TAFL

DATE: / /
PAGE:

Theory of Automata

Study of Abstract computing devices

(i) finite Automata

(ii) Pushdown Automata

(iii) Linear bounded Automata.

(iv) Turing Machine.

Hardware

Software

Computer System

Formal language

Study of Abstract Programming language

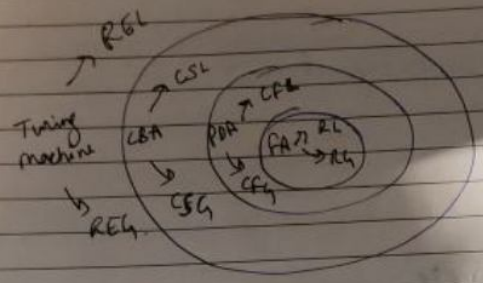
(i) Regular language & Regular Grammar

(ii) context free language & context free Grammar.

(iii) context ~~free~~ sensitive language & context sensitive Grammar.

(iv) Recursive enumerable language & Grammar.

Good Write



Alphabets, String & language.

- Alphabets → Set of Symbol
- String → list of Symbol from Alphabets
- language → Set of String

Alphabets An alphabet is a finite, non-empty set of symbol.
Denoted by Σ

String A string is a finite sequence of symbol chosen from an alphabet.

language A set of strings all of which are chosen from some Σ^*, is called language.

$$L \subseteq \Sigma^*$$
$$L \subseteq \Sigma^1 \cup \Sigma^2 \cup \Sigma^3$$

(i) Empty String → ϵ

(ii) length of string → $|011| = 3$
$|\epsilon| = 0$

Good Write

(ii) Power of an Alphabet → Σ^K

Eg: $\Sigma = \{a, b, c\}$

$\Sigma^1 = \{a, b, c\}$ $= 3^1$

$\Sigma^2 = \{aa, bb, cc, ab, ac, ba, bc, ca, cb\}$ $= 3^2$

$\Sigma^3 = \{aaa, aab, aac, aba, abb, abc \ldots\}$ $= 3^3$

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

(iii) Concatenation of string → $x = 011$, $y = 101$

$xy = 011101$

$yx = 101011$

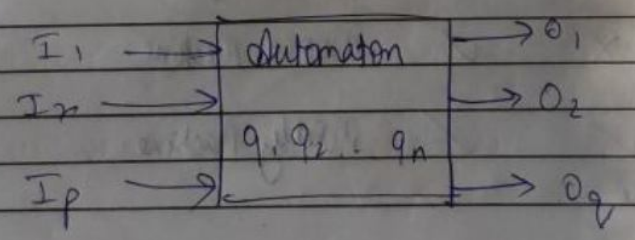
$xy \neq yx$

Automata & Grammars

Automata

An Automaton is defined as a system where energy, materials and information are transferred, transmitted and used for performing some functions without direct participation of humans.

Eg → Automatic Machine, Automatic Packing Machine.



Characteristics of automata

Input
Finite no. of input from Σ eg $I_1, I_2 \cdots I_p$

Output
finite no. of output i.e $O_1, O_2 \cdots O_q$

State
At any instant of time, Automaton can be in one of state
$q_1, q_2 \cdots q_n$

State Relation
Relation b/w present input and present state

Output Relation
Either state only or both state & input

Note

Output depends:	Input	State / Output	Machine	Eg.
	✓	✗	Automaton w/o memory	DFA, NFA
	✓	✓	Automaton w/ memory	Push Down Automata, LBA, Turing
	✗	✓	Moore Machine	
at any instant of time	✓	✓	Mealy Machine.	

Good Write

Grammar

- Mathematical model of grammar was given by Noam Chomsky in 1956.
- BNF (Backus - Naur Form) used to describe definition of grammar.
- A grammar is defined as 4-Tuple

$$G = (V, \Sigma, P, S)$$

(i) $V \rightarrow$ finite non empty set whose elements are called non-terminal

(ii) $\Sigma \rightarrow$ finite non-empty set whose elements are called terminal.

(iii) $V \cap \Sigma = \phi$

(iv) $S \rightarrow$ S is special non terminal ~~(S ∈~~ $(S \in V)$ is start symbol.

(v) $P \rightarrow$ P is set of production Rule Eg: $\alpha \rightarrow \beta$

where α has at least one non terminal.

Note (i) Reverse substitution is not permitted.

Eg: $S \rightarrow AB$ but vice versa not true until specified

(ii) No inversion operation permitted.

Eg: $S \rightarrow AB$ ✓ then $AB \rightarrow S$ ✗

Good Write

Eg $G = (\underbrace{\{S, A, B, E\}}_{V}, \underbrace{\{a, b, c\}}_{\Sigma}, P, S)$

$S \to AB$
$A \to a$
$B \to b$
$E \to c$

- BNF Notation

Developed by John Backus & Peter Naus

$\boxed{\langle \text{Symbol} \rangle ::= \text{expression}}$

$\langle \text{symbol} \rangle \to$ Non Terminal.

expression $\to$ Consisting of one or more symbol separated by |

$::= \to$ left side must be replaced by exp. of right side.

Eg
$\langle \text{Number} \rangle ::= \langle \text{digits} \rangle \mid \langle \text{number} \rangle \langle \text{digit} \rangle$

$\langle \text{digit} \rangle = 0 \mid 1 \mid 2 \mid \ldots \mid 9$

Good Write

Chomsky Hierarchy.

Grammar Type	Type 0	Type 1	Type 2	Type 3
Grammar Accepted	Unrestricted Grammar	Context sensitive Grammar	Context free Grammar	Regular Grammar.
Language Accepted	Recursively Enumerable language	Context Sensitive language	Context Free language	Regular Language
Automaton	Turing Machine	Linear Bounded Automata	Pushdown Automata	Finite State Automata
Example	$\alpha \to \beta$ (At least one non terminal)	$\alpha A \beta \to \alpha \gamma \beta$	$A \to \gamma$	$X \to a$
	$Sab \to ba$	$A \in V$	$A \in V$	a
	$A \to S$	$\alpha, \beta, \gamma \in (\Sigma \cup V)^*$	$\gamma \in (\Sigma \cup V)^*$	$X \to aY$
	$S \to ACaB$	✓ kabhi null nahi hoga!		$X, Y \in V$
		$AB \to AbBc$	$S \to Xa$	$a \in \Sigma$
		$A \to bcA$	$X \to \lambda$	$S \to \Lambda$ (Null)
		$B \to b$	$X \to aX$	
		$bA \to ab$	$X \to \Lambda$	$X \to a \mid aY$
		$bD \to bb$		$Y \to b$
				$X \to \Lambda$

Good Write

DATE: / /
PAGE:

Finite Automata

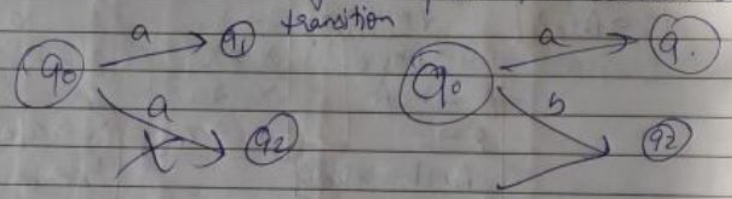
- FA without output
  - DFA
  - NFA
  - ENFA
- FA with output
  - Moore Machine
  - Mealy Machine

FA

A finite automata has a set of states & its 'control' move from state to state in response to external 'input'.

Deterministic finite Automata (DFA)

In DFA on each input there is only one state which the automata can ~~from transform~~ from its current state transition



A DFA consists of 5 tuples. $(Q, \Sigma, \delta, q_0, f)$

$Q \rightarrow$ finite non empty set of state

$\Sigma \rightarrow$ finite non empty set of input.

$\delta \rightarrow$ Transition function $Q * \Sigma \rightarrow Q$

Eg $\delta(q_0, a) \rightarrow q_1$

Good Write

$q_0 \xrightarrow{a} q_1$

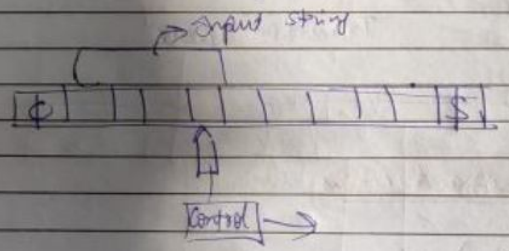
$Q \times \Sigma$ $\qquad$ $Q = \{q_0, q_1, q_2\}$

$3 \times 2 = 6$ $\qquad$ $\Sigma = \{a, b\}$

q_0 $\quad$ $q_0 \in Q$ is initial state

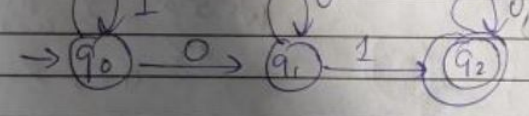
$\rightarrow q_0$

f $\quad$ $f \in Q$ is a set of final state called Acceptance state.



Eg $\quad$ 01.

$L = \{01, 0011, 101, 11011\}$



DFA $\Rightarrow$ $A = \{\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, q_2\}$

Good Write

$\delta = \delta(q_0, 1) \to q_0$

$\delta(q_0, 0) \to q_1$

$\delta(q_1, 0) \to q_1$

$\delta(q_1, 1) \to q_2$

$\delta(q_2, 0) \to q_2$

$\delta(q_2, 1) \to q_2$

Notation for DFA

Transition Diagram

Transition Table

Properties of Transition function

(I) $\delta(q, \wedge) = q$

(II) $\delta(q, w)$

$\delta(q, 1101)$

$\delta(q, xa)$

$\Rightarrow \delta(\delta(q, x), a)$

$w = 1101$

$w = xa \quad x = 110, \; a = 1$

$w = ax \quad a = 1, \; x = 101$

Good Write

DATE: __/__/____
PAGE: _____

Exmpl DFA ⇒ $A = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, q_0)$

	0	1
→ q_0	q_2	q_1
q_1	q_3	q_0
q_2	q_0	q_3
q_3	q_1	q_2

String $w = 110101$

$\delta(q_0, w) = \delta(q_0, 110101)$

$= \delta(q_1, 10101)$

$= \delta(q_0, 0101)$

$= \delta(q_2, 101)$

$= \delta(q_3, 01)$

$= \delta(q_1, 1)$

$= \delta(q_0, \Lambda)$

$= q_0$

language of DFA

A language for DFA $A = (Q, \Sigma, \delta, q_0, F)$ define as

$L(A) = \{w \mid \hat{\delta}(q_0, w) \text{ is in } F$

Regular language

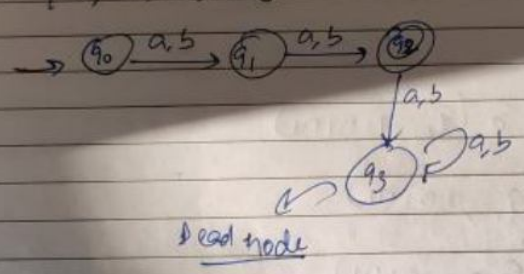
Good Write

Q1 ~~Data~~ Construct a DFA that accept all string over {a, b}

(i) $|w| = 2$ (ii) $|w| \geq 2$ (iii) $|w| \leq 2$

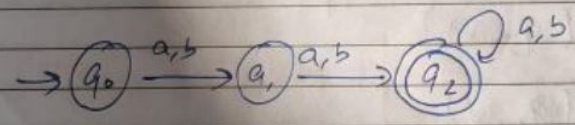
(i) $|w| = 2$

$L = \{aa, bb, ab, ba\}$



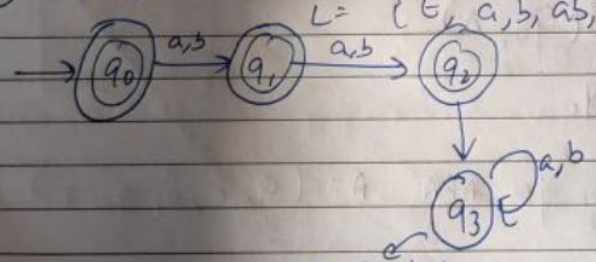
(ii) $|w| \geq 2$

$L = \{aa, ab, ba, ab, abb, aaa \ldots\}$



(iii) $|w| \leq 2$

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



Good Write

DATE: / /
PAGE:

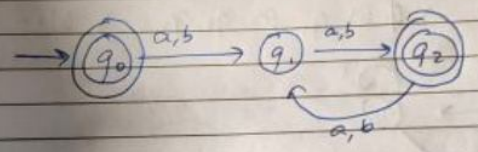
Q2 Construct a DFA $w \in \{a,b\}$

(i) $|w| \bmod 2 = 0$ — Even length

(ii) $|w| \bmod 2 = 1$ — Odd length

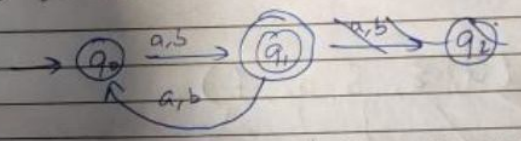
(i) $|w| \bmod 2 = 0$

$L = \{\epsilon, ab, ba, aa, bb, aaaa, bbbb \ldots\}$



(ii) $|w| \bmod 1 = 0$

$L = \{a, b, aaa, bbb \ldots\}$



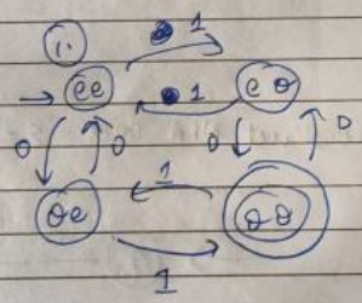
Q3 Construct DFA accepting following language over $\Sigma = \{0,1\}$

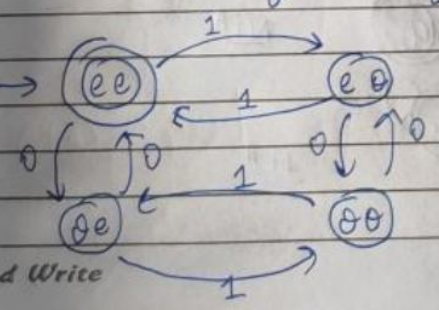
(i) Even 1's & Even 0's

(ii) Odd 1's & Odd 0's.

(i) $L = \{\epsilon$

0	1
e	e
e	o
o	e
o	o





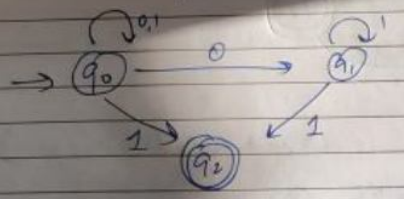
Good Write

Non-Deterministic Finite Automata (NDFA/NFA)

An NFA has the power to be in several states at once for NFA 'δ' is a function that takes a state and input symbol as argument but return zero, one or more state.

Eg $\delta(q_0, a) \rightarrow q_1 \Rightarrow$ DFA

$\delta(q_0, a) \rightarrow \phi$ or q_1 or $q_1, q_2 \Rightarrow$ NFA.

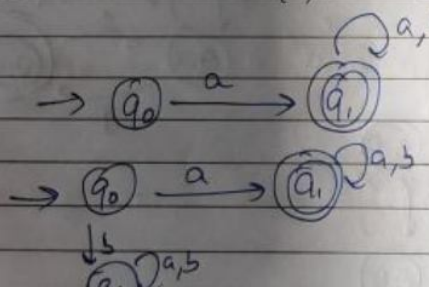


NFA consist 5 tuples $(Q, \Sigma, \delta, q_0, F)$

- Q → Set of State
- Σ → Set of Inputs
- δ → $Q \times \Sigma \rightarrow 2^Q$
- q_0 → $q_0 \in Q$, initial state
- F → $F \subseteq Q$, final state

Eg: $\{1,2\}$ - $(\{\phi\}, \{1\}, \{2\}, \{1,2\})$

Q Construct NFA over $\Sigma = \{a,b\}$, where each string start with a



NFA

DFA

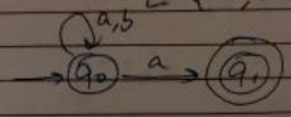
Good Write

DATE: __/__/____
PAGE: _____

Q NFA ending with 'a'

L = {end with 'a'}

L = {a, ba, aa, baa . . . . }



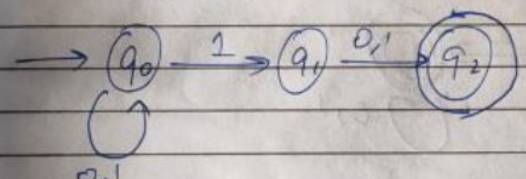
$Q \times \Sigma$
2×2

$(q_0, a) \to \{\phi\}$
$(q_0, b) \to \{q_1\}$
$(q_1, a) \to \{q_0\}$
$(q_1, b) \to \{q_1, q_0\}$

	a	b
q_0	q_0, q_1	q_0
q_1	ϕ	ϕ

Q Construct NFA accepting {0,1} string whose second last element is '1'.

L = {10, 11, 011, 010 . . . . }



$Q \times \Sigma = 2^9$
3 2

$\delta(q_0, 0) \to \{\phi\}$
$\delta(q_0, 1) \to \{q_0\}$
$\delta(q_1, 0) \to \{q_1\}$
$\delta(q_1, 1) \to \{q_2\}$
$\delta(q_2, 0) \to \{q_0, q_1\}$
$\delta(q_2, 1) \to \{q_1, q_2\}$
$\{q_0, q_2\}$
$\{q_0, q_1, q_2\}$

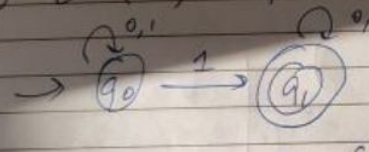
Good Write

DATE: / /
PAGE:

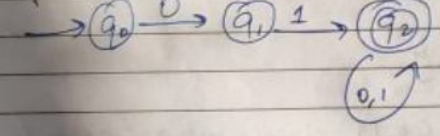
Q Construct NFA that accept all string over {0,1} where

(i) String contain '1'
(ii) Start with 01
(iii) contain 01
(iv) End with 01

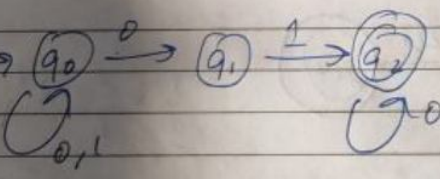
(i) L = { 1, 01, 10, 11 . . . . }



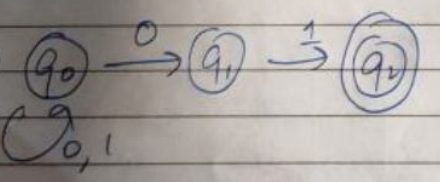
(ii) L = {01, 010, 011 . . . . }



(iii) L = {01, 001, 101, 010, 011 . . . }



(iv)



Good Write

NFA to DFA

Q Construct DFA equivalent to NFA

$m = (\{q_0, q_1\}, \{0,1\}, \delta, q_0, \{q_0\})$

δ

State / E	0	1
→ q_0	q_0	q_1
q_1	q_1	q_0, q_1

(i) Power set. $\{q_0, q_1\} = \phi, \{q_0\}, \{q_1\}, \{q_0, q_1\}$

(ii) Identify initial

q_0

(iii) Identify final

$\{q_0\}, \{q_0, q_1\}$

(iv) δ table

State / E	0	1
ϕ	ϕ	ϕ
→ q_0	$\{q_0\}$	$\{q_1\}$
q_1	$\{q_1\}$	$\{q_0, q_1\}$
→ $\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_1\}$

Q Construct DFA equivalent to NFA.

$m = (\{q_0, q_1, q_2\}, \{a,b\}, \delta, q_0, \{q_2\})$

δ

State / E	a	b
→ q_0	q_0, q_1	q_2
q_1	q_0	q_1
q_2		$q_0 q_1$

State / E	a	b
q_0	$\{q_0, q_1\}$	$\{q_2\}$
$q_0 q_2$	ϕ	$\{q_0 q_1\}$
$q_0 q_1$	$\{q_0 q_1\}$	$\{q_1 q_2\}$
$q_1 q_2$	$\{q_0\}$	$\{q_0 q_2\}$

Good Write

Q Construct DFA equivalent to NFA.

$M = (\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$

$\delta \rightarrow$

State / Σ	0	1
$\rightarrow q_0$	q_0, q_1	q_0
q_1	q_2	q_1
q_2	q_3	q_3
q_3	–	q_2

State / Σ	0	1
$\rightarrow q_0$	q_0, q_1	q_0
q_0, q_1	q_0, q_1, q_2	q_0, q_1
q_0, q_1, q_2	q_0, q_1, q_2, q_3	q_0, q_1, q_3
q_0, q_1, q_2, q_3	q_0, q_1, q_2, q_3	q_0, q_1, q_2, q_3
q_0, q_1, q_3	q_0, q_1, q_2	q_0, q_1, q_2

Good Write

Minimization of DFA

Q Construct minimum state automata equivalent to finite automata described by:

State / Σ	0	1
→ q_0	q_1	q_5
q_1	q_6	q_2
q_2	q_0	q_2
q_3	q_2	q_6
q_4	q_7	q_5
q_5	q_2	q_6
q_6	q_6	q_4
q_7	q_6	q_2

$\pi_0 = \{ \underset{1}{\{q_2\}}, \underset{2}{\{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}} \}$

zero equivalence

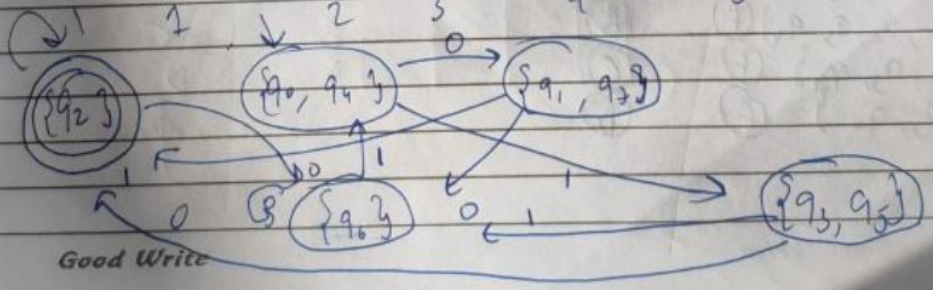
$\pi_1 = \{ \overset{1}{\{q_2\}}, \overset{2}{\{q_0, q_4, q_6\}}, \overset{3}{\{q_1, q_7\}}, \overset{4}{\{q_3, q_5\}} \}$

1 equivalence

$\{q_4, q_6$

$\pi_2 = \{ \overset{1}{\{q_2\}}, \overset{2}{\{q_0, q_4\}}, \overset{3}{\{q_6\}}, \overset{4}{\{q_1, q_7\}}, \overset{5}{\{q_3, q_5\}} \}$

$\pi_3 = \{ \{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\} \}$



Good Write

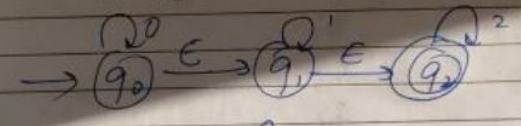
DATE: / /
PAGE:

NFA with ε move

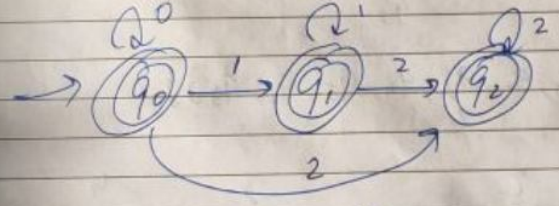
(i) ε - NFA to NFA Conversion
(ii) ε - NFA to DFA Conversion

(i) a) All move q_j should be also originated from q_i

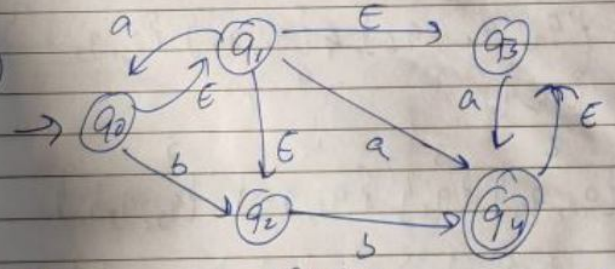
b) If q_j is final, make q_i as final.



q_i q_j



(ii)



NFA State		DFA State	a	b
$\{q_0, q_1, q_2, q_3\}$		→ A	B	C
$\{q_0, q_1, q_2, q_3, q_4\}$	F	B	B	C
$\{q_2, q_3, q_4\}$	F	C	D	D
$\{q_3, q_4\}$	F	D	D	ϕ

Good Write

ε-closure $q_0 = \{q_0, q_1, q_2, q_3\} = A$

Dtransition $[A, a]$ = ε-closure (move (A, a)) =

ε-closure (q_0, q_4)

$\{q_0, q_1, q_2, q_3, q_4\}$

Dtrans $[(A, b)]$ = ε-closure (move (A, b))

= ε-closure (q_2, q_4)

= $\{q_2, q_3, q_4\}$ = C

Dtrans $[B, a]$ = ε-closure (move (B, a))

= ε-closure (q_0, q_4)

= B

Dtrans $[B, b]$ = ε-closure (q_2, q_4) = C

Dtrans $[C, a]$ = $\{q_3, q_4\}$ = D

Dtrans $[C, b]$ = $\{q_3, q_4\}$ = D

Good Write

Myhill - Nerode Theorem

(I) This theorem tests the language is regular or not.

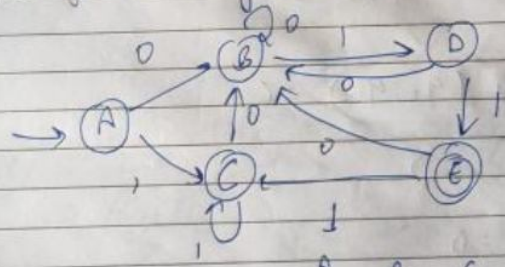
(II) This theorem is used to minimise the DFA.

Given a language L, two string X and Y are said to be in same class, if all possible strings Z, either XZ and YZ are in L or both are not.

i) A language L divides the set of all possible string into mutual exclusive classes.

ii) If L is regular, the no. of classes created by L is finite

iii) If the no. of classes L creates is finite, then L is regular.



	A	B	C	D	E
A					
B	✓				
C		✓			
D	✓	✓	✓		
E	✓	✓	✓	✓	

AB = δ(A, 0) B
~~δ(A, 1) C~~
δ(B, 0) B
δ(B, 1) D
δ(A, 1) C

→ {A, C} {B} {D} {E}

Good Write

Finite State machine with Output

DATE: / /
PAGE:

Moore Machine

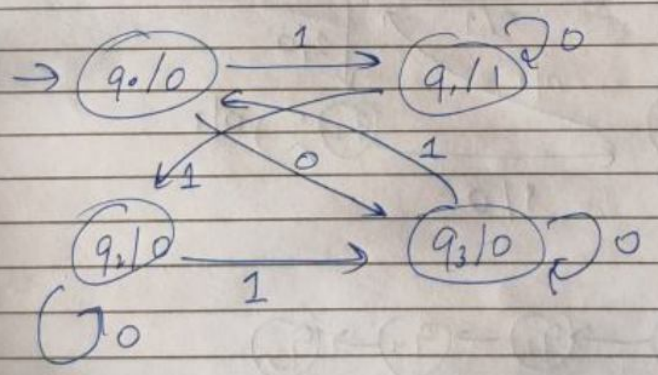
↳ Output depends on present state only

↳ It consists six tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

- Q → Set of state
- Σ → Set of i/p alphabet
- Δ → set of o/p alphabet
- δ → is transition mapping $\Sigma^* Q \to Q$
- λ → is output transition function $Q \to \Delta$
- q_0 → Initial state.

Ex

Present State	Next State (Σ) 0	1	Output λ
→ q_0	q_3	q_1	0
q_1	q_1	q_2	1
q_2	q_2	q_3	0
q_3	q_3	q_0	0



Input
0 1 1 1

Output
→ $q_0 \to q_3 \to q_0 \to q_1$
$\downarrow$
q_2

0 0 0 1 0

Good Write

Moore : $\lambda : Q \to \Delta$

Mealy : $\lambda : Q \times \Sigma \to \Delta$

Mealy machine

Output depends on present state and input.

All tuples remain same

Bas :

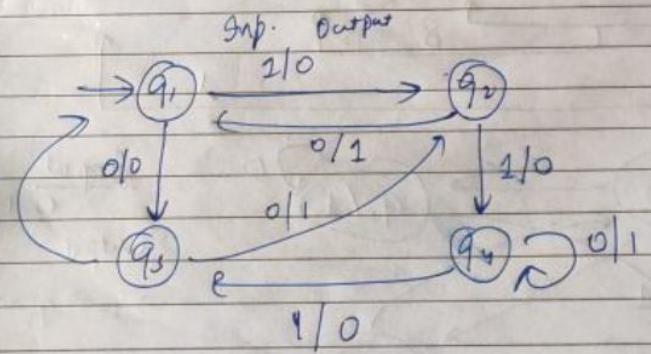
$\lambda \to$ 1's Output function

$\Sigma \times Q \to \Delta$.

$(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

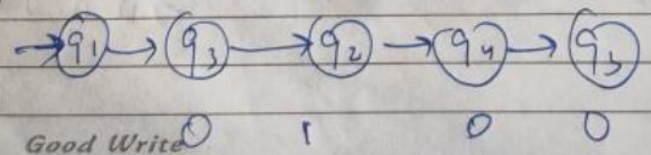
Eg

Present State	0 State	0 Output	1 State	1 Output
$\to q_1$	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0



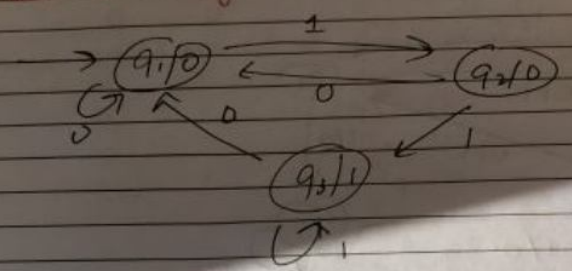
I/P : 0011

O/P :



Good Write

Moore to Mealy



Moore

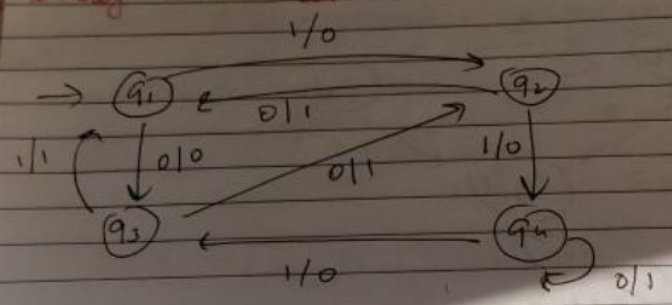
P. State	0	1	λ
→ q_1	q_1	q_2	0
q_2	q_1	q_3	0
q_3	q_1	q_3	1

Mealy

P. State	0		1	
	~~Next State~~ State	Output	State	Output
→ q_1	q_1	0	q_2	0
q_2	q_1	0	q_3	1
q_3	q_1	0	q_3	1

Good Write

Mealy to Moore conversion



Mealy

PS	0 State	0 output	1 State	1 Output
→ q_1	q_3	0	q_2	0
q_2	q_1	1	q_4	0
q_3	q_2	1	q_1	1
q_4	q_4	1	q_3	0

Modified Mealy

$q_1 \rightarrow 1/1$ ☆

$q_2 \rightarrow 1/0$

↳ $q_{20} \rightarrow 0$ ☆

↳ $q_{21} \rightarrow 1$ ☆

$q_3 \rightarrow 0/0$ ☆

$q_4 \rightarrow 1/0$

↳ $q_{40} \rightarrow 0$ ☆

↳ $q_{41} \rightarrow 1$ ☆

Good Write

PS	0 State	0 O/P	1 State	1 O/P
q_1	q_3	0	q_{20}	0
q_{20}	q_1	1	q_{40}	0
q_{21}	q_1	1	q_{40}	0
q_3	q_{21}	1	q_1	1
q_{40}	q_{41}	1	q_3	0
q_{41}	q_{41}	1	q_3	0

Moore

PS	0	1	λ
q_1	q_3	q_{20}	1
q_{20}	q_1	q_{40}	0
q_{21}	q_1	q_{40}	1
q_3	q_{21}	q_1	0
q_{40}	q_{41}	q_3	0
q_{41}	q_{41}	q_3	1

Moore machine : $(n+1)$

Good Write

Regular Expression

1) Any terminal symbol (Σ), Δ, ϕ are RE

2) The Union of two RE R_1 and R_2 written as $R_1 + R_2$ or $R_1 | R_2$

$$\overset{a}{R_1} \qquad \overset{b}{R_2} \rightarrow RE$$

$R_1 + R_2$ $\qquad$ RE $(a+b)$ $\quad L = \{a, b\}$

3) Concatenation of two RE R_1 and R_2 written as $R_1 \cdot R_2$ or $R_1 R_2$

4) Iteration (closure) of RE R_1 written as R^* is RE

$$\overset{a}{R} = R^* = RE$$

$$a^* = RE$$

$$L = \{\epsilon, a, aa, aaa \ldots$$

5) If R is RE then (R) is RE

6) RE obtain recursively by the 1-5 rule is RE.

$$(a+b)^* \quad abb \rightarrow RE$$

Good Write

DATE: / /
PAGE:

Q1 Describe RE:

a) {101} RE = 101

b) {abba} abba

{01, 10} 01+10

{Λ, ab} Λ+ab

{abb, a, b, bba} abb + a + b + bba

{Λ, 0, 00, 000 ...} 0^*

{1, 11, 111 ...} 11^* or 1^+

Q2 Find RE for following language : Σ={1,0}

All string end with 0

$(0+1)^* 00$

Begin w/ 0 and end w/ 1

$0(0+1)^* 1$

{Λ, 11, 1111, 111111 ...}

$(11)^*$

exactly two 0 . $1^* 0 1^* 0 1^*$

At least two 0 . $(0+1)^* 0 1^* 0 (0+1)^*$

Don't end w/ 01

$(0+1)^* (11+10+00)$

11
10
00

Good Write

Algebraic law for RE

Assuming L, M, N are RE

(i) Associativity & Commutativity

$L+M = M+L$ → Comm.

$(L+M)+N = L+(M+N)$] → Assoc.

$(LM)N = L(MN)$

(ii) Identities & Annihilators.

$\phi + L = L + \phi = L$] → Identities

Null. $\epsilon L = L\epsilon = L$

$\phi L = L\phi = \phi$ → Annihilators.

(iii) Distributive

$L(M+N) = LM + LN$ and $(M+N)L = ML + NL$

(iv) Idempotent law.

$L+L = L$

(v) Law involving Closure (*)

$\epsilon^* = \epsilon$ and $\phi^* = \epsilon$

$R^* R^* = R^*$ $RR^* = R^* R$ $(R^*)^* = R^*$

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

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

Good Write

DATE: / /
PAGE:

Q Prove that

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

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

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

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

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

Q P.T $(1+00^*1)+(1+00^*1)(0+10^*1)^*(0+10^*1)$
$= 0^*1(0+10^*1)^*$

$$(1+00^*1)(\Lambda + (0+10^*1)^*(0+10^*1))$$

$$(1+00^*1)(0+10^*1)^*$$

$$(\Lambda+00^*)\ 1\ (0+10^*1)^*$$

$$\underline{0^*\ 1\ (0+10^*1)^*}$$

DATE: / /
PAGE:

* Regular Expression to finite automata (extend Theorem)

i.) Any RL can be accepted by finite automata

ii.) The language accepted by any finite automata is RL

Proof Any RE consist: $\Sigma, \epsilon, \phi, *, +, \cdot$

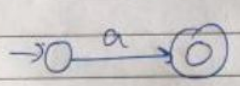
i. If RE consist only one symbol $|R| = 1$

$a \leftarrow \Sigma, \phi, \epsilon$

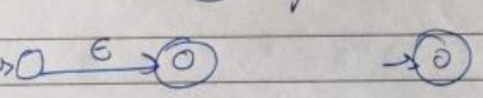
(i) If RE consist only one symbol $|R| = 1$

Σ, ϕ, ϵ

$R = a \quad R = \phi \quad R = \epsilon$

$R = a$ 

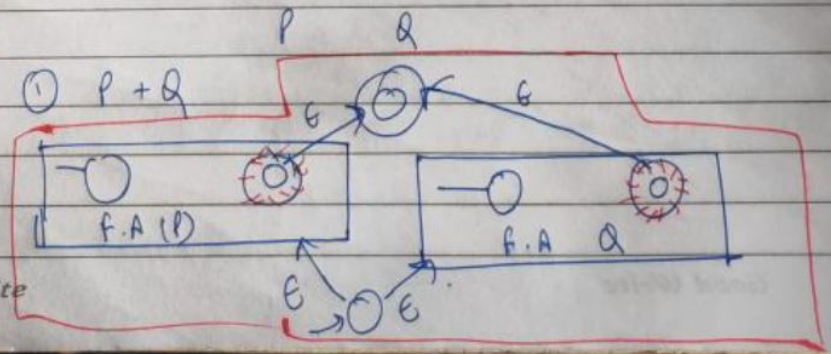
$R = \phi$

$R = \epsilon$ 

(ii) $|R| \geq 1$

① $P + Q$

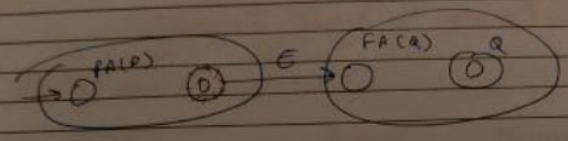
$P+Q$



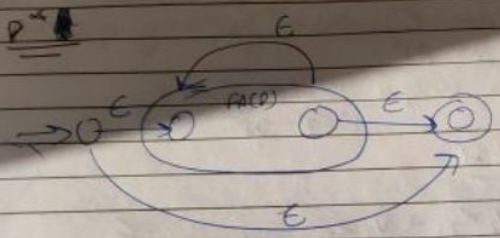
Good Write

DATE: __/__/____
PAGE: _____

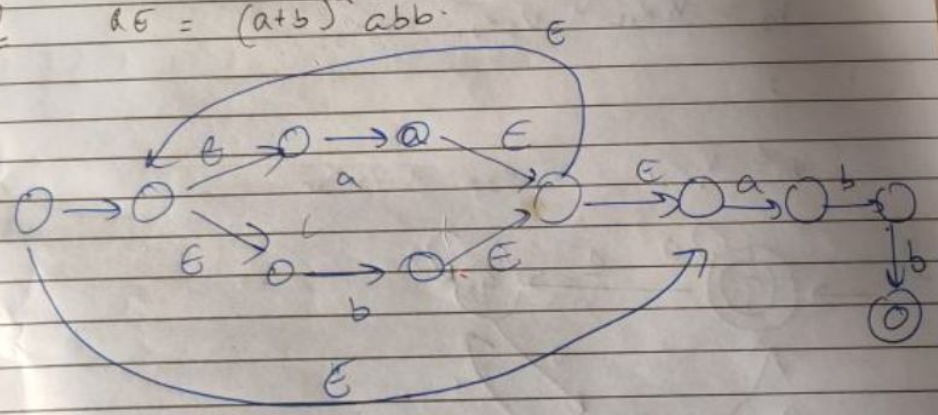
PQ

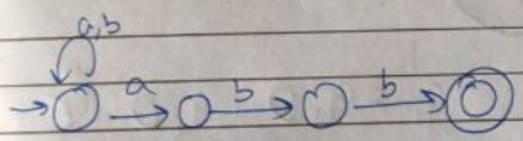


P^*



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





Good Write

Finite Automata to Regular Expression
(Arden's Theorem & Arden's Algebraic Method)

Arden's Theorem

Let P & Q be two regular expression if P does not contain ϵ, then following equation in

$$R = Q + RP$$

has unique solution given by

$$R = QP^*$$

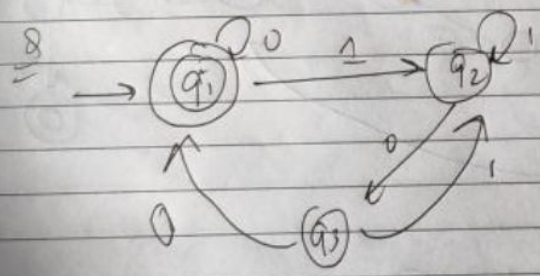
Arden's Algebraic Method

$$V_1 = V_1\alpha_{11} + V_2\alpha_{21} + V_3\alpha_{31} \ldots + \epsilon$$

$$V_2 = V_1\alpha_{12} + V_2\alpha_{22} + V_3\alpha_{32} \ldots$$

$$\vdots$$

$$V_n = V_1\alpha_{1n} + V_2\alpha_{2n} + V_3\alpha_{3n} \ldots$$



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

$$q_2 = q_1 1 + q_2 1 + q_3 1$$

$$q_3 = q_2 0$$

Good Write

$q_2 = q_1 1 + q_2 1 + q_2 01$

$q_2 = q_1 1 + q_2 (1+01)$

$R = Q \quad R \quad P$

$q_2 = q_1 1 (1+01)^*$

$q_3 = q_1 1 (1+01)^* 0$

$q_1 = q_1 0 + q_1 1 (1+01)^* 00 + \epsilon$

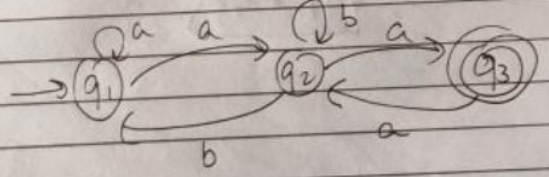
$q_1 = q_1 (0 + 1 (1+01)^* 00) + \epsilon$

$q_1 = \epsilon + q_1 (0 + 1 (1+01)^* 00)$

$R = Q + R \quad P$

$q_1 = \cancel{00} (0 + 1 (1+01)^* 00)^*$

Q



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

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

$q_3 = q_2 a$

Good Write

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

$q_2 = q_1 a + q_2 (b + aa)$

$q_2 = q_1 a (b + aa)^*$

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

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

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

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

$q_3 = (a + a (b + aa)^* b)^* a (b + aa)^* a$

More than One → q_2 & q_3

Good Write

DATE: / /
PAGE:

Pumping lemma for Regular languages

Pumping lemma used to prove that language is not regular but it cannot be used to prove that language is Regular.

Theorem

If L is regular language then L has corresponding FA with 'n' states such that any string 'w' where $|w| \geq n$ may be divided into three part $w = xyz$ such that following condition must be true

i) $xy^iz \in L$ for every $i \geq 0$

ii) $|y| > 0$

iii) $|xy| \leq n$

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

$L = \{a^p \mid p \text{ is any Prime Number}\}$

$L = \{aa, aaa, aaaaa \ldots\}$

Pattern

False → Not Regular.

True → Regular or other

Good Write

Pumping lemma used to prove that L is not regular.

i) language L is regular and FA accepted that language has n state.

ii) Choose $|w| \geq n$ and $w = xyz$ such that
$|y| > 0$
$|xy| \leq n$.

iii) find suitable integer 'i' to prove $xy^iz \notin L$.
(Contradiction)

Q Show that $L = \{a^p \mid p \text{ is prime}\}$ is not regular.
$L = \{aa, aaa, aaaaa \ldots\}$

i) L is regular and FA accepted that L has n state
$= 5$

ii) $w = aaaaaaa$.
$|w| = 7 > 5$ ✓

$w = xyz$

$w = \underbrace{a}_{x}\underbrace{aaaa}_{y}\underbrace{aa}_{z}$

iii) $xy^iz \in L$

$i = 1$

aa aaa aa

$i = 2$

aa aaaaa aa } $\notin L$.

By contradiction,
given language is not regular.

Good Write

DATE: / /
PAGE:

Q $L = \{0^i 1^i \mid i \geq 1\}$

(i) Assume L is reg. & $n = 5$

(ii) $w = 000111$
$|w| = 6 \geq 5$

$w = xyz$
$|y| > 0$
$|xy| \leq 5$

Case I
$x = 0$
$y = 00$
$z = 111$

Case II
$x = 000$
$y = 1$
$z = 11$

Case III
$x = 00$
$y = 01$
$z = 11$

(iii) $xy^iz \in L$

(i)
$0\,00\,111$
$0\,0000\,111 \notin L$

(ii)
$000\,1\,11$
$000\,11\,11 \notin L$

(iii)
$00\,01\,11$
$00\,0101\,11 \notin L$

$\therefore$ Not Regular.

Q Show that $L = \{ww \mid w \in \{a,b\}^*\}$ is not regular.

$L = \{a^n b a^n b$, . . . $w = a^n b$

(i) Assume L is regular FA = n state

$w = a^n b a^n b$

$|w| = 2n + 2 \geq n$

$w = xyz$
$|y| > 0$
$|xy| \leq n$

$w = \underbrace{a}_{x}\ \underbrace{\overset{s-1}{a}}_{y}\ \overset{n-s=n}{a}\ \underbrace{b a^n b}_{y}$

Good Write

(iii) $xy^i z \in L \quad i=1$

$a^s a^r a^{n-s-r} b \mid a^n b \in L$

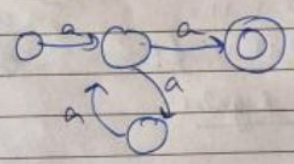
$a^s a^{(r)^2} a^{n-s-r} b \mid a^n b \quad i=2 \notin L$

Not regular

Regular lang

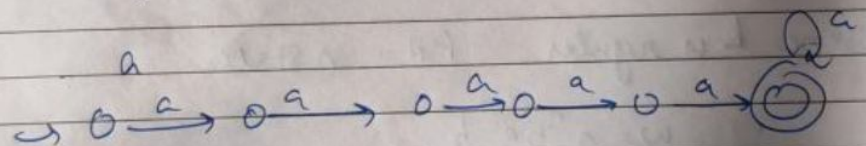
Q Is $L = \{a^{2n} \mid n \geq 1\}$ is regular.

$L = \{aa, aaaa, aaaaaa \ldots$



Q 1) $L = \{a^n \mid n > 4\}$

$L = \{aaaaa, aaaaaa \ldots$

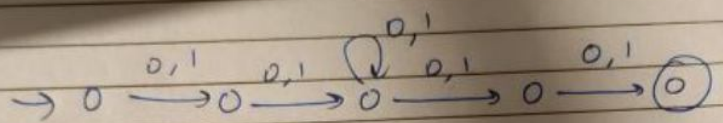


Regular.

DATE: __/__/____
PAGE: _____

a) $L = \{ vwv \mid v, w \in \{0,1\}^* \ |v| = 2 \}$

$L = \{ vvvv \ldots$



Regular ✓