L is denoted by RE

n states

DFA $\longrightarrow$ R.E

DFA = $M = \{ \{q_1, q_2, q_3 \ldots q_n\}, \Sigma, \delta, q_1, F \}$

Let R_{ij}^{k} denotes set of all strings that take the m/c from q_i to q_j without going through any state numbered higher than k.

R_{ij}^{k}: $\delta(q_i, x) = q_j$ and $\delta(q_i, y) = q_e$ where y is prefix of x then $l \leq k$

Since the last state is n hence R_{ij}^{n} denote all string that take machine $q_i \rightarrow q_j$

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

(Recursive)

k=0 (no intermediate state) $= R_{ij}^{0}$

$$R_{ij}^{0} = \begin{cases} a \mid \delta(q_i, a) = q_j & i \neq j \\ a \mid \delta(q_i, a = q_j) \cup \{\epsilon\} & \text{if } i = j \end{cases}$$

$q_i \xrightarrow{a} q_j$

$q_i = q_j$ a, ε

Basis: k=0 R_{ij} is the set of finite string which is either ε or a single symbol r_{ij} can be $a_1 + a_2 + a_3 \ldots a_p$

Case 1

$q_i = q_j$ (a_1, a_2, a_p)

Case 2

$q_i \xrightarrow{a_1, a_2, a_3, a_i} q_j$

Case 3 q_i q_j $a_1 + a_2 + a_3 \ldots a_p + \epsilon$

$r = \phi$.

Scanned with CamScanner

Page No.:

Recursive formula:

R_{ij}^{k} involves union $+$
concatenation $\to \circ$
closure. $*$

$$RE^{k-1} = L(RE^{k-1})$$

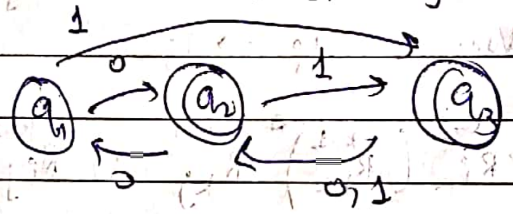
$$= r_{pk}^{k-1}\left(r_{kk}^{k-1}\right)^{*}\left(r_{kj}^{k-1}\right) + r_{ij}^{*}$$

$$L(M) = \bigcup_{q_j \in F} R_{1j}^{n}$$

$$L(M) = r_{1j_1}^{n} + r_{1j_2}^{n} \cdots r_{1j_p}^{n}$$

where $F = \{q_{j_1}, q_{j_2} \cdots q_{j_p}\}$

Example:



$$k=0 \quad \begin{cases} r_{11}^{0} = \epsilon & r_{12}^{0} = 0 & r_{13}^{0} = 1 \\ r_{21}^{0} = 0 & r_{22}^{0} = \epsilon & r_{23}^{0} = 1 \\ r_{31}^{0} = \phi & r_{32}^{0} = 0+1 & r_{33}^{0} = \epsilon \end{cases}$$

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

k=1

$$r_{ij}^{1} = r_{ij}^{0}\left(r_{11}^{0}\right)^{*}\left(r_{1j}^{0}\right) + \left(r_{ij}^{0}\right)$$

$$= r_{i1}^{0}\, r_{1j}^{0} + r_{ij}^{0}$$

i=1,2,3

Date:

Page No.:

$r_{ij}^{1} = r_{i1}^{0} + r_{1j}^{0} + r_{ij}^{0}$

When $i=1$

$r_{1j}^{1} = r_{11}^{0} r_{1j}^{0} + r_{1j}^{0} = r_{1j}^{0} + r_{1j}^{0} = r_{1j}^{0}$

$r_{11}^{1} = \epsilon \quad r_{12}^{1} = 0 \quad r_{13}^{1} = 1$

$i = 2$

$r_{2j}^{1} = r_{21}^{0} r_{1j}^{0} + r_{2j}^{0} = 0 r_{1j}^{0} + r_{2j}^{0}$

$j=1 \quad r_{21}^{1} = 0 r_{11}^{0} + r_{21}^{0} = 0\epsilon + 0 = 0$

$j=2 \quad r_{22}^{1} = 0 r_{12}^{0} + r_{22}^{0} = 00 + \epsilon = \epsilon$

$j=3 \quad r_{23}^{1} = 0 r_{13}^{0} + r_{23}^{0} = 01 + 1 = 01+1$

$i=3$

$j = 1, 2, 3$

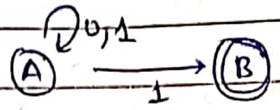
Date :
Page No. :

Neso special.

→ conversion of NFA to DFA

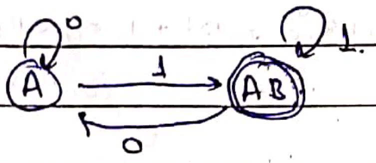
(1) L = {set of strings over (0, 1) ends with 1}

NFA



δ_N	0	1
A	{A}	{A,B}
B	ϕ	ϕ

DFA



δ_D	0	1
A	A	AB
(A ∪ B) = AB	A	AB.

AB is new state

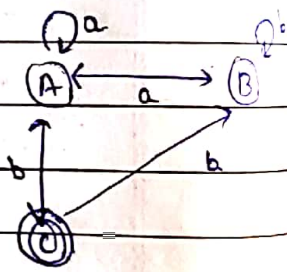
Now there is no way we can reach B.

(2) Given M = [{A, B, C}, (a, b), δ, A, {C}] where

δ_N	a	b
→ A	A,B	C
B	A	B
* C	ϕ	A,B

find DFA.

NFA

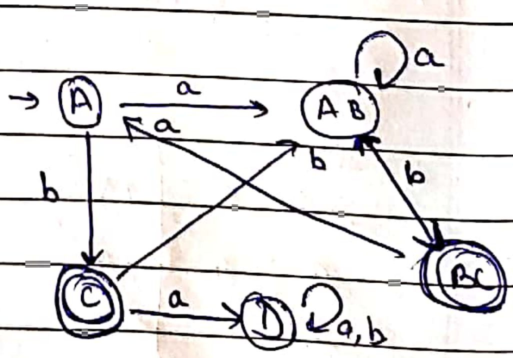


δ_D	a	b
A	AB	C
AB	AB	CB
BC	A	AB
C	D	AB
→ D	D	D

Final state (BC, C)

Dead state (D)

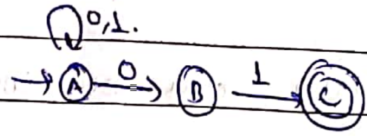
DFA



Scanned with CamScanner

Date :
Page No. :

(3) { Set of all strings (0,1) ends with '01' }

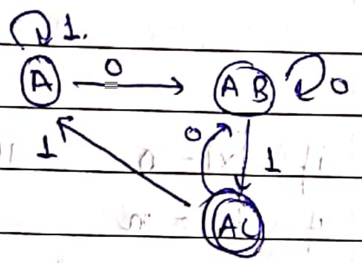


NFA :

δ_N	0	1
A	A,B	A
B	ϕ	C
(C)	ϕ	ϕ

δ_D	0	1
A	AB	A
AB	AB	AC
(AC)	AB	A

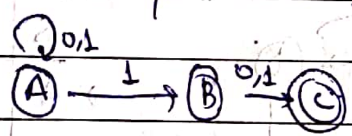
DFA =



(4) { second last symbol is 1. Σ(0,1) }

1010
10
110
11

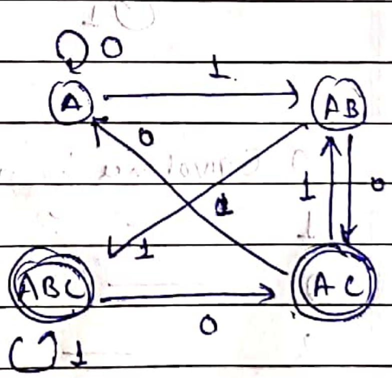
NFA



δ_N	0	1
→ A	A	A,B
B	C	C
(C)	ϕ	ϕ

DFA

δ_D	0	1
A	A	AB
AB	AC	ABC
(AC)	A	AB
(ABC)	AC	ABC



Minimisation of DFA

2 states can be combined if they are equivalent.

i.e

$\delta(A, X) \rightarrow F$ and $\delta(B, X) \rightarrow F$

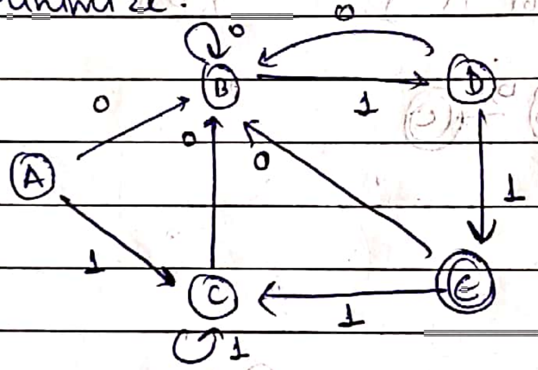
or

$\delta(A, X) \not\rightarrow F$ and $\delta(B, X) \not\rightarrow F$

X is any equivalent state.

if $|X| = 0$: then A and B are 0 equivalent.
if $|X| = n$ ——— n ———.

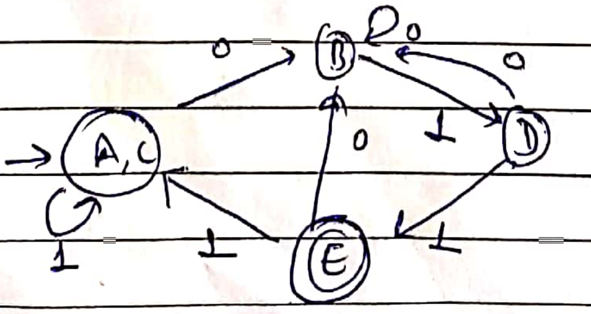
eg Minimize!



δ	0	1
→ A	B	C
B	B	D
C	B	C
D	B	E
(E)	B	C

		Non final	final
0 Equivalence	:	{A,B,C,D}	{E}
1 ———	:	{A,B,C} {D} {E}	
2 ———	:	{A,~~B~~C} {B} {D} {E}	
3 ———	:	{A,~~B~~C} {B} {D} {E}	

Either same state or if state are different, those state should belong to same set.



δ	0	1
{A,C}	B	(A,C)
{B}	B	D
{D}	B	E
{E}	B	(A,C)

TOC 24

TOC-41

→ Empty.

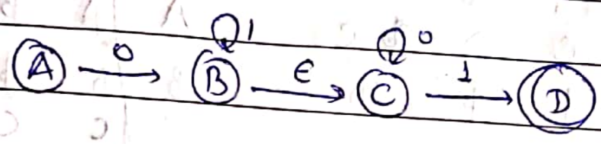
→ $\boxed{\epsilon-NFA}$

$M = \{Q, \Sigma, \delta, q_0, F\}$

$\delta_N = Q \times \Sigma \rightarrow 2^Q$

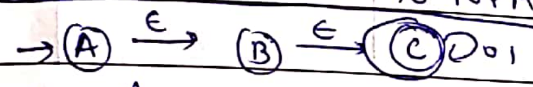
$\delta_{\epsilon N} = Q \times (\Sigma \times \epsilon)^* = 2^Q$

Eg :



Every state on ϵ goes to itself.

Conversion. ϵ-NFA to NFA

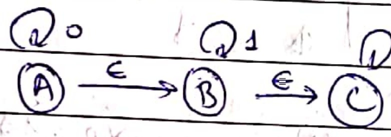


State ϵ^* Input ϵ^*

$\epsilon^*_A \rightarrow \{A, B, C\}$ $\epsilon^*_B = \{B, C\}$ $\epsilon^*_C = \{C\}$

Eg:

$\delta_{\epsilon N}$	0	1
A	A	ϕ
B		
C		



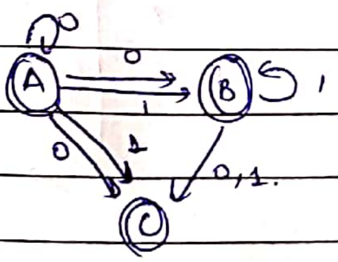
	ϵ^*	0	ϵ^*
A →	A	A	A B C
	B	ϕ	–
	C	C	C

		1	
A →	A	ϕ	
	B.	B.	BC
	C	C	C

A, B can reach C via ϵ

δ_N	0	1
A	{A, B, C}	{B C}
B	BC	BC
C	C	C

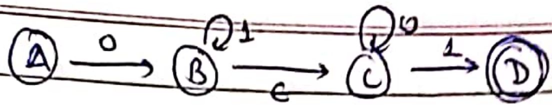
	ϵ^*	0	ϵ^*	1
B	B	ϕ	~~BC~~	
	C	C	C	



Final states ~~are~~ in NFA are those which can reach to final state of ∈ in ∈-NFA.

Date:
Page No.:

(2)

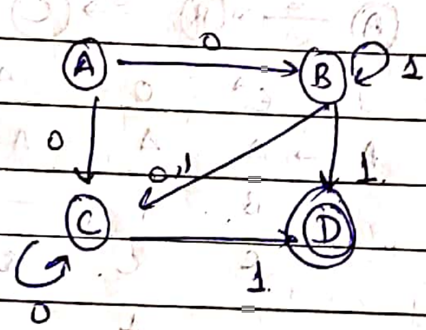


δ_{EN}	0	1	ϵ^*
A	~~B~~ B	ϕ	A
B	ϕ	B	B,C
C	C	D	~~ϕ~~ C
D	ϕ	ϕ	D

~~A~~	ϵ^*	0	ϵ^*		ϵ^*	1	ϵ^*
A	A	B	BC	A	A	ϕ	ϕ
B	B	ϕ	ϕ	B	B	B	BC
	C	C	C		C	D	D
C	C	C	C	C	C	D	D
D	D	ϕ	ϕ		D	ϕ	ϕ

Equivalent NFA.

	0	1
A	B,C	ϕ
B	C	BCD
C	C	D
D	ϕ	ϕ



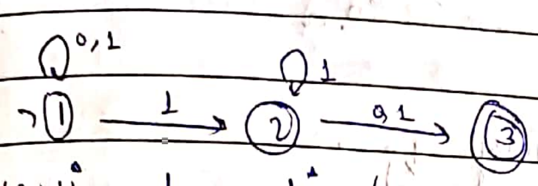
: NFA

44

TOC-44

Date:
Page No.:

25.01.2020



$(0+1)^*$ $\quad 1 \quad 1^* \quad (0+1)$

RE = $(0+1)^* \cdot 1 \cdot 1^* \cdot (0+1)$ (only applicable when we have 1 path)

Arden's Theorem:

let P and Q be 2 regular expression over Σ. If P doesn't contain ε, then following in R:

$\boxed{R = Q + RP}$ has a unique solution $R = QP^*$

$Q + RP$

$= Q + QP^*P = Q(\epsilon + P^*P) = QP^*$ $\quad (\epsilon + R^*R = R^*)$

Proof that is a soln

Hence equation (1) $R = Q + RP$ is satisfied when $R = QP^*$, Hence this is a solⁿ of given equation.

Proof that its a unique soln

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

$= Q + QP + (Q+RP)P^2 = Q + QP + QP^2 + RP^3$

$= Q + QP + QP^2 + (Q+RP)P^3$

$= Q + QP + QP^2 + QP^3 + RP^4$

$= \underbrace{Q + QP + QP^2 + \dots QP^i} + RP^{i+1}$

$= Q(\epsilon + P + P^2 \dots P^i) + RP^{i+1}$

Any string w of length i if $w \in R$ $|w| = i$, then w belongs to.

P doesn't contain ε, $P^i \rightarrow RP^{i+1}$] cannot be here

w belongs to the first form $Q(\epsilon + P + P^2 \dots P^i)$

$\rightarrow Q + QP + QP^2 \dots QP^i + QP^*P^{i+1} = QP^*$

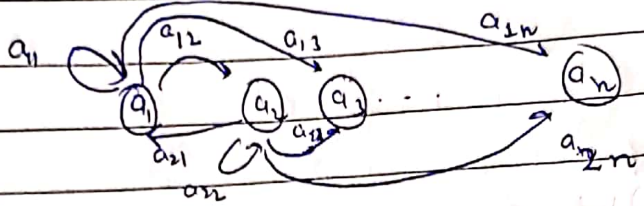
$= Q[\epsilon + P + P^2 \dots P^i + P^*P^{i+1}] = QP^*$

Scanned with CamScanner

28/9/2021

RA $R = Q + RP$

then $R = QP^*$



→ every state q_i to every q_j.

→ if no such transition exist from q_i to q_j

then $a_{ij} = \phi$ $(i=j, i\neq j)$

Ass

(i) There is only 1 start state.

(ii) There is know ∈ moves in m/c

(iii) q_i will also represent the r.e. of all strings accepted by system even through q_i is a state also.

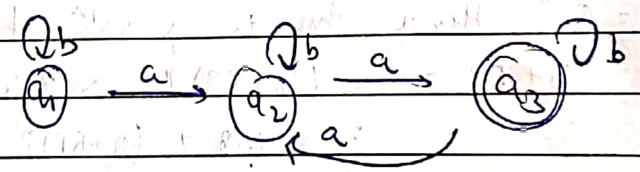
Zvi Kohavi Book

$q_1 = \epsilon + q_1 a_{11} + q_2 a_{21} + q_3 a_{31} \cdots q_n q_{n1}$

$q_2 = q_1 a_{12} + q_2 a_{22} + q_3 a_{32} \cdots q_n a_{n2}$

$q_n = q_1 a_{1n} + q_2 a_{2n} \cdots q_n a_{nn}$

En =



$q_1 = \epsilon + q_1 b$. — ①

$q_2 = q_1 a + q_2 b + q_3 a$ = ②

$q_3 = q_2 a + q_3 b$ → ③

$q_1 = b^*$ $(R = Q + RP = R = QP^*)$

$q_3 = (q_2 a) b^*$

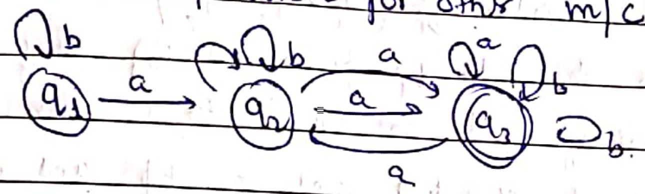
$q_2 = b^* a + q_2 b + (q_2 a) b^* a$

Date:
Page No.:

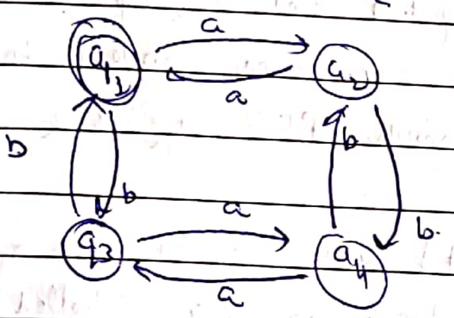
$= \underbrace{b^* a}_{Q} + q_2 (\underbrace{b + ab^* a}_{P})$

$q_2 = (b^* a)(b + ab^* a)^*$

$q_3 = ((b^*a)(b+ab^*a)^* a)\, b^*$ is regular expression for other m/c.



Example:



Accept even a, b.

$q_1 = \epsilon + q_2 a + q_3 b$

$q_2 = q_1 a + q_4 b$

$q_3 = q_1 b + q_4 a$

$q_4 = q_3 a + q_2 b$

$q_1 = \epsilon + (q_1 a + q_4 b) a + (q_1 b + q_4 a) b$

$= \epsilon + q_1 aa + q_4 ba + q_1 bb + q_4 ab$

$= \epsilon + q_1 (aa + bb) + q_4 (ab + ba)$

$q_4 = (q_1 b + q_4 a) a + (q_1 a + q_4 b) b$

$q_4 = q_1 (ba + ab) + q_4 (aa + bb)$

$q_4 = q_1 (ba + ab)(aa + bb)^*$

$q_1 = \epsilon + q_1 [(aa + bb) + (ba + ab)(aa + bb)^* (ab+ba)]$

$q_1 = [(aa + bb) + (ba + ab)(aa + bb)^* (ab + ba)]^*$

$q_1 = ((ba + ab)(aa + bb)^* (ab + ba) + (aa + bb))^*$

Scanned with CamScanner

29/01/2020

State reduction for finding RE of FA.

Sudkamp – Person Algo.

Let q_0 be the start state $q_f \rightarrow$ final.

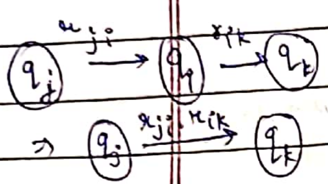
1. do

Chose node q_i $(q_i \neq q_0, q_f)$

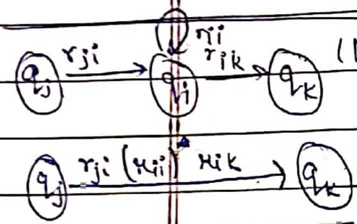
Delete the node q_i from FA(G) using following steps:

{ For every j, k not equal to i (this include $j = k$)

(a) if $r_{ji} \neq \phi$ and $r_{ik} \neq \phi$ and $r_{ii} = \phi$ then add an arc from node j to node k labelled with RE $r_{ji} r_{ik}$



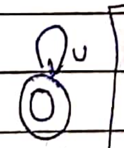
(b) if $r_{ji} \neq \phi$ and $r_{ik} \neq \phi$ and $r_{ii} \neq \phi$ then add arc from q_i to q_k labelled as $r_{ji}(r_{ii})^* r_{ik}$



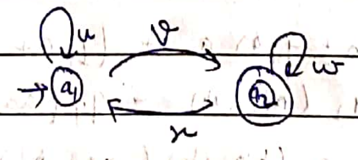
(c) if nodes q_j to q_k have arcs labelled $r_1 \dots r_p$ connecting them, then replace the arcs by a single arc labelled as $r_1 \dots r_p$

}

Remove the nodes q_i and all arcs incident on it in FA(G)



RE : u^*



$q_2 = u^* v (w + x u^* v)^*$

$q_1 = \epsilon + q_1 u + q_2 x$

$q_2 = q_2 w + q_1 x$

Proof:

$q_1 = q_1 u + q_2 x + \epsilon$

$q_2 = q_2 w + q_1 v$

~~$q_1 = \epsilon + q_1 u$~~ $= q_1 = u^*$

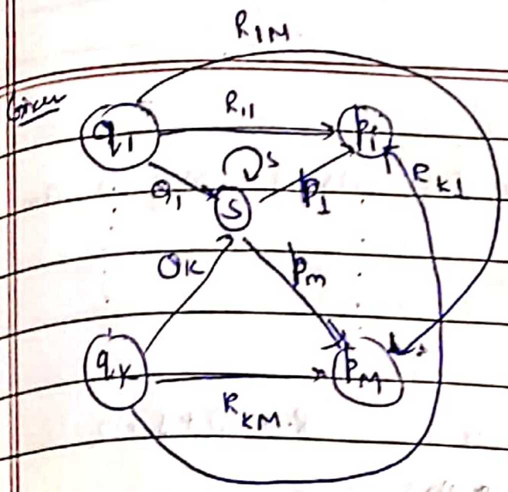
$q_2 = v q_1 +$

$q_1 = \epsilon + q_1 u + (q_2 w + q_1 v) x$

Date:

Page No.:

Given



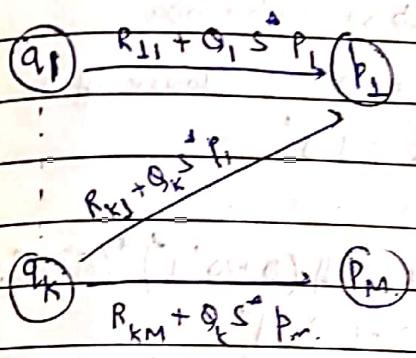
Q_i is label from q_i to S

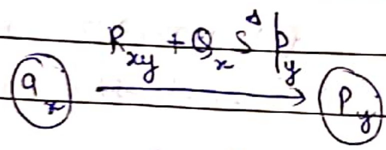
P_i is label from S to p_j

$q_1, q_2 \ldots q_k$] Predecessors

$p_1, \ldots p_m$] successor

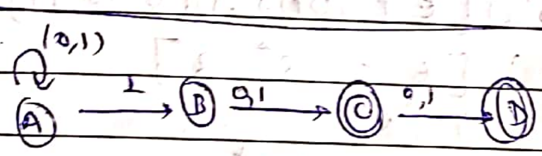
R_{ij} is re from q_i to p_j

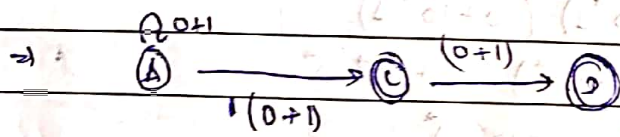




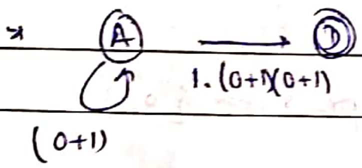
Generally

example





eliminate C



eliminate D

$(0+1)^* \; 1.(0+1)$

$R = (0+1) \qquad S = 1(0+1)(0+1)$

$T = \phi \qquad U = 01$

$(0+1)^* \; 1(0+1)(0+1)$

Ans $\qquad x + y$

Date:
Page No.:

New special part 2

TOC-45

R.E are used for representing sets of string in an algebraic fashion.

sets → RE.

- $\{0,1,2\}$ = 0 or 1 or 2 $R = 0+1+2$
- $\{\epsilon, ab\}$ = $R = \epsilon + ab$
- $\{abb, a, b, bba\}$ $R = abb + a + b + bba$
- $\{\wedge, 0, 00, 000 \ldots\}$ $\Rightarrow R = 0^*$ closure of 0.
- $\{1, 11, 111 \ldots\}$ $\Rightarrow R = 1.1^*$

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

$\Rightarrow (1+00^*1)[\epsilon + (0+10^*1)^*(0+10^*1)]$.
$(-)[\epsilon + R^* R]$
$(-)[R^*]$
$= (1+00^*1)(0+10^*1)^*$
$1(\epsilon+00^*)(0+10^*1)^*$
$1.0^*.(0+10^*1)^*$

→ Design an regular expression. $\{a,b\}$

(1) language string of exactly length 2
$L_1 = \{aa, bb, ab, ba\}$
$R = aa + bb + ab + ba$
$= (a+b)(a+b)$

say of exactly 3 ... $(a+b)(a+b).(a+b)$. ....
~~$(a+b+ \ldots +z)(a+b+ \ldots z)$~~

Scanned with CamScanner

Date:
Page No.:

(2) Atleast 2

$L_1 = \{aa, bb, ab, ba, aaa, aba \ldots \ldots \infty\}$

$R = \underbrace{(a+b)(b+a)}_{\text{exactly 2}} \; (a+b)^*$

exactly 2 $\quad$ more than it.

(3) Atmost 2 :

$L_1 = \{\epsilon, a, b, aa, ab, ba, bb\}$

$\Rightarrow (\epsilon+a+b) + (a+b)(a+b).$

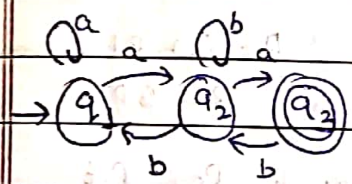
$= \cancel{(\epsilon+a+b) + (\epsilon+a+b)(\epsilon+a+b)}.$

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

so we can take any one from ~~one~~ it and other for this. (since its OR and later we can AND them).

$\Rightarrow \epsilon a = a, \; aa, ab, ba, bb, \; b\epsilon = b, \; \ldots \epsilon.$

RE for given NFA



$q_3 = q_2 . a$

$q_2 = q_1 a + q_2 b + q_3 b$

$q_1 = q_1 a + \epsilon + q_2 . b$

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

$q_2 = q_1 a + q_2 b + q_2 ab.$

$\underbrace{q_2}_{R} = \underbrace{q_1 a}_{Q} + \underbrace{q_2}_{R}\underbrace{(b+ab)}_{P}$ $\qquad R = Q + RP$

$R = QP^*$

$q_2 = q_1 a\,(b+ab)^*$

$q_1 = q_1 a + \epsilon + q_1 a\,(b+ab)^* b.$

$q_1 = \epsilon + q_1\,(a + a(b+ab)^*)\,b.$

$q_1 = \epsilon\,((a + a(b+ab)^*)\,b)^*$

$q_1 = ((a + a(b+ab)^*)\,b)^*$

Scanned with CamScanner

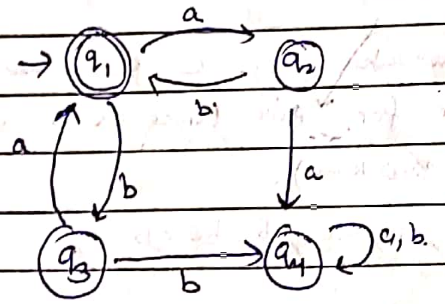
Date:
Page No.:

Now substitute in final state

$$q_3 = q_2 a$$
$$= (q_1 a (b+ab)^*) a$$

$$q_3 = (((a + a(b+ab)^*) b)^* a (b+ab)^*) a$$
= Required RE for given NFA.

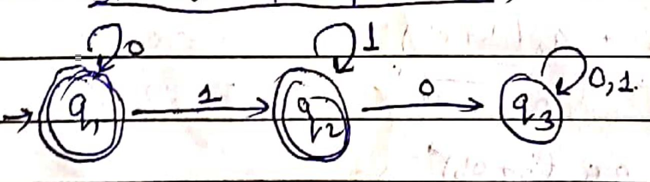
→ DFA to RE.



$q_1 = \epsilon + q_2 b + q_3 a$
$q_2 = q_1 a$
$q_3 = q_1 b$
$q_4 = q_2 a + q_3 b + q_4 (a+b)$

$q_1 = \epsilon + q_1 ab + q_1 b.a$
$q_1 = \epsilon + q_1 (ab+ba)$ $\qquad R = Q + RP$
$q_1 = (ab+ba)^*$ $\qquad R = QP^*$
q_1 is final state, hence is Req R.E

→ ~~RE~~ DFA (multiple final states) to RE



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

$\boxed{q_1 = 0^*}$

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

Arden's Th

So Final states are q_1 and q_2

So $RE = RE_{q_1} + RE_{q_2}$

$$0^* + 0^* 1 . 1^*$$

$$= 0^* (\epsilon + 1.1^*)$$

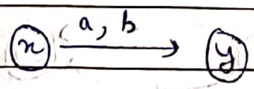
$$\boxed{0^* 1^*} \rightarrow \text{Ans.}$$

$$= \{\epsilon, 0, 00, 000 \ldots\} \{\epsilon, 1, 11, 111 \ldots\}$$

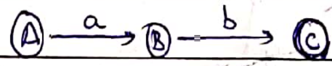
→ **Conversion of RE to FA**

Rules:

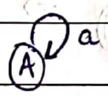
$a+b$



$a.b$

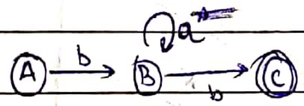


a^*

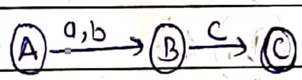


Example:

(1) $b a^* b$



(2) $(a+b) c$

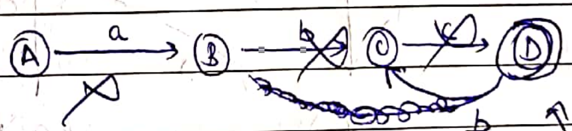


(3) $a (bc)^*$

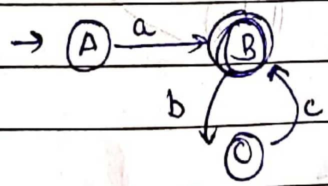
a ✗

a b c

a b c b c



doesn't accept a



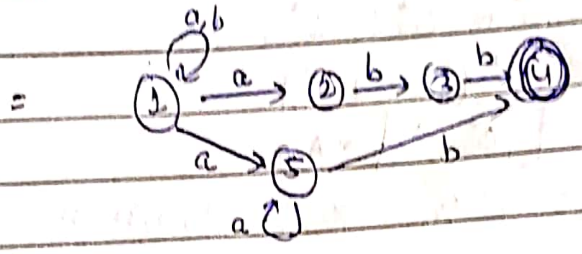
$a^+ = aa^*$

Date :
Page No. :

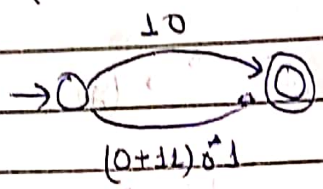
$(a|b)^* (abb \mid a^+ b)$

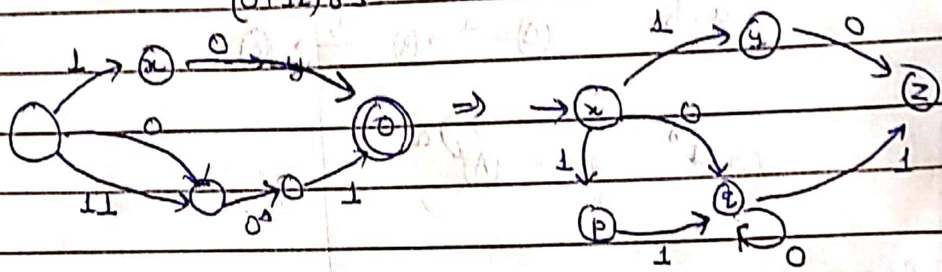
$\rightarrow (a+b)^* (abb + aa^* b)$

=



$\rightarrow 10 + (0+11)0^*1$

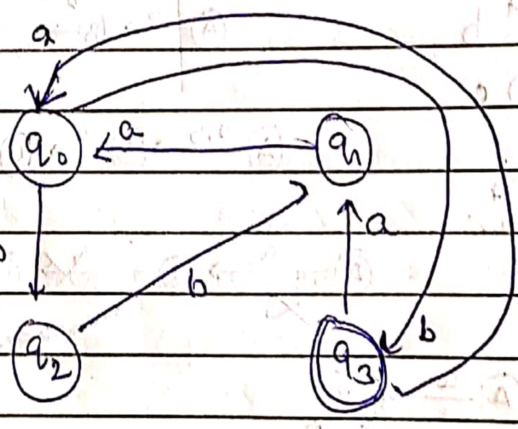




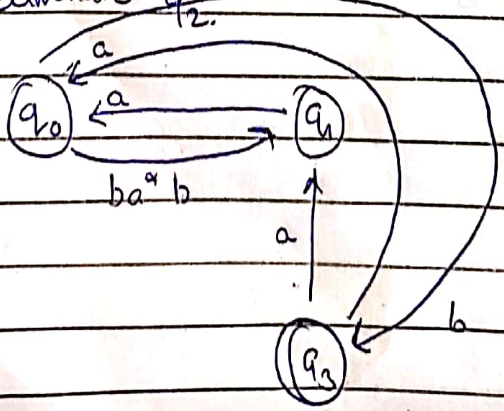
TOC-57

1/02/24

Example



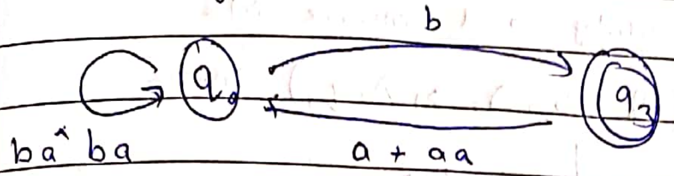
lets eliminate q_2.



Scanned with CamScanner

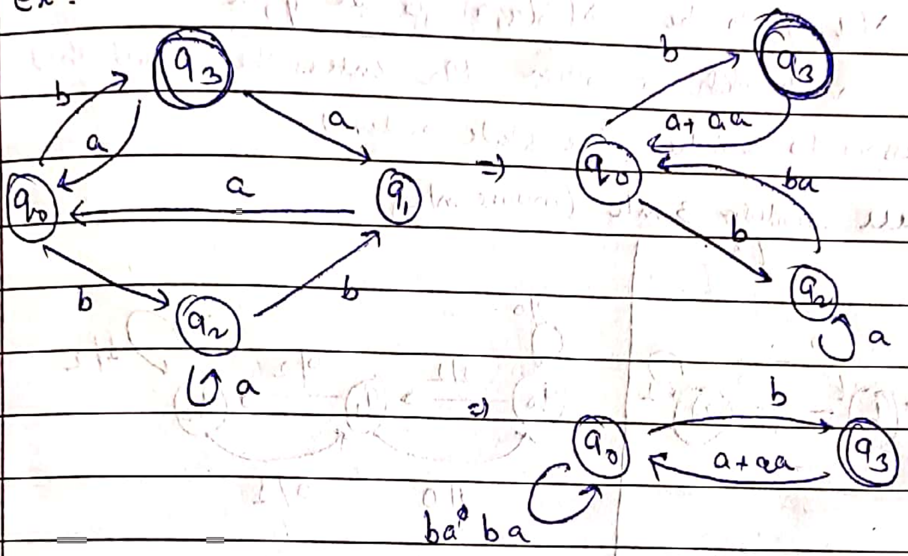
Date :
Page No. :

Eliminating q_2



$$q_3 = (ba^* ba)^* b \left((a+aa)(ba^* ba)^* b \right)^*$$

Ex :



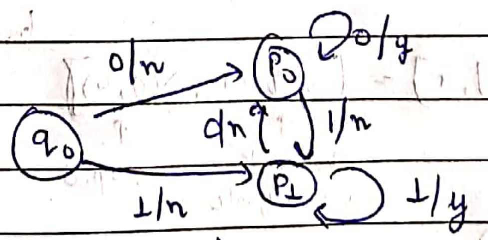
4/02/2021 Mealy m/c.

(Same intro to mealy / moore i.g.)

I missed some part ! :(

Example: Mealy m/c for language who last two sybol are same.

= $\Sigma = \{0, 1\}$ i.e 00, 11 $\Delta : \{y, n\}$ yes no.



$Q = \{q_0, P_0, P_1\}$
$\Sigma = \{0, 1\}$
$\Delta = \{y, n\}$
q_0 = initial state

δ	0	1
q_0	P_0	P_1
P_0	P_0	P_1
P_1	P_0	P_1

λ	0	1
q_0	n	n
P_0	y	n
P_1	n	y

Scanned with CamScanner

Equivalence of mealy and moore m/c

$M_1 = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ moore

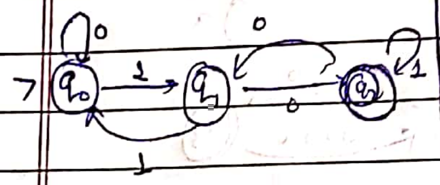
↓ equivalent

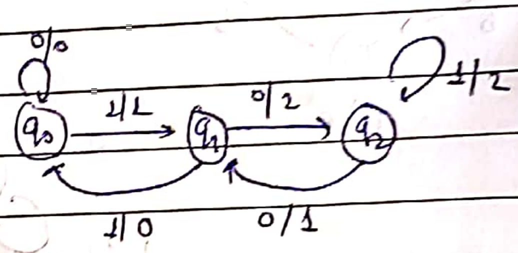
$M_2 = (Q, \Sigma, \Delta, \delta, \lambda', q_0)$ mealy

$\lambda'(q,a)$ to be $\lambda(\delta(q,a))$ for all q, a with each transition M_2 enters the output that M_1 associates with the state entered.

Residue modulo 3 m/c (moore m/c)

%3





% 3 = 0 1 2

Moore m/c | mealy m/c

Theorem: let $M_1 = (Q, \Sigma, \Delta, \delta, \lambda, q_0)$ be a mealy m/c. then their is equivalent Moore m/c M_2 for M_1.

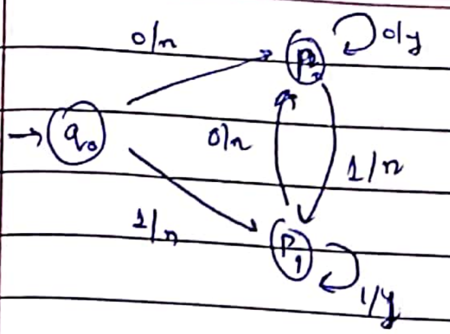
let $M_2 =$ ~~(Q×δ~~ $(Q \times \Delta, \Sigma, \Delta, \delta', \lambda', [q_0, b_0])$

$b_0 \rightarrow$ arbitrary symbol from Δ.

$$\delta'([q, b], a) = [\delta(q,a), \lambda(q,a)]$$

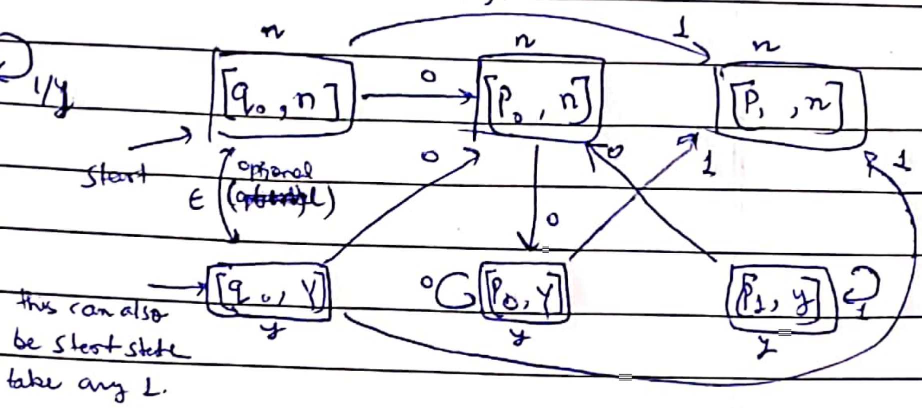
$\delta(q,a)$ ↑ Next state In Mealy m/c; $\lambda(q,a)$ ↑ output

$$\lambda'([q, b]) = b$$

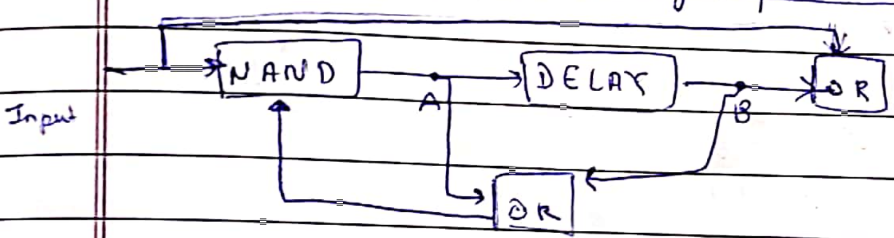


$Q \times \Delta$ $Q = \{q_0, p_0, p_1\}$

$\Delta = \{y, n\}$



Design a mealy m/c for following sequential ckt -:



A	B	
0	0	q_0
0	1	q_1
1	0	q_2
1	1	q_3

$A' = A_{next}$

$A = A_{old}$

$B' = A$ Delay acts like a DFF

$A' = (A \text{ or } B)$ NAND (input)

Output = (input) or B

Let q_0 ie $A = 0$ $B = 0$ input $= 0$

$B' = 0$ $A' = (0)$ NAND (input) $= 1$

output = (input) or B = input = 0

m/c moves to state q_2 with output 0.

input = 1 $B' = 0$ $A' = 1$ output = 1

m/c moves to q_2 with output 1.

Scanned with CamScanner