Total No. of Pages 1 — *Roll No.* 2K20/CO/

V-SEMESTER — *B.Tech.(CO)*
MID SEMESTER EXAMINATION(Summer) — June-2023

CO-303 THEORY OF COMPUTATION

Time: 1:30 Hours — Max. Marks: 25

Note: Answer all questions. Assume suitable missing data, if any

---

Total No. of Pages 1 — Roll No. 2K20/

V-SEMESTER — B.Tech.(CO)
MID SEMESTER EXAMINATION — Sept- 2022

CO-303 THEORY OF COMPUTATION

Time: 1:30 Hours — Max. Marks: 25

Note: Answer all questions. Assume suitable missing data, if any

---

Total No. of Pages 1 — *Roll No.*..........

V-SEMESTER — *B.Tech.(COE)*
MID SEMESTER EXAMINATION — Sept-2019

CO-303 THEORY OF COMPUTATION

Time: 1:30 Hours — Max. Marks: 25

Note: Answer all questions. Assume suitable missing data, if any

---

Total No. of Pages 1 — *Roll No.*..........

V-SEMESTER — *B.Tech.(COE)*
MID SEMESTER EXAMINATION — Sept- 2018

CO 303 THEORY OF COMPUTATION

Time: 1:30 Hours — Max. Marks: 20

Note: Answer all questions. Assume suitable missing data, if any

---

Old

Total No. of Pages 2 — *Roll No.*..........

V-SEMESTER — *B.Tech.(CO) Old Scheme*
END SEMESTER EXAMINATION — Nov - 2018

CO 303 THEORY OF COMPUTATION

Time: 3:00 Hours — Max. Marks: 70

Note: Answer all questions by selecting any two parts from each question. Assume suitable missing data, if any

---

Total No. of Pages 2 — *Roll No.*..........

V-SEMESTER (old) — *B.Tech.(CO/SE)*
END SEMESTER EXAMINATION — Feb- 2018

CO/SE-303 THEORY OF COMPUTATION

Time: 3:00 Hours — Max. Marks: 70

Note: Answer all questions by selecting any two parts from each question. Assume suitable missing data, if any

---

Total No. of Pages 2 — *Roll No.*..........

V-SEMESTER — *B.Tech.(CO)*
END SEMESTER EXAMINATION — Nov- 2018

CO-303 THEORY OF COMPUTATION

Time: 3:00 Hours — Max. Marks: 50

Note: Answer all questions by selecting any two parts from each question. Assume suitable missing data, if any

---

Total No. of Pages 2 — *Roll No.*..........

V-SEMESTER — *B.Tech.(COE)OLD*
END SEMESTER EXAMINATION — Nov/Dec- 2019

CO-303 THEORY OF COMPUTATION

Time: 3:00 Hours — Max. Marks: 70

Note: Answer all questions by selecting any two parts from each question. Assume suitable missing data, if any

---

Total No. of Pages 2 — Roll No. 2K20/CO/

V-SEMESTER — B.Tech.(CO)
END SEMESTER EXAMINATION — Nov- 2022

CO303 THEORY OF COMPUTATION

Time: 3:00 Hours — Max. Marks: 50

Note: Attempt any five questions. Assume suitable missing data, if any

---

Total No. of Pages 2 — Roll No..........

V-SEMESTER (SUPPLY) — B.Tech.(CO/SE)
END SEMESTER EXAMINATION — Feb- 2018

CO/SE-303 THEORY OF COMPUTATION

Time: 3:00 Hours — Max. Marks: 70

Note: Answer all questions by selecting any two parts from each question. Assume suitable missing data, if any

---

Total No. of Pages 2 — Roll No..........

V-SEMESTER — B.Tech.(COE)
END SEMESTER EXAMINATION — Nov/Dec-2019

CO-303 THEORY OF COMPUTATION

Time: 3:00 Hours — Max. Marks: 50

Note: Answer all questions by selecting any two parts from each question. Assume suitable missing data, if any

---

Total No. of Pages: 2 — **B.TECH (COE)**

FIFTH SEMESTER
SUPPLIMENTARY EXAMINATION (FEBRUARY 2018)

CO-303 THEORY OF COMPUTATION

Time: 3 hours — Max. Marks: 50

Note: Attempt any five questions.
Assume suitable missing data, if any.

Prepared by Madhav Gupta (2K21/CO/262)

Date : / /
Page No.

UNIT-1

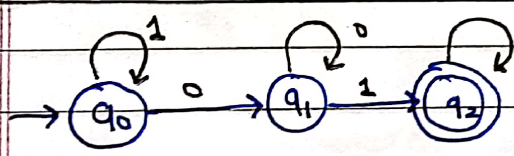
Q.No. 1 2X3=06

A) Differentiate between DFA & NDFA with suitable example and construct a DFA that accepts strings over {0, 1} those have exactly two zero's anywhere. 2018M

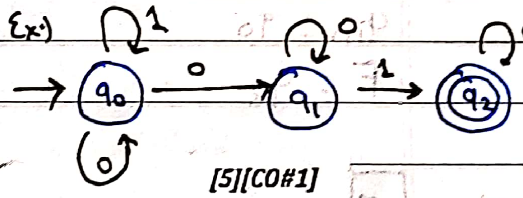
DIFFERENCE B/W DFA and NDFA

DETERMINISTIC FINITE AUTOMATA	NON-DETERMINISTIC FINITE AUTOMATA
→ A machine that can exist in only one state at a time	A machine that can exist in more than one state as we have more possi
→ The transition from a state is to a single particular next state for each input symbol. Hence, its deterministic	→ The transition from a state can be to multiple next states for each input symbol. Hence, its non-deterministic.
→ Empty strings are not seen and back tracking is allowed	→ Empty string transitions are permitted and backtracking isnt possible
→ Simple and easy to decide, one choice	→ next state can be chosen at random paralelly.
→ Requires more space	Requires less space
transition function → $\delta = Q \times \Sigma \rightarrow Q$	$\delta = Q \times \Sigma \rightarrow 2^Q$

Ex.)



Ex.)



Q.No. 1 [5][CO#1]

Construct a Finite Automata(FA) accepting the set of strings over $\Sigma = \{0,1\}$ having exactly two 0's anywhere. 2023M

Q.No. 1 A) Construct a DFSA for strings over {0, 1} which accepts strings those have exactly two zero's anywhere. 2X7=14 2018E (old)

DFA

$M = \{Q, \Sigma, \delta, q_{in}, F\}$

$Q : \{S, q_0, q_1\}$

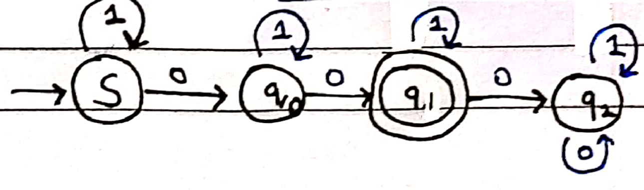
$\Sigma : \{0, 1\}$

$\delta : Q \times \Sigma \rightarrow Q$

$q_{in} : \{S\}$

$F : \{q_1\}$

$Q \backslash \Sigma$	0	1
→ S	q_0	S
q_0	q_1	q_0
q_1	q_2	q_1
q_2	q_2	q_2



Q.No. 1

A) Construct a Finite Automata(FA) for accepting strings of 0's and 1's that contain equal numbers of 0's and 1's,and no prefix of the string should contain two more 0's than 1's or two more 1's than 0's. 2019E (old)

Q.No. 1 2X4=08

A) Construct a Finite Automata(FA) for accepting strings of 0's and 1's that contain equal numbers of 0's and 1's,and no prefix of the string should contain two more 0's than 1's or two more 1's than 0's. 2019M

$M = \{Q, \Sigma, \delta, q_{in}, F\}$

$Q : \{q_0, q_1, q_2\}$

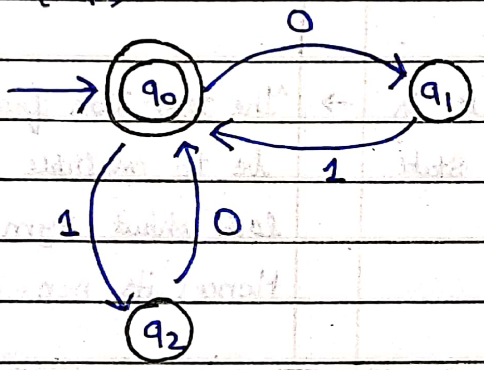
$\Sigma : \{0, 1\}$

$\delta : Q \times \Sigma \to Q$

$q_{in} : \{q_0\}$

$F : \{q_0\}$

Q\Σ	0	1
→ q_0	q_1	q_2
q_1	–	q_0
q_2	q_0	–



Q.No. 1 [5x2=10]

A. Construct a Nondeterministic Finite Automata(NDFA) accepting the set of strings over $\Sigma = \{a, b\}$ ending in "aba" use it to construct a DFA accepting same set of string. 2022E [CO#1]

NFA

$M_1 = \{Q, \Sigma, \delta, q_{in}, F\}$ #State = 3+1 = 4

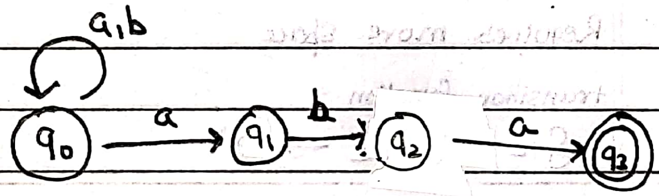
$Q : \{q_0, q_1, q_2, q_3\}$

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

$\delta : Q \times \Sigma \to 2^Q$

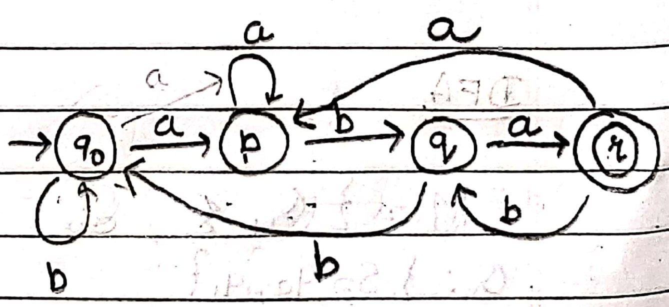
$q_{in} : q_0$

$F : q_3$



DFA

Q\Σ	a	b
→ q_0	$[q_0, q_1]$ p	q_0
p = $[q_0, q_1]$	$[q_0, q_1]$ p	$[q_0, q_2]$ q
q = $[q_0, q_2]$	$[q_0, q_1, q_3]$ r	q_0
r = $[q_0, q_1, q_3]$	$[q_0, q_1]$ p	$[q_0, q_2]$ q



Date : / /
Page No.

Q.No. 1 [5][CO#1]

Construct a Finite Automata(FA) accepting the set of strings over Σ= {0,1} having 101 or 110 as a sub-string. 2022M

$M = \{Q, \Sigma, \delta, q_{in}, F\}$

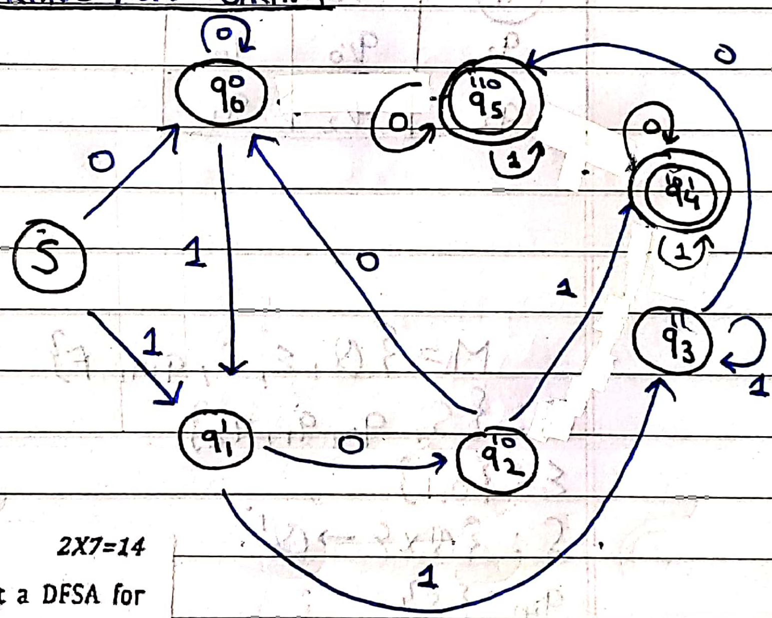
$Q: \{S, q_0, q_1, q_2, q_3, q_4, q_5\}$
$\Sigma: \{0,1\}$
$\delta: Q \times \Sigma \rightarrow Q'$
$q_{in}: \{S\}$
$F: \{q_4, q_5\}$

ending at '0' → q_0
ending at '1' → q_1
ending at '10' → q_2
ending at '11' → q_3
ending at '101' → q_4
ending at '110' → q_5

TRANSITION-TABLE

Q \ Σ	0	1
→ S	q_0	q_1
q_0	q_0	q_1
q_1	q_2	q_3
q_2	q_0	q_4
q_3	q_5	q_3
q_4	q_4	q_4
q_5	q_5	q_5

TRANSITION GRAPH



Q.No. 1 2X7=14

A) What are the limitations of Finite Automata? Construct a DFSA for strings over {0, 1} having an even number of 1's and odd no of 0's. 2018E (old)

LIMITATIONS

Limitations of Finite Automata include:

→ Inability to understand non-regular languages
→ Lack of memory to remember past states
→ Limited expressive power compared to turing machines.

Q.No. 1 2X7=14

A) What are the limitations of Finite Automata? Construct a DFSA for strings over {0, 1} having an even number of 1's and odd no of 0's. 2018E (old)

Date : / /
Page No.

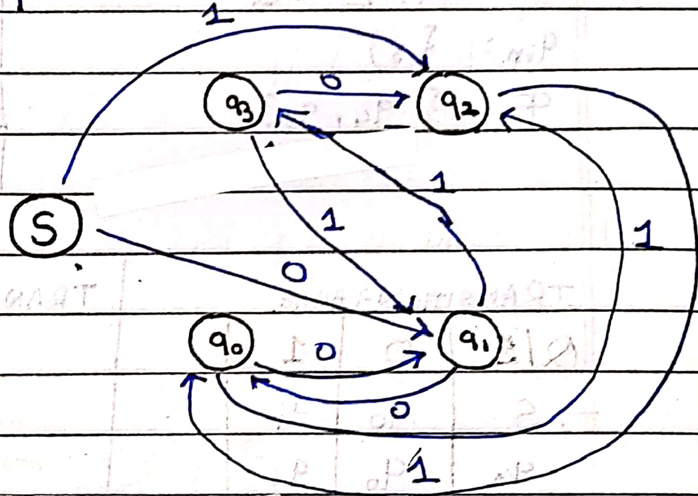
DFA (even 1s, odd 0s)

$M = \{Q, \Sigma, \delta, q_{in}, F\}$

$Q : \{S, q_0, q_1, q_2, q_3\}$
$\Sigma : \{0, 1\}$
$\delta : Q \times \Sigma \rightarrow Q'$
$q_{in} : \{S\}$
$F : \{q_1\}$

even 1s even 0s $\rightarrow q_0$
even 1s odd 0s $\rightarrow q_1$
odd 1s even 0s $\rightarrow q_2$
odd 1s odd 0s $\rightarrow q_3$

$Q \backslash \Sigma$	0	1
$\rightarrow S$	q_1	q_2
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

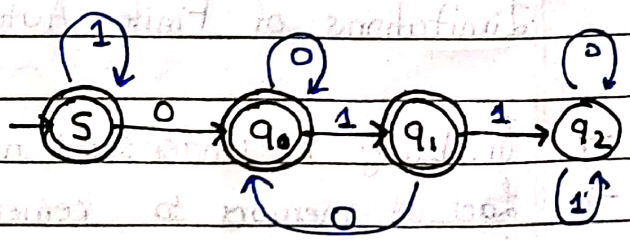


Q.No. 1 2X5=10

A) Construct a Deterministic Finite Automata (DFA) that accepts all possible strings of 0's and 1's that do not contain 011 as a substring. 2019 E

$M = \{Q, \Sigma, \delta, q_{in}, F\}$
$Q : \{S, q_0, q_1, q_2\}$
$\Sigma : \{0, 1\}$
$\delta : \{Q \times \Sigma \rightarrow Q'$
$q_{in} : \{S\}$
$F : \{S, q_0, q_1\}$

$Q \backslash \Sigma$	0	1
$\rightarrow S$	q_0	S
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_2



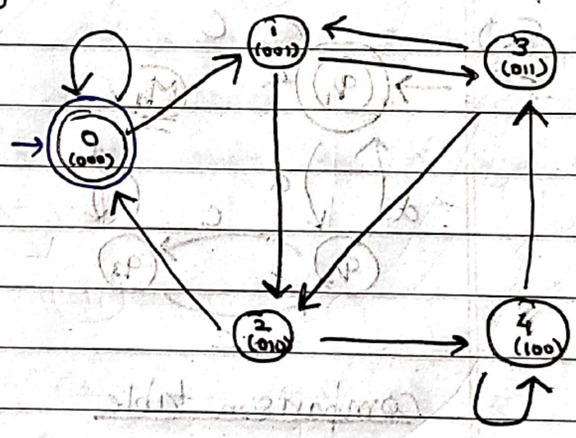
Date : / /
Page No.

[b] Design DFA for the language over 0 and 1 which accepts the binary strings whose decimal equivalent is divisible by 5? 2018(old)

DFA, $M = \{Q, \Sigma, \delta, q_{in}, F\}$

$Q : \{0, 1, 2, 3, 4\}$
$\Sigma : \{0, 1\}$
$\delta : Q \times \Sigma \to Q'$
$q_{in} : \{0\}$
$F : \{0\}$

δ



Date : / /

Page No.

8. What is myhill-Nerode theorem? Explain equivalence of two finite automata(FA) with suitable example. 2022 E [CO#1]

Equivalence of FA

ii. Equivalence of two finite automata

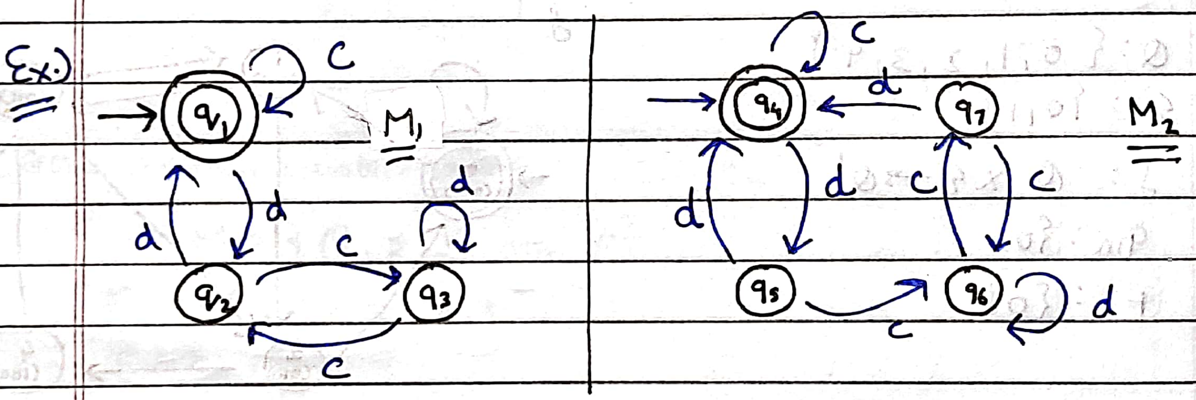
Two finite automata over Σ are equivalent if they accept the same set of string over Σ (i.e. perform the same task).

Steps:

i) for any pair of states $\{q_i, q_j\}$ the transition for input $a \in \Sigma$ is defined by $\{q_a, q_b\}$ where $\delta\{q_i, a\} = q_a$ and $\delta\{q_j, a\} = q_b$.
The two automata are not equivalent if for the pair $\{q_a, q_b\}$ one is intermediate state and the other is final state.

ii) If initial state is final state of one automaton, then in second automation also, initial state must be final state for them to be equivalent.

Ex.)



comparison table

STATES	c	d
(q_1, q_4)	(q_1^F, q_4^F)	(q_2^I, q_5^I)
(q_2, q_5)	(q_3^I, q_6^I)	(q_1^F, q_4^F)
(q_3, q_6)	(q_2^I, q_7^I)	(q_3^I, q_6^I)
(q_2, q_7)	(q_3^I, q_6^I)	(q_1^F, q_4^F)

Since FS corr with FS
and IS —— IS
M_1 and M_2 are equivalent

Date : / /
Page No.

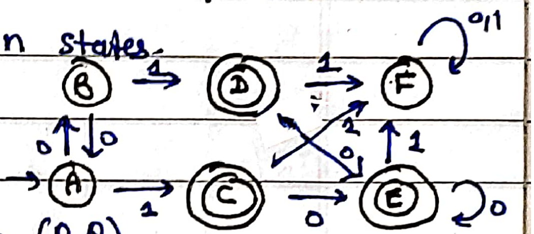
ii. Myhill-Nerode Theorem. 2018E
C) Explain following terms with example
i. Myhill-Nerode Theorem. 2018E

MYHILL-NERODE'S THEOREM

ii. Myhill-Nerode Theorem. 2018S

Myhill Nerode's theorem is a fundamental result which is used to prove whether or not a language is regular and it is also used for minimization of states in DFA.

$\therefore$ A language is regular if and only if $\equiv_L$ partitions Σ^* into finitely many equivalence classes. If $\equiv_L$ partitions Σ^* into n equivalence classes, then a minimal DFA recognizing L has exactly n states.

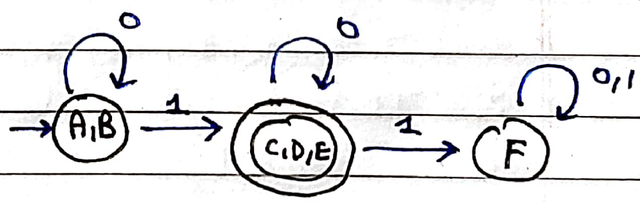


For minimization Steps

1.) Draw a table for all pairs of states (P, Q)
2.) Mark all pairs where P∈F and Q∉F.
3.) If there are any unmarked pairs (P, Q) such that [δ(P, x), δ(Q, x)] is marked, then mark [P, Q] where 'x' is an input symbol (Repeat till no more markings can be made)
4.) Combine all the unmarked pairs and make them a single state in minimized DFA.

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

unmarked pairs : (A, B) (D, C) (E, C) (E, D)



Date : / /

Page No.

[b] Minimize the DFA whose transition-function table is given below (the start state is indicated by ↓; and accept state is indicated by *

	↓1	2	3	*4	5	6	7	*8
a	2	5	4	8	2	1	3	8
b	7	3	3	5	3	6	2	2

2018S (old)

4. Draw the Non Deterministic Push Down Automata for Palindrome Language. 2018S (10 marks)

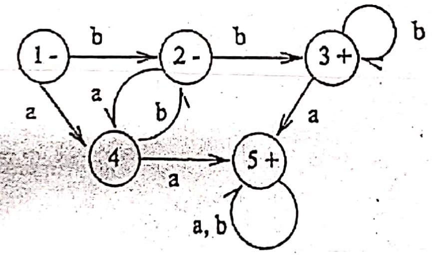
Date : / /

Page No.

5. Let L1 be the language of all the strings that don't contain substring aa and L2 be all the strings with odd no of letter. Find L1.L2 using Kleen's Theorem. (10 marks)

2019S

2. Derive the Regular Expression from the following NFA using Kleen's Theorem (10 marks)



2018S

$(a+b)^* = \{a, b, ab, bba, \ldots\}$

Date: $0' = a'$
Page No. = 1

C) Design a Mealy machine which reads the input from $(a + b)^*$ and produces following outputs:
i. If input ends at "aba", output is A
ii. If input ends at "aab", output is B
iii. Otherwise output is C

2018 E

UNIT - 2

2X7=14

Q.No. 3

A) Design a Mealy machine which reads the input from $(a + b)^*$ and produces following outputs:
i. If input ends at "bba", output is A
ii. If input ends at "aab", output is B
iii. Otherwise output is C

2019E (old)

Q.No. 2 [5][CO#1]

Design Moore and Mealy machine for binary input sequence if it ends in 110, output is 'X' if it ends in '101' output is 'Y', otherwise 'Z'.

2023M

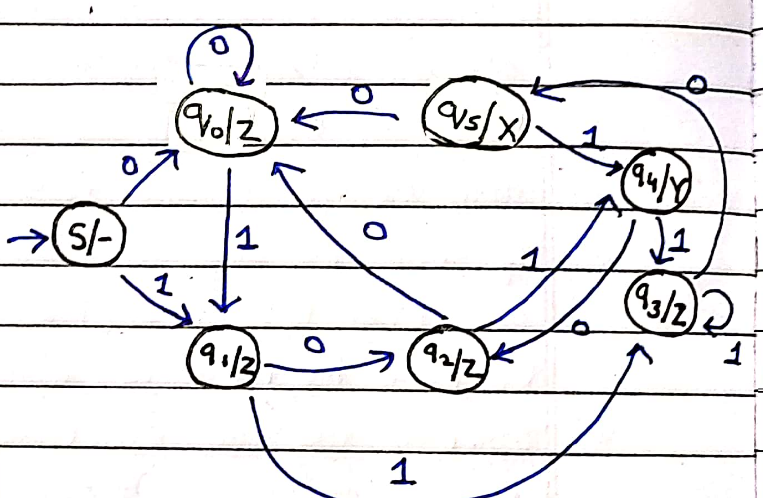
A = Y
B = X
C = Z

Moore Machine

$M_1 = \{Q, \Sigma, \Delta, \delta, \lambda, q_{in}\}$
$Q: \{S, q_0, q_1, q_2, q_3, q_4, q_5\}$
$\Sigma: \{0, 1\}$
$\Delta: \{X, Y, Z\}$
$\delta: Q \times \Sigma \rightarrow Q$
$\lambda: Q \rightarrow \Delta$
$q_{in}: \{S\}$

States
$q_0 \rightarrow$ ending at 0
$q_1 \rightarrow$ ending at 1
$q_2 \rightarrow$ ending at 10
$q_3 \rightarrow$ ending at 11
$q_4 \rightarrow$ ending at 101
$q_5 \rightarrow$ ending at 110

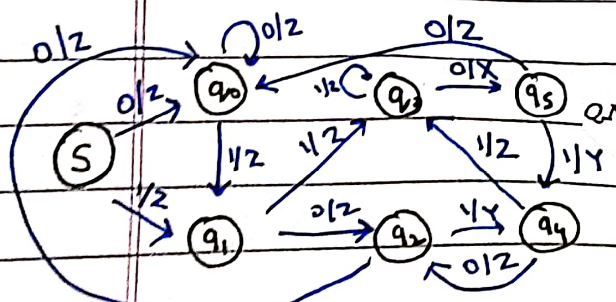
	Q/Σ	0	1	Δ
initial	S	q_0	q_1	–
ending at '0'	q_0	q_0	q_1	Z
'1'	q_1	q_2	q_3	Z
'10'	q_2	q_0	q_4	Z
'11'	q_3	q_5	q_3	Z
'101'	q_4	q_2	q_3	Y
'110'	q_5	q_0	q_4	X



Mealy Machine

$M_2 = \{Q, \Sigma, \Delta, \delta, \lambda, q_{in}\}$
$Q: \{S, q_0, q_1, q_2, q_3, q_4, q_5\}$
$\Sigma: \{0, 1\}$
$\Delta: \{X, Y, Z\}$
$\delta: Q \times \Sigma \rightarrow Q$
$\lambda: Q \times \Sigma \rightarrow \Delta$
$q_{in}: \{S\}$

Q/Σ	0	o/p	1	o/p
S	q_0	Z	q_1	Z
q_0	q_0	Z	q_1	Z
q_1	q_2	Z	q_3	Z
q_2	q_0	Z	q_4	Y
q_3	q_5	X	q_3	Z
q_4	q_2	Z	q_3	Z
q_5	q_2	Z	q_4	Y



Q.No. 5 2X5=10

A) Design a Mealy machine which reads the input from $(0 + 1)^*$ and produces following outputs:
i. If input ends at 101, output is A
ii. If input ends at 110, output is B
iii. Otherwise output is C

2019E

Q.No. 3

A) What is difference between moore and mealy machine? Construct a Moore Machine equivalent to given Mealy Machine as below 2019M

Present State	Next state	O/P	Next state	O/P
	Input=0		Input=1	
→q1	q2	A	q3	A
q2	q2	B	q3	A
q3	q2	A	q3	B

C) Explain following terms:
i. Moore and Mealy machine 2018G

Q.No. 2

2X7=14

A) What is difference between moore and mealy machine? Construct a Moore Machine equivalent to given Mealy Machine as below 2019E(old)

Present State	Next state	O/P	Next state	O/P
	Input=0		Input=1	
→q1	q2	A	q3	A
q2	q2	B	q3	A
q3	q2	A	q3	B

C) Explain following terms:
i. Moore and Mealy machine 2018S

MOORE MACHINE	MEALY MACHINE
→ The output depends only on the present State	→ The output depend on present state and present input.
→ It generally has more states than mealy	→ It generally has fewer states.
→ Length of o/p string is 1 more than that of input string (unless q_{in} is undefined)	→ Length of o/p string is equal to input string
output function ∃ $\lambda : Q \to \Delta$	output function ∃ $\lambda : Q \times \Sigma \to \Delta$

Mealy Machine to Moore Machine

new states are: $q_1 \to q_1$

$q_2 \to q_{2A}, q_{2B}$

$q_3 \to q_{3A}, q_{3B}$

∴ State table can be written for Mealy Machine as:

PS	NS	OP	NS	OP
q_1	q_{2A}	A	q_{3A}	A
q_{2A}	q_{2B}	B	q_{3A}	A
q_{2B}	q_{2B}	B	q_{3A}	A
q_{3A}	q_{2A}	A	q_{3B}	B
q_{3B}	q_{2A}	A	q_{3B}	B

Thus, the moore machine can be represented by the TT as.

Present State	0	1	Output
→ q_1	q_{2A}	q_{3A}	—
q_{2A}	q_{2B}	q_{3A}	A
q_{2B}	q_{2B}	q_{3A}	B
q_{3A}	q_{2A}	q_{3B}	A
q_{3B}	q_{2A}	q_{3B}	B

B) Explain equivalence of two Finite automata(FA) with example and Construct a Moore Machine equivalent to given Mealy Machine as below 2018M

Present state	Next state	O/P	Next state	O/P
	Input=0		Input=1	
→S1	S2	Z1	S3	Z1
S2	S2	Z2	S3	Z1
S3	S2	Z1	S3	Z2

Date : / /

Page No.

B) What is difference between moore and mealy machine? Construct a Moore Machine equivalent to given Mealy Machine as below

2019 E

Present State	Next state Input=0	O/P	Next state Input=1	O/P
→q0	q0	1	q1	0
q1	q3	1	q3	1
q2	q1	1	q2	1
q3	q2	0	q0	1

new states:

$q_0 \to q_0$

$q_1 \to q_{11}, q_{10}$

$q_2 \to q_{20}, q_{21}$

$q_3 \to q_3$

Mealy

Q\Σ	NS 0	NS 1	O/P
q_0	q_0	q_{10}	1
q_{10}	q_3	q_3	0
q_{11}	q_3	q_3	1
q_{20}	q_{11}	q_{21}	0
q_{21}	q_{11}	q_{21}	1
q_3	q_{20}	q_0	1

Q.No. 2

2X7=14

A) Construct a Moore Machine equivalent to given Mealy Machine as below

2018 E

Present State	Next state Input=a	O/P	Next state Input=b	O/P
→q0	q3	a	q1	b
q1	q1	b	q2	a
q2	q2	a	q3	a
q3	q3	a	q3	a

new States are:

$q_0 \to q_0$

$q_1 \to q_1$

$q_2 \to q_2$

$q_3 \to q_3$

PS	NS (i/p = a)	NS (i/p = b)	o/p (Δ)
→ q_0	q_3	q_1	—
q_1	q_1	q_2	b
q_2	q_2	q_3	a
q_3	q_3	q_3	a

Date : / /
Page No.

Q.No. 2 [5x2=10]

A. Design Moore and Mealy machine that will read sequence made up of letters a, e, i, o, u and will give as output same characters except when an 'i' is followed by 'e' it will be changed to 'u'. 2022E [CO#3]

Design moore machine that will read sequence made up of letter a,e,i,o,u and will give as output same characters except when an 'i' is followed by 'e', it will be changed to 'u'. also design mealy machine for the same. 2018E

Moore Machine

$M_1 = \{Q, \Sigma, \Delta, \delta, \lambda, q_{in}\}$

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

$\Sigma : \{a, e, i, o, u\}$

$\Delta : \{a, e, i, o, u\}$

$\delta : Q \times \Sigma \rightarrow Q$

$\lambda : Q \rightarrow \Delta$

q_{in} :

last $a \rightarrow q_0$
— $e \rightarrow q_1$
— $i \rightarrow q_2$
— $o \rightarrow q_3$
— $u \rightarrow q_4$.

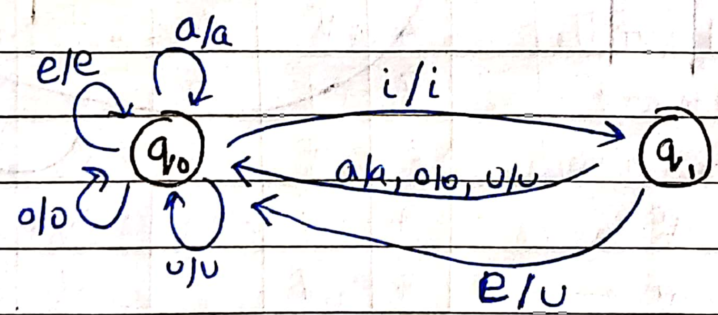
transition state table :

Q \ Σ	a	e	i	o	u	Δ
q_0	q_0	q_1	q_2	q_3	q_4	a
q_1	q_0	q_1	q_2	q_3	q_4	e
q_2	q_0	q_4	q_2	q_3	q_4	i
q_3	q_0	q_1	q_2	q_3	q_4	o
q_4	q_0	q_1	q_2	q_3	q_4	u

diagram not needed

Mealy Machine

for mealy machine, we concentrate on states and we find that there will be only two states, one which recognizes 'i' and other which cant be reached on i.



$q_0 \rightarrow$ reached on a, e, o, u

$q_1 \rightarrow$ reached on i

No. 2 2X3=06

A) Design a Mealy machine which reads the input from $(0 + 1)^*$ and produces residue mod-4 for each binary string treated as binary integer. 2018M

moore

$M_1 = \{Q, \Sigma, \Delta, \delta, \lambda, q_{in}\}$

$Q = \{$

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

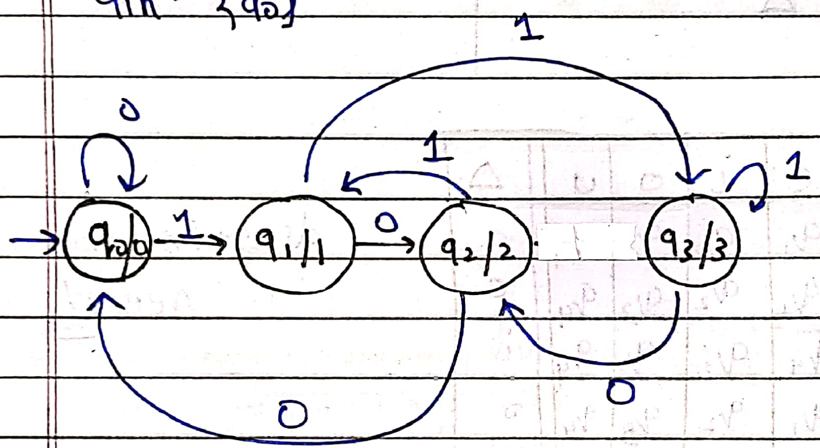
$\Delta = \{0, 1, 2, 3\}$

$\delta = Q \times \Sigma \rightarrow Q$

$\lambda = Q \rightarrow \Delta$

$q_{in} = \{q_0\}$

0 → 0000
1 → 0001
2 → 0010
3 → 0011
4 → 0100
5 → 0101
6 → 0110
7 → 0111
8 → 1000
9 → 1001
10 → 1010



Mealy (convert)

moore

$Q \setminus \Sigma$	0	1	Δ
q_0	q_0	q_1	0
q_1	q_2	q_3	1
q_2	q_0	q_1	2
q_3	q_2	q_3	3

→

Mealy

$Q \setminus \Sigma$	0	o/p	1	o/p
q_0	q_0	0	q_1	1
q_1	q_2	2	q_3	3
q_2	q_0	0	q_1	1
q_3	q_2	2	q_3	3

Date : / /

Page No.

B. Explain decision properties of regular languages and Construct Moore machine equivalent to given mealy as below [CO#3]

2022E

Present state	Next state (Input=a)	O/P	Next state (Input=b)	O/P
→A	C	X	B	X
B	A	Y	D	X
C	B	Y	A	Y
D	D	Y	C	X

Decision properties of regular languages: are the ones who can be ~~computed~~ decided using a finite automaton. We use machine model to proof ~~decidable~~ decision properties:

i.) Emptiness and non Emptiness

Step 1:) select the state that can't be reached from initial state and del them

step 2:) if machine contains atleast one final states, so then FA accepts non emp. lang

step 3:) if resulting machine is free from final state, then FA accepts empty lang.

ii.) Finiteness and Infiniteness

step 1:) select and remove unreachable states (")

step 2:) select and remove the states from which we can't reach final st. (dead states)

step 3:) If ~~FA~~ mach. contains loops or cycles then FA accepts infinite lang

step 4) If machine don't contain loops or cycles then FA accepts finite lang.

iii.) Membership

Membership is a property to verify an arbitrary string is accepted by a FA or not. (i.e. its a member of the lang. or not).

iv) Equality

Two finite state automata M_1 & M_2 are said to be equal iff they accept the same language.

∴ Minimize the FSA and the minimal DFA will be unique.

new states

$A \to A\ (Y)$

$B \to B_X(X),\ B_Y(Y)$

$C \to C\ (X)$

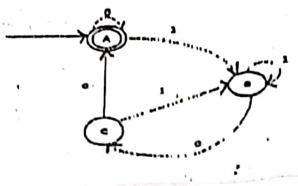
$D \to D_X(X),\ D_Y(Y)$

$S \to$ new initial state

PS	a	b	Δ
A	C	B_X	Y
B_X	A	D_X	X
B_Y	A	D_X	Y
C	B_Y	A	X
D_X	D_Y	C	X
D_Y	D_Y	C	Y
→ S	C	B_X	–

B) What is Arden's theorem? Construct a regular expression(RE) corresponding to the following FA using Arden's theorem

2019 M

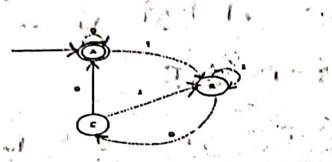


[5][CO#1]

Q.No. 3

A) What is Arden's theorem? Find a regular expression(RE) corresponding to the following FA using Arden's theorem

2022M



Date : / /

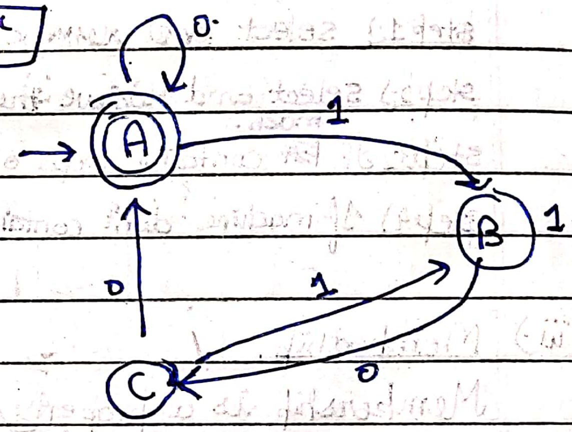
Page No.

Arden's theorem is used to find out the regular expression of a finite automaton.

Arden's theorem states:

"If P and Q are two regular expressions over Σ, and if P does not contain ϵ, then the following equation in R is given by $R = Q + RP$ has a unique solution, i.e.

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



$A = A.0 + C.0 + \epsilon$ — ①

$B = A.1 + B.1 + C.1$ — ②

$C = B.0$ — ③

⇒ find value of A

put C in B

$\Rightarrow B = A.1 + B.1 + B.0.1$

$\Rightarrow B = \underset{R = Q}{A.1} + \underbrace{B(1+0.1)}_{R.P}$

$\Rightarrow B = A.1\,(1+0.1)^*$ — ④

put ④ in ①

$A = A.0 + B.1.1 + \epsilon$

$\Rightarrow A = A.0 + A.1.(1+0.1)^* 1.1 + \epsilon$

$\Rightarrow \underset{R}{A} = \underset{Q}{\epsilon} + \underset{R}{A}\,\underset{P}{(0 + 1(1+0.1)^* 1.1)}$

$\Rightarrow A = \epsilon\,[0 + 1.(1+0.1)^* 1.1]^*$

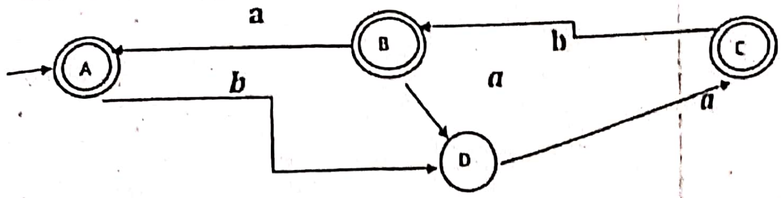
$$\Rightarrow \boxed{A = [0 + 1.(1+0.1)^* 1.1]^*}$$ R E.

Prepared by Madhav Gupta (2K21/CO/262)

Date : / /
Page No.

Q.No. 3 [5x2=10]

A. What is Arden's theorem? Find a regular expression(RE) corresponding to the following FA using Arden's theorem. 2022E [CO#6]



RE = A+B+C

here,

$A = B.a + \Lambda$ — (1)

$B = C.b$ — (2)

$C = D.a$ — (3)

$D = A.b + B.a$ — (4)

from 3,4

$C = A.ba + B.a.a$ — (5)

from 2,5

$B = A.bab + B.aab$ — (6)

put in (1)

$A = A\,baba + B\,aabb + \Lambda$

$= (\Lambda + Baabb) + A\,(baba)$

use arden

$A = (\Lambda + B\,aabb)\,((baba))^*$ — (8)

put in (6) and arden

$B = (\Lambda + Baabb)(baba)^*\,bab\,(aab)^*$ — (7)

put (7) (8) in (5)

$C = (\Lambda + Baabb)\,(baba)^*\,b + (\Lambda + Baabb)(baba)(aab)^*.a$

$C = (\Lambda + Babb)\,(baba)^*\,(b + (aab)^*.a)$ — (9)

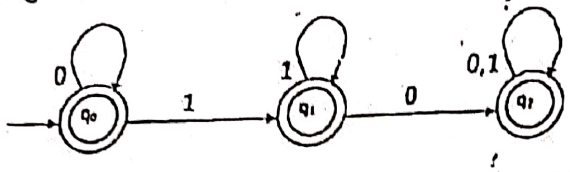
from 7,8,9

$\therefore$ RE = A+B+C

Date : / /
Page No.

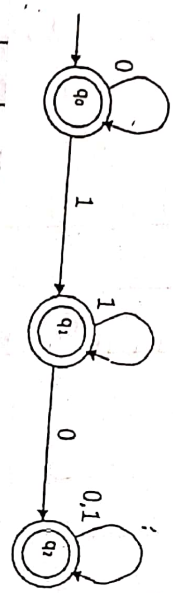
C) Construct a regular expression(RE) corresponding to the following FA using Arden's theorem

2018 E(old)



C) Construct a regular expression(RE) corresponding to the following FA using Arden's theorem

2018 S



here

$\therefore\ q_0 = q_0 \cdot 0 + \epsilon$ — (1)

$\therefore\ q_1 = q_0 \cdot 1 + q_1 \cdot 1$ — (2)

$\therefore\ q_2 = q_1 \cdot 0 + q_2 (0+1)$ — (3) (~~Arden~~)

using arden on (1)

$q_0 = \epsilon 0^* = 0^*$ — (4)

So put in (2)

$q_1 = 0^* \cdot 1 + q_1 \cdot 1$

$\Rightarrow q_1 = (0^* 1) \cdot 1^*$ — (5)

put in (3)

$q_2 = (q_1 \cdot 0) \cdot (0+1)^*$

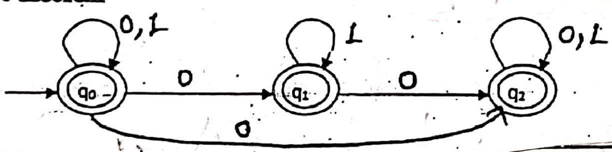
$= ((0^* 1) 1^* 0) \cdot (0+1)$

$q_2 = ((0^* 1) 1^* \cdot 0)(0+1)$

$\therefore\ RE = q_0 + q_1 + q_2 = 0^* + (0^* 1) \cdot 1^* + ((0^* 1) \cdot 1^* \cdot 0)(0+1)$

C) Construct a regular expression(RE) corresponding to the following FA using Arden's theorem

2019E



here

$q_0 = q_0 \cdot 0 + q_0 \cdot 1 + \Lambda$ — (1)

$q_1 = q_0 \cdot 0 + q_1 \cdot 1$ — (2)

$q_2 = q_0 \cdot 0 + q_1 \cdot 0 + q_2 \cdot 0 + q_2 \cdot 1$ — (3)

use arden on (1) $q_0 = \Lambda + q_0(0+1) \Rightarrow q_0 = \Lambda (0+1)^*$ — (4)

put (4) in (2) : $q_1 = \Lambda (0+1)^* \cdot 0 + q_1 (1)$ — (5)

~~put~~ use arden on 5 : $q_1 = \Lambda (0+1)^* \cdot 0 (1)^*$ — (6)

put (4)(6) in (3) $q_2 = \Lambda (0+1)^* \cdot 0 + \Lambda (0+1)^* 0 (1)^* \cdot 0 + q_2 (0+1)$

use arden $q_2 = \Lambda (0+1)^* \cdot 0 (\Lambda + (1)^* 0)(0+1)^*$ — (7)

using (4), (6), (7)

$RE = q_0 + q_1 + q_2$

$RE = \Lambda (0+1)^* + \Lambda (0+1)^* 0 (1)^* + \Lambda (0+1)^* \cdot 0 (\Lambda + 1^* 0)(0+1)^*$

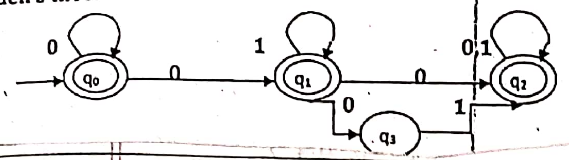
$= \Lambda (0+1)^* (\Lambda + 0 (1)^* + (0 (\Lambda + 1^* 0)(0+1)^*))$

B) Construct a regular expression(RE) corresponding to the following FA using Arden's theorem

2018 E(old)

Date : / /

Page No.



here,

$q_0 = \epsilon + q_0 \cdot 0$ —(1)

$q_1 = q_0 \cdot 0 + q_1 \cdot 1$ —(2)

$q_2 = q_1 \cdot 0 + q_2 \cdot (0+1) + q_3 \cdot (1)$ —(3)

$q_3 = q_1 \cdot 0$ —(4)

using arden in (1); $q_0 = \Lambda (0)^*$ —(5)

using (5) in (2); $q_1 = \Lambda (0)^+ + q_1 \cdot 1$

⇒ using arden $q_1 = \Lambda (0)^+ 1^*$ —(6)

using (6) in (4); $q_3 = \Lambda (0)^+ 1^* (0)$ —(7)

using (6),(7) in (3); $q_2 = \Lambda (0)^+ 1^* 0 + \Lambda (0^+) 1^* (0) \cdot 1 + q_2 (0+1)$

⇒ $q_2 = \Lambda (0)^+ 1^* (0+1) + q_2 (0+1)$

using arden; $q_2 = \Lambda (0)^+ 1^* (0+1)^+$ —(8)

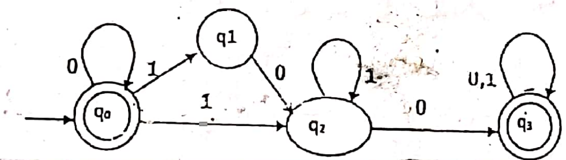
$RE = q_0 + q_1 + q_2 = \Lambda (0)^* + \Lambda (0)^+ 1^* + \Lambda (0)^+ 1^* (0+1)^+$

$= \Lambda 0^* [\epsilon + 01^* + 01^* (0+1)^+]$

$\boxed{RE = \Lambda 0^* [\epsilon + 01^* (0+1)^*]}$

B) Construct a regular expression(RE) corresponding to the following FA using Arden's theorem

2018 M



here,

$q_0 = q_0 \cdot 0 + \Lambda$ —(1)

$q_1 = q_0 \cdot 1$ —(2)

$q_2 = q_0 \cdot 1 + q_1 \cdot 0 + q_2 \cdot 1$ —(3)

$q_3 = q_2 \cdot 0 + q_3 (0+1)$ —(4)

use arden in (1)

$q_0 = \Lambda (0)^* \Rightarrow q_0 = 0^*$ —(5)

put (5) in (2)

$q_1 = 0^* 1$ —(6)

put (5)(6) in (2)

$q_2 = 0^* 1 + 0^* 1 \cdot 1 + q_2 \cdot 1 = 0^* 1 (\Lambda + 1) + q_2 \cdot 1$

use arden

$q_2 = 0^* 1 (\Lambda + 1)(1)^*$ —(7)

use (7) in (4) and arden

$q_3 = 0^* 1 (\Lambda + 1)(1)^* 0 (0+1)^*$ —(8)

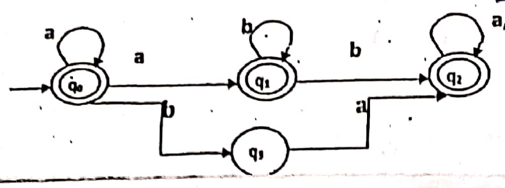
~~use~~

$RE = q_0 + q_3$

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

Date : / /
Page No.

B) What is Arden's theorem? Construct a regular expression(RE) corresponding to the following FA using Arden's theorem 2018 E



$q_0 = q_0 \cdot a + \Lambda$ -(1)

$q_1 = q_0 \cdot a + q_1 \cdot b$ -(2)

$q_2 = q_1 \cdot b + q_2(a+b) + q_3 \cdot a$ -(3)

$q_3 = q_0 \cdot b$ -(4)

put (4) in (3) $q_2 = q_1 \cdot b + q_2 \cdot (a+b) + q_0\, ba$ -(5)

use arden in (1) $q_0 = \Lambda a^* = a^*$ -(6)

put (6) in (2) $q_1 = a^* b + q_1 \cdot b$

use arden $q_1 = a^* b b^* = a^* b^+$ -(7)

$RE = q_0 + q_1 + q_2$

put (6)(7) in (5)

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

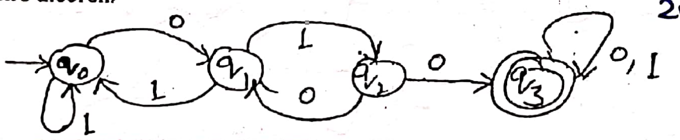
use arden

$$q_2 = [a^*b^+ba + (a^*b + a^*b^+b)] + q_2(a+b)$$

$$\Rightarrow q_2 = [a^*b^+ba + (a^*b + a^*b^+b)](a+b)^*$$

$$\therefore RE = a^*b^* + [a^*b^+ba + (a^*b)b^*][a+b]^*$$

C) Construct a regular expression(RE) corresponding to the following FA using Arden's theorem 2019E



$q_0 = q_0 \cdot 1 + q_1 \cdot 1 + \Lambda$ -(1)

$q_1 = q_0 \cdot 0 + q_2 \cdot 0$ -(2)

$q_2 = q_1 \cdot 1$ -(3)

$q_3 = q_2 \cdot 0 + q_3 \cdot 0 + q_3 \cdot 1$ -(4)

put (3) in (2) $q_1 = q_0 \cdot 0 + q_1 \cdot 1 \cdot 0$

use arden $\Rightarrow q_1 = q_0 \cdot 0\,(10)^*$ -(5) $\cancel{q_2 =}$

put (5) in (1) $\Rightarrow q_0 = q_0 \cdot 1 + q_0 \cdot 0(10)^* \cdot 1 + \Lambda = \Lambda + q_0(1 + 0(10)^* \cdot 1)$

use arden $\Rightarrow q_0 = \Lambda(1 + 0(10)^* \cdot 1)^*$ -(6)

put (6) in (5) $\Rightarrow q_1 = \Lambda(1 + 0(10)^* \cdot 1)^* \cdot 0(10)^*$ -(7)

put (7) in (3) $\Rightarrow q_2 = \Lambda(1 + 0(10)^* \cdot 1)^* \cdot 0(10)^* \cdot 1$ -(8)

put (8) in (4) $q_3 = \Lambda(1 + 0(10)^* \cdot 1)^* \cdot 0(10)^* \cdot 1 + q_3(0+1)$

use arden $q_3 = \Lambda(1 + 0(10)^* \cdot 1)^* \cdot 0(10)^* \cdot 1\,(0+1)^*$

RE $= q_3 =$

Date : / /
Page No.

B) What is Arden's theorem? Prove (b + aa*b) + (b + aa*b)(a + ba*b)* (a + ba*b) =a*b(a + ba*b)*.
2018E

LHS: $(b + aa^*b) + (b + aa^*b)(a + ba^*b)^*(a + ba^*b)$

$= (b + aa^*b)(\epsilon + (a + ba^*b)^*(a + ba^*b)^*)$ [$\because (P+Q)R = PR + QR$]

$= (b + aa^*b)(a + ba^*b)^*$ [$\because \epsilon + RR^* = R^* = \epsilon + R^*R$]

$= (\epsilon + aa^*)b(a + ba^*b)^*$ [$\because PR + QR = (P+Q)R$]

$= a^*b(a + ba^*b)^*$ [$\epsilon + RR^* = R^*$]

Hence Proved

Q.No. 3 [5][CO#1]

What is Arden's theorem? Prove that {a(a+b)*a+b(a+b)*b}* is a regular expression.
2023 SM

?

B) What is Arden's theorem? Prove (1 + 00*1) + (1 + 00*1)(0 + 10*1)* (0 + 10*1) =0*1(0 + 10*1)*.
2019E(old)

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

$= (1 + 00^*1)(\Lambda + (0 + 10^*1)^*(1 + 10^*1))$ [$PQ + PR = P(Q+R)$]

$= (1 + 00^*1)(\Lambda + (0 + 10^*1)^+)$ [$P^*P = P^+$]

$= (1 + 00^*1)(0 + 10^*1)^*$ [$\Lambda + P^+ = P^*$]

$= (\Lambda + 00^*).1(0 + 10^*1)^*$ [$PQ + RQ = (P+R)Q$]

$= (\Lambda + 0^+).1(0 + 10^*1)^*$ [$\Lambda + P^+ = P^*$]

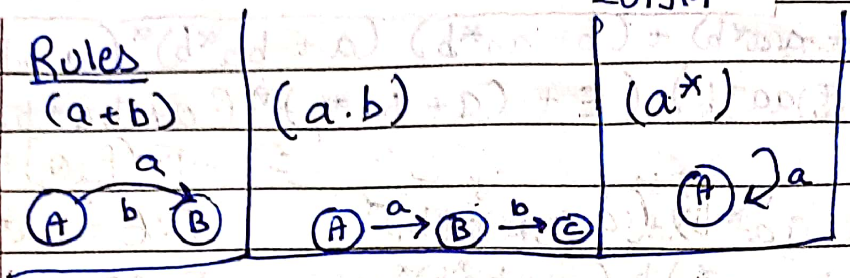
$= 0^*1(0 + 10^*1)^*$

Date : / /

Page No.

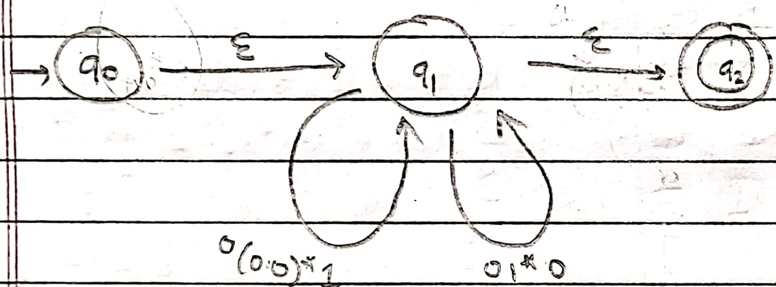
B) Construct FA for a regular expression $(0(00)^*1 + 01^*0)^*$ with and without null moves.

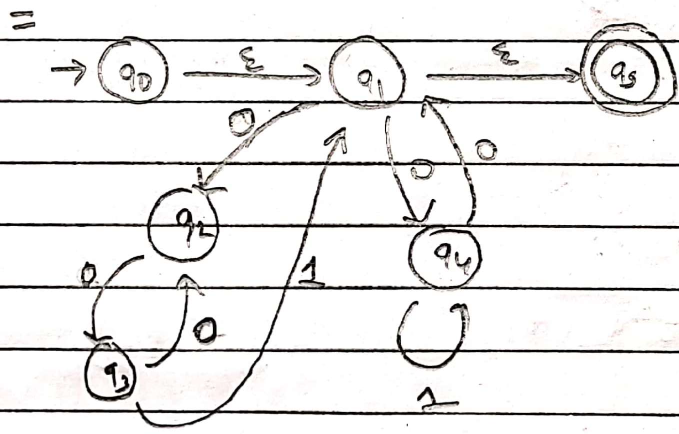
2019M



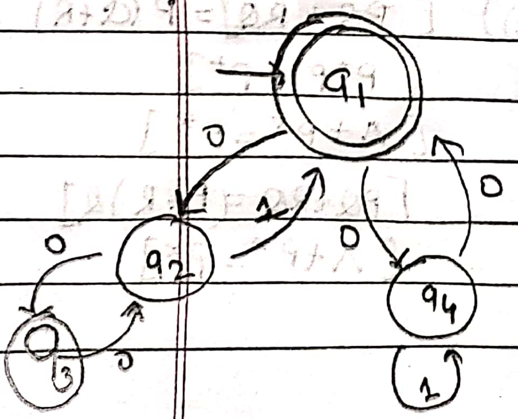
$$RE = (a(a.a)^*.b + ab^*a)^*$$

WITH NULL





WITHOUT NULL



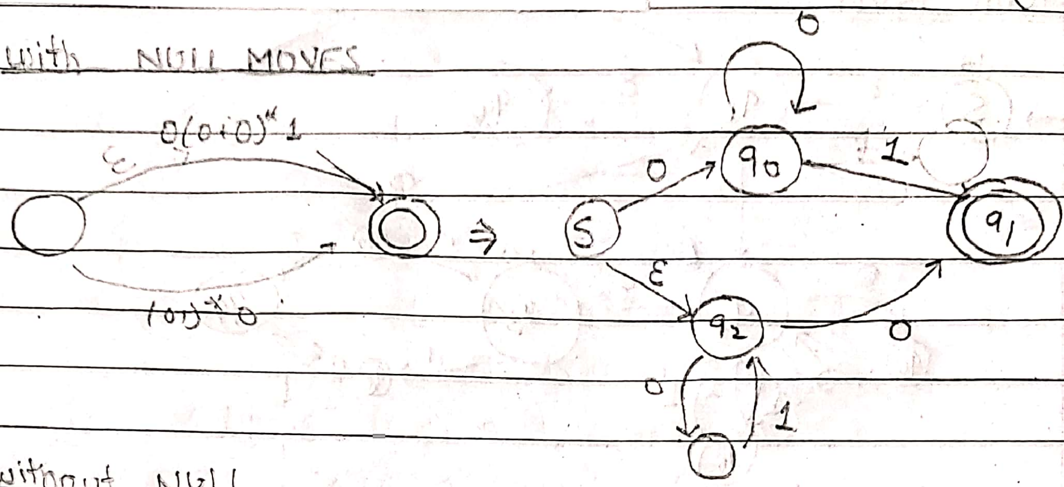
Date : / /
Page No.

B) Given a regular expression $(0(0+0)^*1 + (01)^*0)$. First construct NFA with null moves, then eliminate null moves from the same. 2018 E (old)

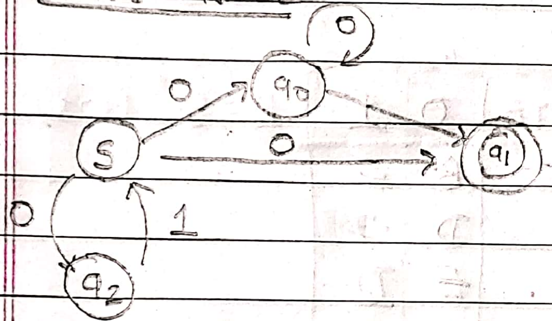
B) Given a regular expression $(0(0+0)^*1 + (01)^*0)$. First construct NFA with null moves, then eliminate null moves from the same. 2018 S

$(0+0) = 0$

with NULL MOVES



without NULL

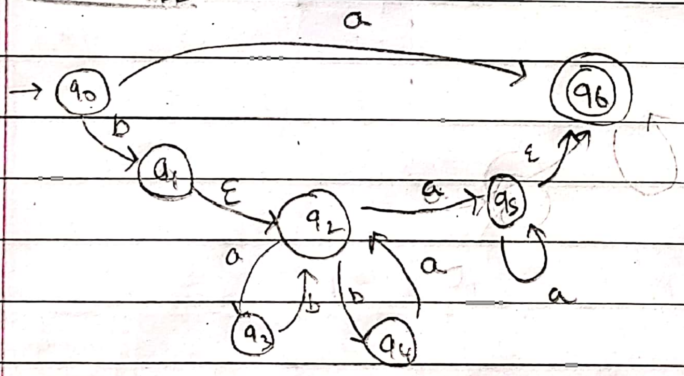


Q.No.3

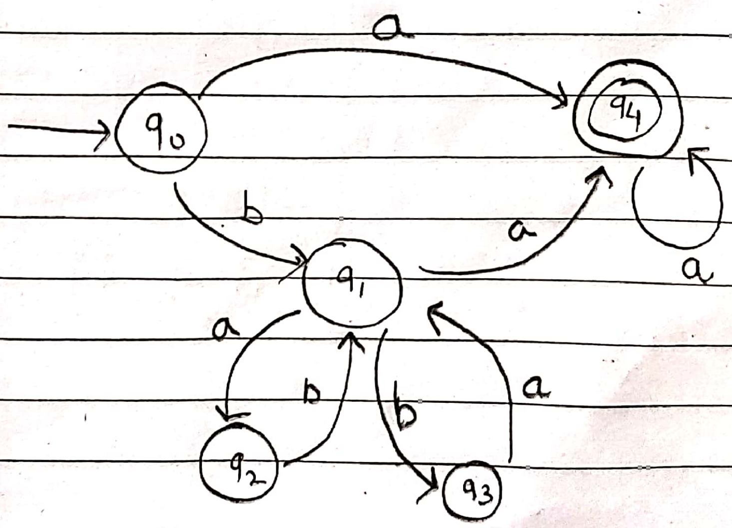
A) Given a regular expression $(a+b(ab+ba)^*aa^*)$, construct NFA with and without null moves. 3+3+2=08

2018M

with NULL



without NULL



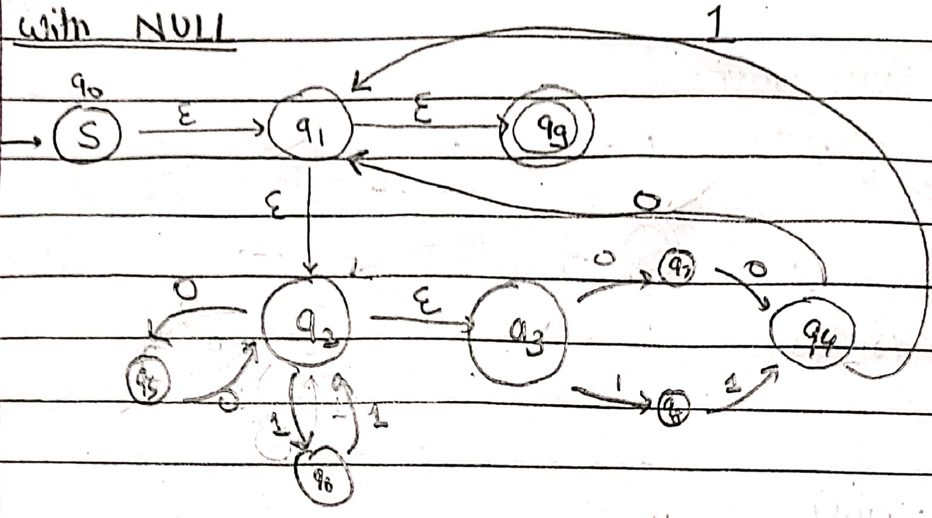
B) Given a regular expression ((00+01) * (11+10) (0+1))*, construct NFA with null moves, then eliminate null moves from the same. 2018E (old)

B) Given a regular expression ((00+11) * (00+11) (0+1))*, construct NFA with null moves, then eliminate null moves from the same. 2019E (old)

Date : / /

Page No.

With NULL 1



without NULL

ε-closure	NFA State	DFA STATE	0	1
{0}	{0,1,9,2,3}	A	B	C
{5,7}	{5,7, }	B	D	F
{6,8}	{6,8 }	C	F	D
{2,4}	{2,3,4}	D	E	F
{5,7,1}	{5,7,1,9,2,3}	E		
{6,8,1}	{6,8,1,2,3,9}	F		

2, 7,

0

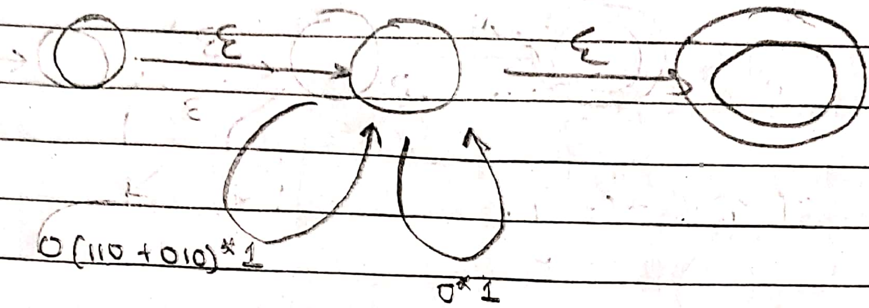
Date : / /

Page No.

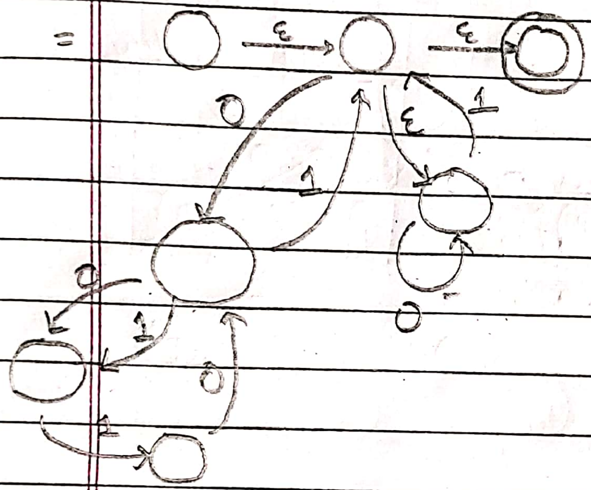
B) Given a regular expression $(0(110 + 010)^*1 + 0^*1)^*$. First constru NFA with null moves, then eliminate null moves from the same.

2018E

without NULL



=



WITHOUT NULL

?

Date : / /
Page No.

c) Let L1 and L2 are regular. Prove L1⊕L2 is also regular. 2018S

PROOF :

If L_1 and L_2 are regular languages then there exists regular expression r_1 and r_2 such that :

$$L_1 = L(r_1)$$
and $$L_2 = L(r_2)$$

as by defination, we know that
$r_1 + r_2$ is regular expression, so correspondingly $L_1 \oplus L_2$ or $L_1 \cup L_2$ is also a regular language.

Hence proved.

b) Given, $min(L) = \{w \in L$ *there is no* $u \in L, v \in \Sigma^+$ *, such that* $w = uv\}$. Show that the family of regular languages is closed under the min operation

We can observe that min(L) can be redefined as:

$min(L) = \{w \mid w$ is in L, but no proper suffix of w is in $L\}$

Eg.) if $L = \{a, ab, aba, ba, bb, baba\}$, then $min(L) = \{a, ba, bb\}$
if $L = \{a^n b \mid n \geq 0\}$, then $min(L) = L$
if $L = \{a^n b^n \mid n \geq 1\}$, then $min(L) = L$
if $L = \{a^n b^n \mid n \geq 0\}$, then $min(L) = \epsilon$

1. by showing DFA can be constructed
Let L be a regular language. So, we have a DFA 'D' for L.
from D, we can make DFA for min(L).
now, from L we want to remove ALL extensions of w i.e. remove all wx which belong to L, where $x \neq \epsilon$.
Now to implement in DFA, we need to,
as soon as we get to final state, say on w, we dont accept any further extensions of w. ~~if we dont~~.
So, from every final state, all transitions lead to a dead state
Since DFA will accept min(L) so it will be closed.

Date : / /
Page No.

(2) Describe the strings which are ineligible for min (L) and exclude them using set difference.

The ineligible strings are $L\Sigma^+$, since $w \in L\Sigma^+$ means that $w = xy$ where $x \in L$. i.e. w has a proper prefix which is in L.

Thus, $min(L) = L - L\Sigma^+$, which is regular since we know that regular languages are closed under set difference

Prove the following statements (any two): (5x2 = 10 marks)
a) If L is a regular Language, prove that the languages $\{uv : u \in L, v \in u^R\}$ is also regular.

2018 ES

Given L is regular and $L' = \{UV, U \in L, V \in U^R\}$
we need to build a DFA can be built for L'.

Now, L is regular and M be the finite automata that recognizes L'
then, DFA for L^R can be generated from M by
→ reversing arcs of M
→ setting initial state in M to final state and connect all final states with ϵ transition to state and set it as initial state.

This would be M^R

→ Now, connect M and M^R with epsilon transition
$U \in M$ and $V \in M^R$

This machine recognizes L'

Since we have built finite automata for L', it is a regular language

Hence Proved.

Date : / /

Page No.

Q.No. 2 [5][CO#1]

Design Moore and Mealy machine for input from $(0+1+2)^*$ print the residue mod 7 of the input string treating it as ternary (base 3, with digits 0, 1,2) number. 2022M

Q.No. 2 2X4.5=09

A) Design Moore machine for input from $(0+1+2)^*$ print the residue mod 7 of the input string treating it as ternary (base 3, with digits 0, 1,2) number. 2019M

B) Design Moore machine for input from $(0+1+2)^*$ print the residue mod 7 of the input string treating it as ternary (base 3, with digits 0, 1,2) number. 2019E (old)

Moore : $M_1 = \{Q, \Sigma, \Delta, \delta, \lambda, q_{in}\}$

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

$\Sigma = \{0, 1, 2\}$

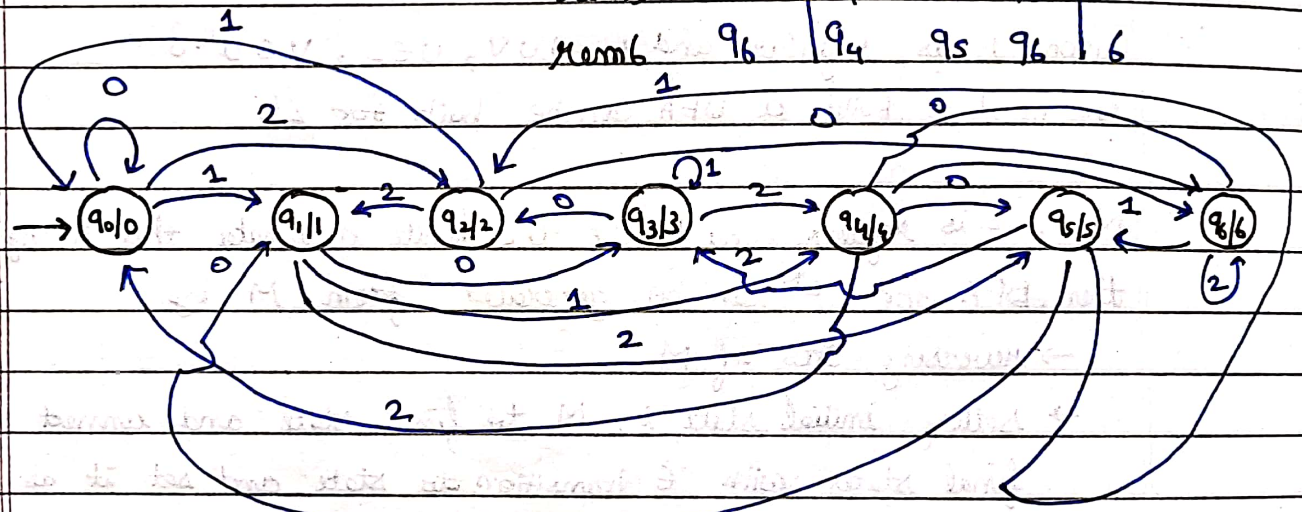
$\Delta = \{0, 1, 2, 3, 4, 5, 6\}$

$\delta = Q \times \Sigma \rightarrow Q$

$\lambda = Q \rightarrow \Delta$

$q_{in} = q_0$

	Q\Σ	0	1	2	Δ
rem0	q_0	q_0	q_1	q_2	0
rem1	q_1	q_3	q_4	q_5	1
rem2	q_2	q_6	q_0	q_1	2
rem3	q_3	q_2	q_3	q_4	3
rem4	q_4	q_5	q_6	q_0	4
rem5	q_5	q_1	q_2	q_3	5
rem6	q_6	q_4	q_5	q_6	6



Mealy Machine : $M_2 = \{Q, \Sigma, \Delta, \delta, \lambda, q_{in}\}$

$Q = \{q_0 - - - q_6\}$

$\Sigma = \{0, 1, 2\}$

$\Delta = \{0, 1, 2 - - 6\}$

$\delta = Q \times \Sigma \rightarrow Q$

$\lambda : Q \times \Sigma \rightarrow \Delta$

$q_{in} = q_{in}$

Q\Σ	0	o/p	1	o/p	2	o/p
q_0	q_0	0	q_1	1	q_2	2
q_1	q_3	3	q_4	4	q_5	5
q_2	q_6	6	q_0	0	q_1	1
q_3	q_2	2	q_3	3	q_4	4
q_4	q_5	5	q_6	6	q_0	0
q_5	q_1	1	q_2	2	q_3	3
q_6	q_4	4	q_5	5	q_6	6

Date : / /
Page No.

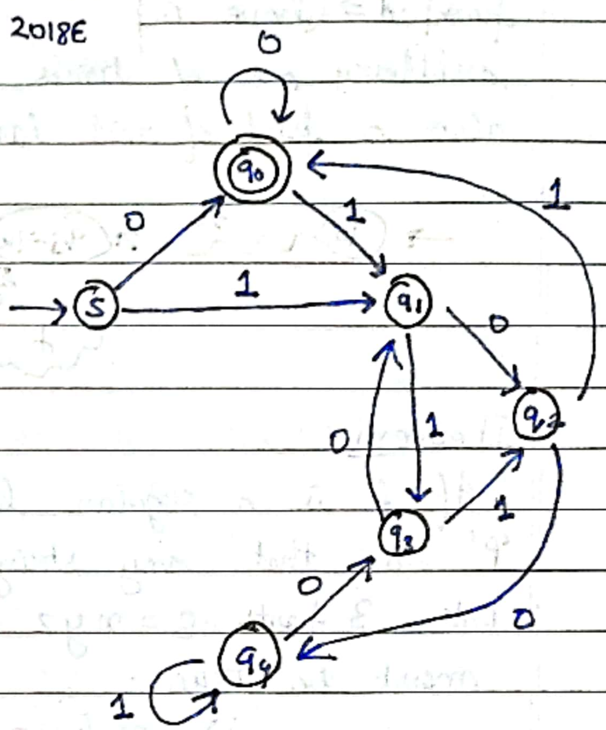
C) Explain Arden's theorem and construct finite automata equivalent to regular expression (RE)=01[((10)*+111)*+0]*1. 2019E

→ Unbalanced Brackets in Question!!!

Q.No. 1 2X5=10

A) Design a finite state machine which can check that the decimal of given binary string is divisible by 5 and also write a regular expression for the language L={$a^n b^m$|(n + m) is even}. 2018E

Q\Σ	0	1
S	q_0	q_1
rem'0' q_0	q_0	q_1
'1' q_1	q_2	q_3
'2' q_2	q_4	q_0
'3' q_3	q_1	q_2
'4' q_4	q_3	q_4



Given,

$$L = \{a^n b^m \mid (n+m) \text{ is even}\}$$

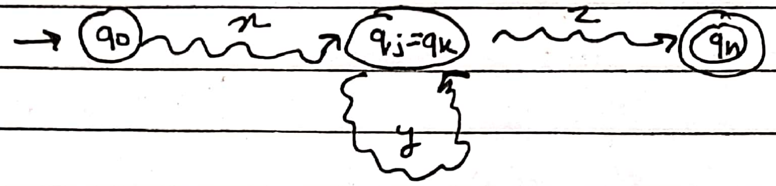
$$= [a(aa)^* \cdot b(bb)^*] + ((aa)^* + (bb)^*)$$

Date : / /
Page No.

Q.No. 4 [5][CO#1]

What is pumping lemma for regular expression? Prove that the language L= {$a^n b^n c^n$ |n ≥1} is not regular. 2023 SM

Pumping lemma for RE is a lemma that describes an essential property of all RE. i.e it says that all sufficiently long strings in a regular language may be pumped ⇒ (have a middle section of the string repeated an arbitrary no. of times to produce a new string that is also a part of that language)



Theorem

If A is a regular language, then A has a pumping length 'p' such that any string 's' where $|S| \geq P$ may be divided into 3 parts $S = xyz$ such that the following conditions must be true :

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

ii.) $|y| > 0$

iii.) $|xy| \leq P$

Q.No. 3

A) What is pumping lemma for regular expression? Show that the language L={$a^n b^n c^n$ | n ≥1 } is not regular. 2X7=14

2018 E (old)

Step 1.) Assume that lang L is regular and its accepted by machine 'M' with 'n' no. of states.

Step 2.) Let us assume that $w = a^n b^n c^n$; $n+n+n = 3n$

$\Rightarrow |w| = 3n > n$ ✓

$w = (xyz)$ so consider the case

$|xy| \leq n$ ✓

$y \neq \epsilon$ ✓

$z \rightarrow$ no restriction

$$\Rightarrow \underbrace{a^{n-1}}_{x} \underbrace{a}_{y} \underbrace{b^n c^n}_{z}$$

C. Explain pumping lemma for regular expression and Prove that the language L= {$a^n b^n c^n$ |n ≥1} is not regular. 2022 E [CO#3]

Date : / /
Page No.

Step3.) $w = xy^i z$
$= a^{n-1} a^i b^n c^n$

for $i=2$

$a^{n-1} a^2 b^n c^n \Rightarrow a^{n+1} b^n c^n \notin L$

This contradicts our assumption,
hence $L = \{a^n b^n c^n \mid n \geq 1\}$ is not regular lang.
Hence proved.

Q.No. 4 [5][CO#1]

What is pumping lemma for regular expression? Prove that the language $L = \{0^n 1^{2n} \mid n \geq 1\}$ is not regular. 2022M

$\boxed{0^{2n} 1^n}$ Step 1:) Assume the lang L is a regular lang and is accepted by a machine M with n states.

Step2:) choose $w = 0^{2p} 1^p$

w can be $x = 0^\alpha$
$y = 0^\beta$
$z = 0^{2p-\alpha-\beta} 1^p$

step3.) Choose an i such that $xy^i z \notin L$
$xy^i z = 0^\alpha 0^{i\beta} 0^{2p-\alpha-\beta} 1^p$
$= 0^{2p+i\beta-\beta} 1^p$

for it to be in L

$\cancel{2p} + i\beta - \beta = \cancel{2p}$
$\Rightarrow i = 1$

but not valid for $i = 2$
↳ Contradiction

∴ not regular

similarly for $0^n 1^{2n}$

Date : / /

Page No.

B) What is pumping lemma for regular expression? Show that the language L= {0^s, where s is a perfect square} is not regular. 2019 E(old)

B) What is pumping lemma for regular expression? Show that the language L= { 0^n | n is a perfect square} is not regular. 2018 M

Step 1:) let L be a regular language and 'n' be a const. let Z be a string from L such that $|Z| \geq n$

L = { ϵ, 0, 0000, 000000000, 0000000000000000, ---}

for i=2 ; Z= 0000 ; $|Z| \geq n \Rightarrow |0000| \geq 2$
$\Rightarrow 4 \geq 2$ ✓

Step 2:)

divide into 3 parts as U V W in such a way that $|UV| \leq n$; $|V| > 0$

for $i \geq 0$ UV^iW is in L

a:) $\underbrace{0}_{U}\ \underbrace{0}_{V}\ \underbrace{0\ 0}_{W}$

i:) $|UV| \leq n \Rightarrow 2 \leq 2$ ✓
ii:) $|V| > 0 \Rightarrow 1 > 0$ ✓

Step 3:) UV^iW

for i=0

Z = 000 ∉ L

This contradicts our assumption

Hence language L is not regular Hence proved.

B) What is pumping lemma for regular language? Show that the language L= { $a^i b^j$ | i>j} is not regular. 2019M

Step 1:) Assume that L is a regular language and n be states

let $\omega = a^{n+1} b^n$; $|\omega| \geq n$ ✓

Step 2:) Split $\omega = xyz$ such that

1.) $y \neq \epsilon$ or $|y| > 0$ ✓
ii.) $|xy| \leq n$ ✓

for n=2 : $\omega = \underbrace{aa}_{x}\underbrace{a}_{y}\underbrace{bb}_{z}$

Step 3) for all $k \geq 0$; $xy^kz \in L$

pick k = 3 | xy^3z = aaabbbb

$\therefore \omega \notin L$ $i=3$ | $j=4$ and $(i<j) \neq (i>j)$

Date : / /
Page No.

Hence, theres a contradiction and language L is not regular. Hence Proved.

Q.No. 3 2X5=10

A) What is pumping lemma for regular expression? Show that the language $L = \{0^n 1^m \mid n \leq m\}$ is not regular. 2018E

Step 1:) Assume L is a regular language with n states.

Step 2:) Let $w = 0^n b^{n+1}$ $|w| \geq n$ ✓

Split $w = xyz$ such that
1) $|y| > 0$ ✓
2) $|xy| \leq n$ ✓

let $n = 2$, $w = 00111$
$x = 0 \mid y = 0 \mid z = 111$

Step 3:) for all $k \geq 0$; $w = xy^k z \in L$ prov

consider $k = 3$ | ∴ $w = 0000111$
here $n = 4 \mid m = 3$
contradiction but $n \leq m$ fails
Thus, its not a regular language. H.P.

2019E

B) Explain Pumping lemma for CFL and prove that $L = \{0^n 1^m \mid \gcd(m,n) > 1\}$ is not regular.

: / /

No.

Let $L = \{a^{n!} : n >= 0\}$. Prove L is not regular using Pumping Lemma theorem.

2018

Suppose L is regular,

$w = a^{m!}$

Now, take $w_0 = a^{m! - k}$

and so,

$m! - k > (m-1)!$

not possible

so,

Contradiction

Not regular.

Date : / /
Page No.

C) Construct a regular grammar accepting L= {w ∈ {0, 1}* |w is a string over {0,1} such that the number of 1's is 3 mod 4}. 2018E (old)

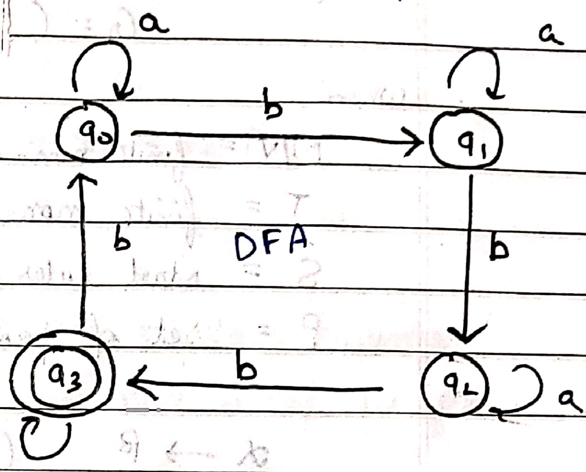
C) Construct a regular grammar accepting L= {w ∈ {0, 1}* |w is a string over {0,1} such that the number of 1's is 3 mod 4}. 2018S

B) Construct a regular grammar accepting L= {w ∈ {0, 1}* |w is a string over {0,1} such that the number of 1's is 3 mod 4. 2019E

C) Construct a regular grammar accepting L= {w ∈ {a, b}* |w is a stri over {a,b} such that the number of b's is 3 mod 4}. 2018E

3 % 4 = 3 places

∴ 4k + 3 places



$G = (V, T, S, P)$

where

$V = \{A_0, A_1, A_2, A_3\}$

$T = \{a, b\}$

S = ~~State Symbol~~ A_0

P: $A_0 \rightarrow aA_0$

$A_0 \rightarrow bA_1$

$A_1 \rightarrow aA_1$

$A_1 \rightarrow bA_2$

$A_2 \rightarrow aA_2$

$A_2 \rightarrow bA_3$

$A_2 \rightarrow b$

$A_3 \rightarrow aA_3$

$A_3 \rightarrow bA_0$

Date : / /

Page No.

UNIT -4 : Context free Language

UNIT - 3 : context-free grammar

C) Explain Chomsky Classification of grammar with suitable examples. 2018M

(a) Explain the Chomsky classification of grammar.

A grammar 'G' can be formally described using 4 tuples,

$$G = (V, T, S, P)$$

Where

N/V = finite non-empty set of variables / non-terminals

T = finite non-empty set of constants / terminals

S = start rules / symbol

P = set of production rules for terminals and non terminals

$\alpha \rightarrow \beta$ (one step prod^n)

$\alpha \rightarrow \alpha_1 \rightarrow \alpha_2 - - - \rightarrow \alpha_n \rightarrow \beta$ or $\alpha \xrightarrow{n} \beta$ (n-step prod^n)

CHOMSKY HEIRARCHY / CLASSIFICATION OF GRAMMAR.

Chomsky classified grammer into four types based on the production rule

→ TYPE 0 / PSG

A grammer is called type 0 or phase structured grammer if all productions of the grammar are of the form :

$\alpha \rightarrow \beta$; where $\alpha, \beta \in \{T \cup N\}^*$

$AaB \rightarrow a\alpha b$

→ TYPE 1 / CSG

A grammer is called type 1 or context sensitive grammer if all productions of the grammar are of the form

$\phi_1 A \phi_2 \rightarrow \phi_1 \alpha \phi_2$; $\phi_1, \phi_2 \in (N \cup T)^*$

$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \alpha \alpha_2$ $\quad A \in N$; $\alpha \neq \epsilon$

Date: / /

→ TYPE 2 / CFG

A grammar is called type 2 or context free grammar if every production of the grammar is of the form

$$A \rightarrow \alpha \quad ; \text{ where } A \in N$$
$$\alpha \in (T \cup N)^*$$

→ TYPE 3 / RE

A grammar is called type 3 or regular grammar if every production of the grammar is of the form :

$$A \rightarrow a \quad \text{or} \quad A \rightarrow aB \quad ; \quad A, B \in N$$
$$a \in T.$$

Q.No. 5 [5][CO#2]

What is chomsky's classification for the grammar? Design a Context Free Grammar (CFG) for the language L= {$a^n b^m$ | n≠m}. 2022M

$G = (V, T, P, S)$

where,

$V = \{X, Y, S\}$

$T = \{0, 1\}$

S = Start symbol

P : $S \rightarrow aSb$ | $X \rightarrow aX$ | $Y \rightarrow a$
$S \rightarrow X$ | $X \rightarrow a$ | $Y \rightarrow aY$
$S \rightarrow Y$

c) What is context free grammar? Write CFG which accepts the language $L = \{0^i 1^j 0^k | j > i + k\}$. 2019 E(odd)

consider $i, k \geq 1$.

a = 0
b = 1

C) What is context free grammar? Write CFG which accepts the language $L = \{a^i b^j a^k | j > i + k\}$. 2019 E

$G = (V, T, S, P)$

where

$V = \{X, Y, Z, S\}$

$T = \{a, b\}$

$S = \{\text{Start symbol}\}$

P: $S \rightarrow XYZ$

$X \rightarrow 0X1 \mid X \rightarrow 01$

$Y \rightarrow 1Y \mid Y \rightarrow 1$

$Z \rightarrow 1Z0 \mid Z \rightarrow 10$

Date : / /
Page No.

C) Write a Context free grammar(CFG) which accepts the language $L = \{0^i 1^j 2^k \mid k \le i \text{ or } k \le j\}$

2018E

let

$L = L_1 \cup L_2$ where, $L_1 = \{0^i 1^j 2^k \mid k \le i\}$

$L_2 = \{0^i 1^j 2^k \mid k \le j\}$

Grammar for $L_1 \rightarrow G_1$

$G_1 = (V_1, T_1, S_1, P_1)$

where

$V_1 = \{S_1, X, Y\}$

$T_1 = \{0, 1\}$

S_1 = Start symbol

P_1: $S_1 \rightarrow S_1 2$

$S_1 \rightarrow 0X2$

$X \rightarrow Y$

$X \rightarrow 0X$

$X \rightarrow \epsilon$

$Y \rightarrow 1Y$

$Y \rightarrow \epsilon$

Grammar for $L_2 \rightarrow G_2$

$G_2 = (V_2, T_2, S_2, P_2)$

where

$V_1 = \{S_2, A, B\}$

$T_2 = \{0, 1\}$

S_2 = Start symbol

P_2: $S_2 \rightarrow AB$

$A \rightarrow 0A$

$A \rightarrow \epsilon$

$B \rightarrow 1B2$

$B \rightarrow 1B$

$B \rightarrow \epsilon$

thus, for Language L

$\therefore$ $G = (V, T, P, S)$

$V = \{S_1, S_2, A, B, X, Y, S\}$

$T = \{0, 1\}$

S = Start symbol

P : $S \rightarrow S_1 / S_2$

$S_1 \rightarrow S_1 2 / 0X2$ | $S_2 \rightarrow AB$

$X \rightarrow Y / 0X / \epsilon$ | $A \rightarrow 0A / \epsilon$

$Y \rightarrow 1Y / \epsilon$ | $B \rightarrow 1B2 / 1B / \epsilon$

Ans.

Date : / /
Page No.

B) Explain Chomsky's Hierarchy of grammars and write a context free grammar (CFG) to generate the language $L = \{0^x 1^y 2^z \mid |x - y| = z, where\ x, y, z > 0\}$. 2018E

C) Write CFG for the language $L = \{a^i b^j c^k \mid i = j + k\}$.

$a^i b^j c^k = a^i b^j c^j c^i$

for $j, k \geq 1$

$G = (V, T, P, S)$

$V = \{S, X\}$
$T = \{a, b, c\}$
S = Start symbol

P:

$S \to aSc \mid S \to aXc$
$X \to bXc \mid X \to bc$

if $j, k \geq 0$

$S \to aSc \mid S \to X$
$X \to bXc \mid X \to \epsilon$

Date : / /
Page No.

Q.No. 5 [5][CO#2]

What is chomsky's classification for the grammar? Design a Context Free Grammar (CFG) for the language which do not contain 3-consecutive 'b's over Σ= {a, b}.

2023 SM

let G = (V, T, S, P)

V = { S,

T = {a, b}

S = Start symbol

P :

S → A | B

A → a | aS

B → b | bb | bA | bbA

Q.No. 5 2X7=14

A) Explain Chomsky's Hierarchy of grammars with suitable examples and consider production rules of following CFG to find an equivalent grammar in Greibach Normal Form(GNF).

S → S+S/S*S
S→b
S→a

2018 E (old)

① S → S+S / S*S

S → b | bS

S → a | aS

② S → SX | SY

S → b | bS

S → a | aS

X → +S

Y → *S

③ S → bX / bSX / aX / aSX

S → bY | bSY | aY | aSY

S → b | bS | a | aS

X → +S

Y → *S

GNF

2X7=14

Date : / /
Page No.

Q.No. 4

A) Construct a reduced grammar equivalent to the grammar
$S \rightarrow aAa$, $A \rightarrow Sb|DaA|bCC$, $C \rightarrow abb|DD$
$E \rightarrow aC$, $D \rightarrow aDA$

2018

Step 1.) $W_1 = \{C\}$ as $C \rightarrow abb$ is the only production with a terminal string on RHS.

$W_2 = \{C\} \cup \{E, A\}$

$\because E \rightarrow aC \rightarrow aabb$

$A \rightarrow bCC \rightarrow babbabb$

$W_{i+1} = W_i \cup \{A \in N \mid$ there exists some production $A \rightarrow \alpha$ with $\alpha \in (T \cup W_2)^* \}$

$W_3 = \{C, E, A\} \cup \{S\}$

as $S \rightarrow aAa$ and

$aAa \in \{T \cup W_2\}^*$

$W_4 = W_3 \cup \phi$

$W_4 = W_3$

$\boxed{N' = W_3 = \{S, A, C, E\}}$

$P' = \{A' \rightarrow \alpha \mid \alpha \in (N' \cup T)^*\}$

$= \boxed{\{S \rightarrow aAa, A \rightarrow Sb \mid bCC, C \rightarrow abb, E \rightarrow aC\}}$

$\boxed{\therefore G_1 = \{N', \{a, b\}, P', S\}}$

Step 2.) $W_1 = \{S\}$

as we have $S \rightarrow aAa$

$S \in W_1$

$W_2 = \{S\} \cup \{A, a\}$

as $A \rightarrow Sb \mid bCC$

$W_3 = \{S, A, a\} \cup \{S, b, C\}$

$= \{S, A, C, a, b\}$

as we have $C \rightarrow abb$

$W_4 = W_3 \cup \{a, b\} = W_3$

hence, $P'' = \{A_1 \rightarrow \alpha \mid A_1 \in W_3\}$

$= \{S \rightarrow aAa, A \rightarrow Sb \mid bCC, C \rightarrow abb\}$

therefore, $G' = (\{S, A, C\}, \{a, b\}, P'', S\}$

is reduced grammar.

C) Explain the ambiguity of context free grammar with suitable example and also discuss steps to remove ambiguity. 2018S

C) Explain the ambiguity of context free grammar with suitable example and also discuss steps to remove ambiguity. 2018E (dd)

A CFG is called ambiguous grammar if it generate same String by using more than one path / derivation tree.

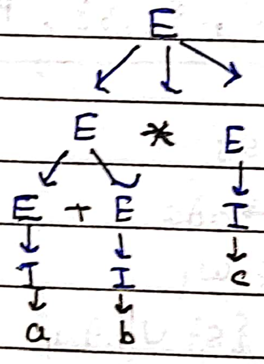
⇒ The grammars which have more than one derivation tree or parse tree are ambiguous grammar. They cant be parsed by any parsers.

Eg.)
P1: $E \to E+E$
P2: $E \to E \times E$
P3: $E \to (E)$
P4: $E \to I$
P5: $I \to a/b/c$

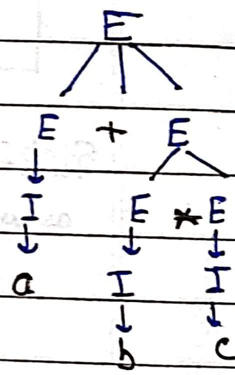
Consider leftmost derivation

①		②	
$E \to E+E$	(P1)	$E \to E \times E$	(P2)
$\to I+E$	(P4)	$\to E+E \times E$	(P1)
$\to a+E$	(P5)	$\to I+E \times E$	(P4)
$\to a+E \times E$	(P2)	$\to a+E \times E$	(P5)
$\to a+I \times E$	(P4)	$\to a+I \times E$	(P4)
$\to a+b \times E$	(P5)	$\to a+b \times E$	(P5)
$\to a+b \times I$	(P4)	$\to a+b \times I$	(P4)
$\to a+b \times c$	(P5)	$\to a+b \times c$	(P5)

TREE



TREE



Removing Ambiguity

OUT OF SYLLABUS

5 [a] Explain parse tree and ambiguity of grammar with example.

When a string is to be generated from the given production rules then it will be very convenient to show the generates pictorially. It will be in form of a tree and its known as parse tree or derivation tree.
The tree will have labels to all the nodes. These could be non-terminal symbols or terminal symbol 'ε', also the root is always labelled start. All the nodes having labels as non-terminal are interior nodes. All the leaf nodes will be terminal symbols.

Date : / /

Page No.

C) Explain Pumping lemma for context free Language and show that following grammar is ambiguous

S → A | B | b , A → aAB | ab , B → abB | ∈ .

2018 E (old)

Q.No. 5 2X7=14

A) Explain Chomsky's Hierarchy of grammars and check whether the following grammar is ambiguous for string "ibtibtaea?

1. S → iCtS / 2. iCtSeS
3. C → b
4. S → a

2019 E (old)

use LMD

① LMD :

S → iCtS (using 1)
S → ibtS (using 3)
S → ibtiCtSeS (using 1)
S → ibtibtSeS (using 3)
S → ibtibtaeS (using 4)
S → ibtibtaea (using 4)

② LMD :

S → iCtSeS (using 2)
S → ibtSeS (using 3)
S → ibtiCtSeS (using 2)
S → ibtibtSeS (using 3)
S → ibtibtaeS (using 4)
S → ibtibtaea (using 4)

Since grammar can derive the string "ibtibtaea" using more than 1 LMD, grammar is ambiguous.

B) State and prove Pumping lemma for Context Free Language (CFL). 2016E

B) What is pumping lemma for CFG? Prove Pumping lemma for Context Free Language (CFL).

"The pumping lemma for context free ~~grammar~~ language states that there are always two short substrings closure to each that can be repeated both same number of times as often as you like."

LEMMA: Let L is a CFL, there exists a constant n depending on L, let $w \in L$. $|w| \geq n$ then we may write $w = uvxyz$, such that,

1. $|vy| \geq 1$
2. $|vxy| \leq n$

for every $i \geq 0$ $\quad uv^ixy^iz \in L$

PROOF:

B) What is pumping lemma for CFG? Prove Pumping lemma for Context Free Language (CFL). 2018S

Date : / /
Page No.

B) Explain the ambiguity of context free grammar with example and show that Language L= { $a^m b^m c^n$ | m≤ n≥ 2m} is not a context free language(CFL). 2018E

$m \le n \le 2m$

Let L be a context free language.
⇒ exists a pumping length p for L

choose $\omega = a\ b\ c$

C) Explain Pumping lemma for context free Language and show that L={ $a^i b^j$ | j= i² } is not context free Language. 2019E

Let L be context free
⇒ exists a pumping length p for L

∴ choose $\omega = a^p b^{p^2}$
decompose into uvxyz acc to rules.

Case 1: v,y lie in a.	Case 2: v,y lie in b	Case 3: lie in both
$\underbrace{aaa}_{v\,x\,y}$ bbbbbbbbb	aa $\underbrace{bbbb}_{vxy}$	$\underbrace{aa}_{v}\underbrace{bbbb}_{y}$
∴ pump up to i=2 aa aaa bbbbbbbbb $(\# a)^2 \neq \# b \Rightarrow 25 \neq 9$	∴ pump to i=2 aa bbbbbb $(\# a)^2 \neq (\# b) \Rightarrow 4 \neq 6$	pump to i=2 aaa bbbbb $3^2 \neq 5$

∴ ω ∉ L ⇒ contradiction

Hence, our assumption that L is a CFL is wrong. ∴ It's not CFL.

Date : / /

Page No.

B) Decide whether the Language $L = \{ a^n b^{2n} a^n \mid n \geq 0 \}$ is context free. Prove that if L is context free language and F is a regular language then L-F is a context free language.

2018E/ 2018E(cont)

Date : / /

Page No.

C) Write CFG for the language $L = \{a^i b^j c^k \mid i = j + k\}$. 2018S

Given $L = \{a^i b^j c^k \mid i = j+k\}$ is language.

let grammar be $G = (N, T, P, S)$

$\therefore N = \{S_0, S_1, S_2\}$

$T = \{a, b, c\}$

S = Start symbol $\{S_0\}$

$P \Rightarrow$

$S_0 \rightarrow a S_1 b S_2$	$S_1 \rightarrow a S_1 b$, $S_1 \rightarrow \epsilon$
$S_0 \rightarrow S_1 b S_2 c$	$S_2 \rightarrow b S_2 c$, $S_1 \rightarrow \epsilon$
$S_0 \rightarrow \epsilon$	

$\therefore$ CFG, $G = \{\{S_0, S_1, S_2\}, \{a, b, c\}, \{S_0 \rightarrow a S_1 b S_2 \mid S_1 b S_2 c \mid \epsilon$,
$S_1 \rightarrow a S_1 b \mid \epsilon$
$S_2 \rightarrow b S_2 c \mid \epsilon\}, \{S_0\}\}$

Q.No. 4 [5x2=10]

A. Explain pumping lemma for context free language(CFL) and Design a Context Free Grammar (CFG) for the language L= { x| x ∈ {a, b}* ,the number of a's in x is a multiple of 3). 2022E [CO#2]

Given language $L = \{x \mid x \in \{a,b\}^*$, # a in x is multiple of 3 $\}$

let grammar be $G = (N, T, P, S)$

$\therefore N = \{S, A, B\}$

$T = \{a, b\}$

$S = \{S\}$

P :

$S \rightarrow bS$	$A \rightarrow aB$
$S \rightarrow b$	$B \rightarrow aS$
$S \rightarrow aA$	
$S \rightarrow a$	

$\therefore$ CFG, $G = \{\{S, A, B\}, \{a, b\}, \{S \rightarrow bS \mid b \mid aA \mid a, A \rightarrow aB, B \rightarrow aS\}, \{S\}\}$

Date : / /

Page No.

8. Explain CNF and GNF forms of context free grammar and
i. Convert the following CFG to CNF
S=>abSb| a|aAb, A=>bS|aAAb [CO#2]

2022E

CHOMSKY NORMAL FORM

In the Chomsky Normal Form (CNF), we have restrictions on the length of R.H.S. and the nature of symbols in the R.H.S. of productions.

∴ A CFG, G is in Chomsky normal form if every production of the grammar is of the form

$A \rightarrow a$ or $A \rightarrow BC$, where $A \in N, B \in N, C \in N, a \in T$

and $S \rightarrow \Lambda$ is in G if $\Lambda \in L(G)$ (S dont appear on RHS).

GRIEBACK NORMAL FORM

A CFG, G is in GNF if every production of grammar is of the form

$A \rightarrow a x$, where $A \in N$, $a \in T$, $x \in N^*$

Q.No. 4 2X5=10
A)
i. Convert the following grammar to CNF
S→abSb|a|aAb , A→bS|aAAb

2018E

Given,

$S \rightarrow abSb$

$S \rightarrow a$

$S \rightarrow aAb$

$A \rightarrow bS$

$A \rightarrow aAAb$

clearly, $S \rightarrow abSb \mid aAb$ is not in CNF,

replace Ab by P, Sb by R,

$\therefore S \rightarrow abR \mid a \mid aP$ | $P \rightarrow Ab$

$A \rightarrow bS \mid aAP$ | $R \rightarrow Sb$

clearly $S \rightarrow abR$, $A \rightarrow aAP$ is not in CNF

replace bR by T, aA with U

$\therefore S \rightarrow aT \mid a \mid aP$ | $P \rightarrow Ab$ | $T \rightarrow bR$

$A \rightarrow bS \mid UP$ | $R \rightarrow Sb$ | $U \rightarrow aA$

replace a by V, b by W

$\therefore S \rightarrow VT \mid a \mid VP$ | $P \rightarrow AW$ | $U \rightarrow VA$

P $A \rightarrow WS \mid UP$ | $R \rightarrow SW$ | $V \rightarrow a$

$T \rightarrow WR$ | $W \rightarrow b$

Date : / /
Page No.

∴ Grammar becomes,

$G = (\{S, A, P, T, R, U, V, W\}, \{a, b\}, P, \{S\})$

Q.No. 4 2X7=14

A) Differentiate between CNF and GNF with suitable example and Convert the following grammar to CNF

S → bA|aB , A → bAA|aS|a , B → aBB|bS|b

2019 E (old)

Given,
S → bA | aB
A → bAA | aS | a
B → aBB | bS | b

Replace a by Q, b by P
S → PA | QB
P → b
Q → a

here, A → PAA, B → QBB are not in CNF
∴ replace AA by R, BB by T
A → PR | R → AA
B → QT | T → BB

∴ P : S → PA | QB
P → b
Q → a
A → PR | QS | a
B → QT | PS | b
R → AA
T → BB

∴ CFG becomes,

$G = (\{S, P, Q, R, T, S, A, B\}, \{a, b\}, P, \{S\})$

3. Convert the following grammar into Chomsky Normal Form (CNF): (10 marks)

$S \to aXbX$
$X \to aY \mid bY \mid \varepsilon$
$Y \to X \mid c$

2018 S

① the variable X is nullable, and so therefore is Y, after ε elimination,
S → aXbX | aXb | abX | ab
X → aY | bY | a | b
Y → X | c

② unit production Y → X (eliminate!)
S → aXbX | abX | aXb | ab
X → aY | bY | a | b
Y → aY | bY | a | b | c

③ break up RHS of S, and replace a by A, b by B, c by C
∴ S → EF | AF | EB | AB
X → AY | BY | a | b
Y → AY | BY | a | b | c
E → AX
F → BX
A → a
B → b
C → c

∴ $G = (\{A, B, X, Y, E, F, S\}, \{a, b, c\}, P, \{S\})$

Q.No. 4

A)

i. Convert the following grammar to CNF
$S \to AbA, \quad A \to Aa, A \to \wedge$ 2018 S

Q.No. 4 2X7=14

A)

i. Convert the following grammar to CNF
$S \to AbA, \quad A \to Aa, A \to \wedge$

(1) ε move elimination $A \to \wedge$

$S \to AbA \mid bA \mid Ab \mid b$

$A \to Aa \mid a$

(2) replace Ab by T

$\therefore S \to TA \mid bA \mid Ab \mid b$

$A \to Aa \mid a$

$T \to Ab$

(4) replace a by C, b by B

$S \to TA \mid BA \mid AB \mid b$

$A \to AC \mid a$

$T \to AB$

$B \to b$

$C \to a$

CNF ✓

Q.No. 4 2X5=10

A)

i. Convert the following grammar to CNF 2019E
$S \to AB|0C2$; $A \to 0A1|01$; $B \to 1B2|12$; $C \to 0C2|0D1|01$;
$D \to 0D1|01$

CNF

$A \to BC \mid a$

$S \to AB \mid 0C2$

$A \to 0A1 \mid 01$

$B \to 1B2 \mid 12$

$C \to 0C2 \mid 0D1 \mid 01$

$D \to 0D1 \mid 01$

introduce

$X \to 0$ | $Z \to 2$

$Y \to 1$

$\therefore S \to AB \mid XCZ$

$A \to XAY \mid XY$

$B \to YBZ \mid YZ$

$C \to XCZ \mid XDY \mid XY$

$D \to XDY \mid XY$

introduce,

$E \to XC$ | $X \to 0$

$F \to XD$ | $Y \to 1$

$G \to XA$ | $Z \to 2$

$H \to YB$

$S \to AB \mid EZ$

$A \to GY \mid XY$

$B \to HZ \mid YZ$

$C \to EZ \mid FY \mid XY$

$D \to FY \mid XY$

CNF

ii. Convert following CFG to GNF
S → ABA, A → aA, A → ∧ , B → bB, B → ∧

ii. Convert following CFG to GNF
S → ABA, A → aA, A → ∧ , B → bB, B → ∧ 2018S

Date : / /
Page No.

Given,
S → ABA
A → aA
A → ∧
B → bB
B → ∧

Remove null productions
i) A → ∧
S → BA | AB | B | ABA
A → a | aA
B → bB
B → ∧

ii) B → ∧
S → A | BA | AB | ∧ | AA | ABA | B
A → a | aA
B → b | bB

∴ S' → S ∪ {∧}
S → A | BA | AB | AA | B | ABA
A → a | aA
B → b | bB

Now convert to GNF : $A \to a\alpha \mid \alpha \in (N^*)$

∴ A → a | aA
B → b | bB

put all values of A or B in corr first Non terminal of S's production

S → a | aA | bA | bBA | aB | aAB | aA | aAA | b | bB | aBA | aABA
A → a | aA
B → b | bB

B) Consider the following grammar and eliminate all the null productions from the grammar without changing the language generated by the grammar
S → ABAC
A → aA | ∈ , B → bB | ∈
C → c . 2019E

Given,
S → ABAC
A → aA | ∧
B → bB | ∧
C → c

① remove A → ∧
S → BAC | ABC | ABAC
A → a | aA
B → bB | ∧
C → c

② remove B → ∧
S → AC | BAC | ABC | AC | ABAC | AAC
A → a | aA
B → b | bB
C → c

removed Null Productions !!

NT → T

NT → T NT₁ NT₂ NT₃

Date: / /

Page No.

11. Convert following CFG to GNF
$S \to AA \mid 0 \quad A \to SS \mid 1$ 2018 E

In GNF, $S \to AA$, $A \to SS$ are not in GNF

∴ for $S \to AA$	for $A \to SS$	Answer
$S \to 1A$	$A \to 0S$	Prodn
$S \to SSA$	$A \to AAS$	$S \to a \mid bA \mid aSA \mid bSS \mid$
$S \to 0SA$	$A \to 1AS$	$A \to b \mid aS \mid bAS \mid aAA$
$S \to ASS$	$A \to SAA$	
$S \to bSS$	$A \to 1AA$	$G = \{\{A, S\}, \{0, 1\}, P, \{S\}\}$

11. Convert following CFG to GNF
$S \to aAS \mid a, \quad A \to SbA \mid SS, \quad A \to ba$ 2018 E

Given,

$S \to aAS / a$

$A \to SbA / SS / ba$

It should be of the form $A \to a\alpha$ $(\alpha \in (NT)^*)$

$S \to aAS / a$

$A \to aASbA / abA / aASS / aS$

⇒ $S \to aAS / a$

$B \to bA$

$A \to aABA \mid aB \mid aASS / aS$

It is in GNF!!!

Date : / /

Page No.

11. Convert the following CFG to GNF
$S \Rightarrow ab, S \Rightarrow aS, S \Rightarrow aaS$ 20226

Given,

$S \to ab$

$S \to aS$ /

$S \to aaS$

$\Rightarrow S \to ab \mid aS \mid aaS$

It should be of the form $A \to a\alpha$; $\alpha \in (N_T^*)$

$\therefore S \to aB \mid aS \mid aAS$

$B \to b$

$A \to a$

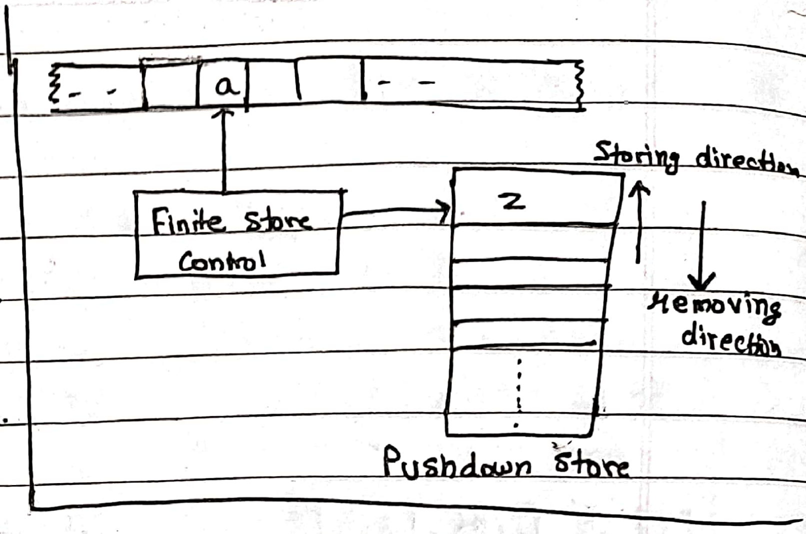
GNF

UNIT 5 - Push Down Automata (PDA)

Date : / /
Page No.

C) What is PDA? Design a Push Down Automata (PDA) which recognize the Language $L = \{0^n 1^n \text{ where } n \geq 0\}$.
2019 E(odd)

A pushdown automata (PDA) is a way to implement a context-free grammar in a machine.



A pushdown automata consists of a collection of 7 parameters-

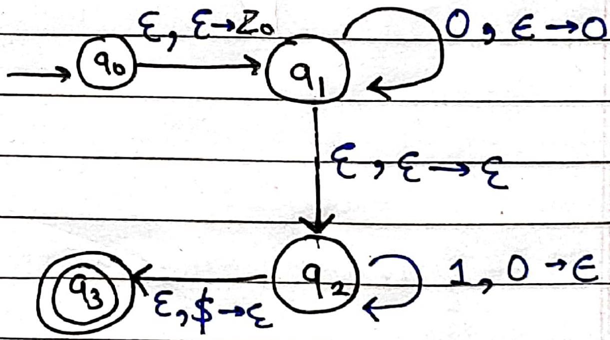
$$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$$

where,

- Q: non empty set of finite states.
- Σ: input alphabets
- Γ: set of stack symbols
- δ: transition function, $Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$
- q_0: initial state of machine
- Z_0: initial symbol of stack
- F: set of finite states.

Given, $L = \{0^n 1^n ; n \geq 0\}$

$\therefore$ PDA, $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$



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

$q_0 = \{q_0\}$

$Z_0 = \{Z_0\}$

$F = \{q_3\}$

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

$\Gamma = \{Z_0, 0, 1\}$

δ :

$\delta(q_0, Z_0, \lambda) = (q_1, Z_0)$

$\delta(q_1, 0, Z_0) = (q_1, 0Z_0)$

$\delta(q_1, 0, 0) = (q_1, 00)$

$\delta(q_1, \varepsilon, Z_0) = (q_1, Z_0)$

$\delta(q_1, \varepsilon, 0) = (q_1, 0)$

$\delta(q_2, \varepsilon, Z_0) = (q_3, Z_0)$

$\delta(q_2, 1, 0) = (q_2, \lambda)$

$0 = a$
$1 = b$

Date : / /
Page No.

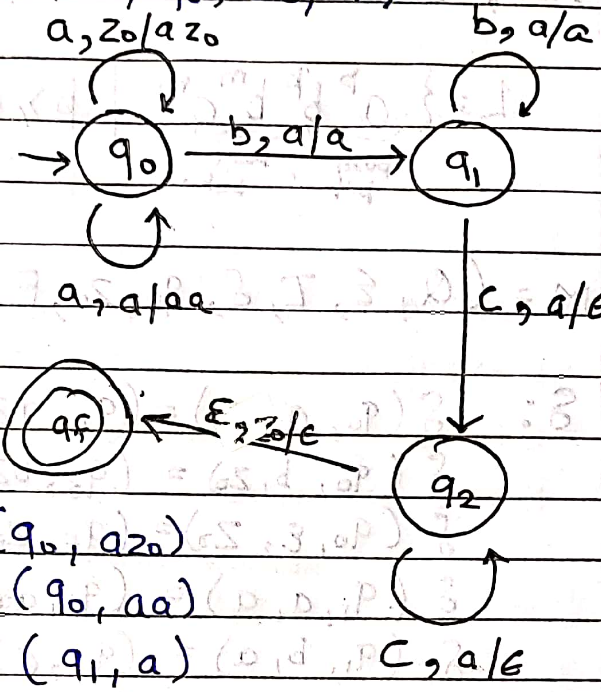
C) Design push down automata (PDA) for the following Language
$L = \{ a^n b^m a^n \mid m, n \geq 1 \}$. 2018 E

Q.No. 5

A) Construct a push down automata for the following Language
$L = \{ 0^n 1^m 0^n \mid$ where n and m > 0$\}$. 2018 E|S

PDA, $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = \{q_0, q_1, q_2, q_f\}$
$\Sigma = \{a, b\}$
$\Gamma = \{Z_0, a, b\}$
$q_0 = \{q_0\}$
$Z_0 = \{Z_0\}$
$F = \{q_f\}$

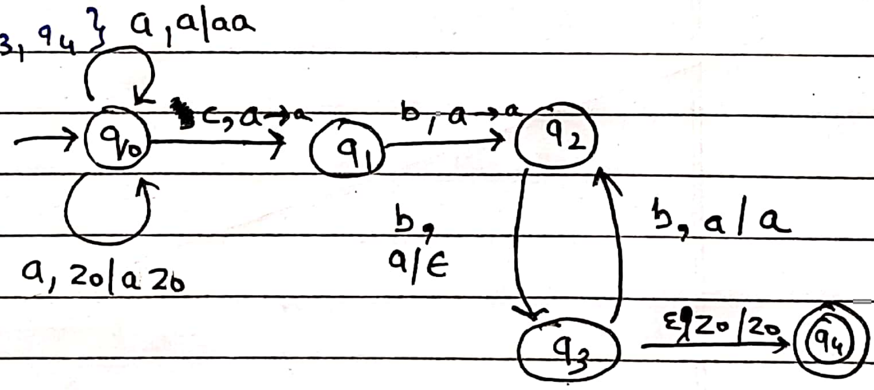


$\delta : \delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, a)$
$\delta(q_1, b, a) = (q_1, a)$
$\delta(q_1, c, a) = (q_2, \epsilon)$
$\delta(q_2, c, a) = (q_2, \epsilon)$
$\delta(q_2, \epsilon, Z_0) = (q_f, \epsilon)$

C) What is PDA? Design a Push Down Automata (PDA) which recognize the Language $L = \{ a^n c b^{2n}$ where $n \geq 1\}$ over $\Sigma = \{a, b, c\}$.
2018 E

PDA, $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$
$\Sigma = (0, 1)$
$\Gamma = \{Z_0, 0\}$
$q_0 = \{q_0\}$
$Z_0 = \{Z_0\}$
$F = \{q_4\}$



$\delta : \delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, c, a) = (q_1, a)$
$\delta(q_1, b, a) = (q_2, a)$
$\delta(q_2, b, a) = (q_3, \epsilon)$
$\delta(q_3, b, a) = (q_2, a)$
$\delta(q_3, \epsilon, Z_0) = (q_4, Z_0)$

Date : / /

Page No.

Q.No. 3 2X5=10

A) Construct a push down automata (PDA) for the following Language $L=\{a^p b^q c^m \mid p+m=q\}$. 2019 E

here, L can be written as,

$$L=\{\underbrace{a^p b^p}_{\substack{push\\pop}} \underbrace{b^m c^m}_{\substack{push\\pop}} \mid p \geq 0, q \geq 0\}$$

$M=(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

$\delta:$ $\delta(q_0, a, z_0) = (q_1, az_0)$

$\delta(q_0, b, z_0) = (q_3, bz_0)$

$\delta(q_0, \epsilon, z_0) = (q_f, z_0)$

$\delta(q_1, a, a) = (q_1, aa)$

$\delta(q_1, b, a) = (q_2, \epsilon)$

$\delta(q_2, b, a) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, z_0) = (q_f, z_0)$

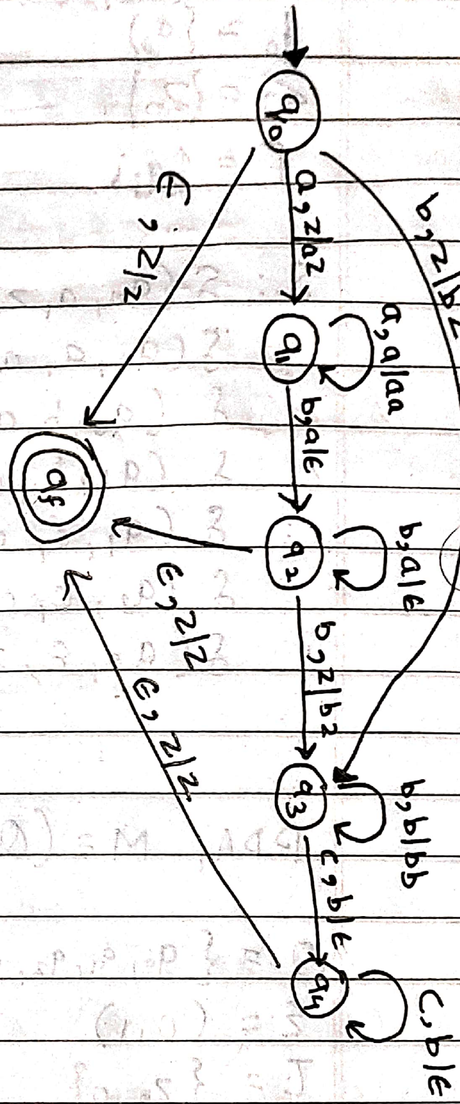
$\delta(q_2, b, z_0) = (q_3, bz_0)$

$\delta(q_3, b, b) = (q_3, bb)$

$\delta(q_3, c, b) = (q_4, \epsilon)$

$\delta(q_4, c, b) = (q_4, \epsilon)$

$\delta(q_4, \epsilon, z_0) = (q_f, z_0)$



Q.No. 5 [5x2=10]

A. Design a Push down automata(PDA) for the following

i. Language $L=\{a^n b^{n+2} \mid n \geq 1\}$ 2022 E [CO#4]

C) Construct a push down automata for the following Language $L=\{0^n 1^{n+2} \mid n=1, 2, 3, \ldots\ldots\}$. 2018 E

PDA, $M=(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

$Q = (q_0, q_1, q_2, q_3, q_f)$

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

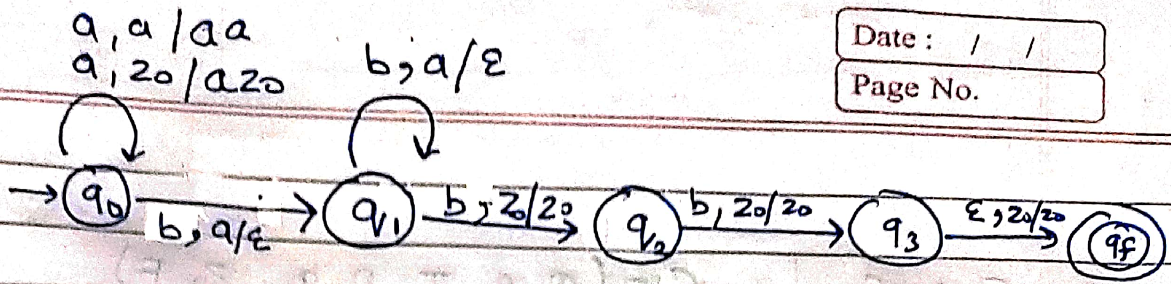
$\Gamma = \{a, b, z_0\}$

$q_0 = q_0$

$z_0 = z_0$

$F = q_f$

Date : / /
Page No.



Transition Diagram for PDA

Q.No. 3 2X7=14

A) Design a PDA that will accept strings over the alphabet {a, b} with equal number of a's and b's and contain at least one pair of a's. 2018S

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

$q_0 = \{q_0\}$

$z_0 = z_0$

$F = q_f$

$Q = \{q_0, q_1, q_f\}$

$\Gamma = \{a, b, z_0\}$

$\delta : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*$

$\delta(q_0, a, z_0) = (q_0, az_0)$	$\delta(q_1, a, z_0) = (q_p, az_0)$
$\delta(q_0, b, z_0) = (q_0, bz_0)$	$\delta(q_1, b, z_0) = (q_1, bz_0)$
$\delta(q_0, a, a) = (q_1, aa)$	$\delta(q_1, a, a) = (q_1, aa)$
$\delta(q_0, b, b) = (q_0, bb)$	$\delta(q_1, b, b) = (q_1, bb)$
$\delta(q_0, a, b) = (q_0, \wedge)$	$\delta(q_1, a, b) = (q_1, \wedge)$
$\delta(q_0, b, a) = (q_0, \wedge)$	$\delta(q_1, b, a) = (q_1, \wedge)$
$\delta(q_0, \wedge, z_0) = (q_0, z_0)$	$\delta(q_1, \wedge, z_0) = (q_f, z_0)$

Date: / /
Page No.

II. Grammar (G): S=>aS|bA, A=>a|aS|bAA, B=>aBB|bS|b, where S is the start symbol of G. 2022E

B) Design a Push down automata for the following grammar: 2018S
S → aS|bA , A → a|aS|bAA , B → aBB|bS|b

prodⁿ rules: $S \to aS$
$S \to bA$
$A \to a$
$A \to aS$
$A \to bAA$
$B \to aBB$
$B \to bS$
$B \to b$

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$\delta:$ ($\delta(q, \Lambda, S) \to (q, aS)$
$\delta(q, \Lambda, S) \to (q, bA)$
$\delta(q, \Lambda, S) \to (q, a)$
$\delta(q, \Lambda, A) \to (q, bAA)$
$\delta(q, \Lambda, B) \to (q, bS)$
$\delta(q, \Lambda, A) \to (q, aS)$
$\delta(q, \Lambda, B) \to (q, aBB)$
$\delta(q, \Lambda, B) \to (q, b)$ } R_1

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

$\delta(q, a, a) \to (q, \Lambda)$, $\delta(q, b, b) \to (q, \Lambda)$) R_2

C) Construct PDA for $L = \{x \in \{a, b, c\}^* \mid |x|_a + |x|_b = |x|_c\}$ ~~by showing~~ all δ-mappings. Also realize the same with the help of example. 2019 E

$PDA = M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$\Sigma = \{a, b, c\}$ | $F = \{q_f\}$
$q_0 = q_0$ | $Q = \{q_0, q_f\}$
$Z_0 = Z_0$ | $\Gamma = \{Z_0, a, b\}$

$Q = \{q_0, q_f\}$

$\delta: (Q \times \Sigma \times \Gamma) \to Q \times \Gamma^*$

$\therefore$ $\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$
$\delta(q_0, c, Z_0) = (q_0, cZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$
$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, c, c) = (q_0, cc)$

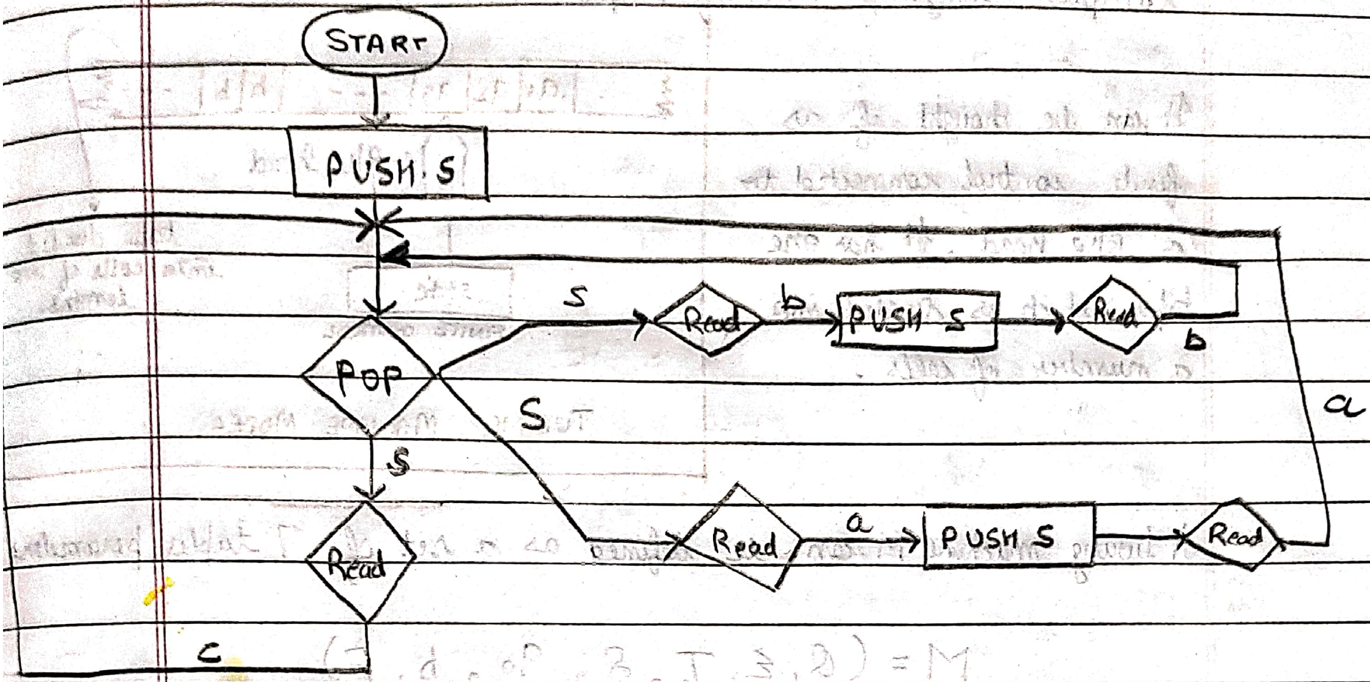
$\delta(q_0, a, c) = (q_0, \Lambda)$
$\delta(q_0, c, a) = (q_0, \Lambda)$
$\delta(q_0, b, c) = (q_0, \Lambda)$
$\delta(q_0, c, b) = (q_0, \Lambda)$

$\delta(q_0, \Lambda, Z_0) = (q_f, Z_0)$

Date : / /
Page No.

B) Explain recursive and recursive enumerable languages and Design a push down automata(PDA) for the following grammar : $S \to aSa|bSb$, $S \to c$. 2018 E

INFORMAL METHOD



FORMAL METHOD

$M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

Convert to normal form,

$C_1 \to a$

$C_2 \to b$

$S \to aSC_1 / bSC_2 / c$

$Q = \{q_0\}$	$z_0 = S$
$\Sigma = \{a, b, c\}$	$F = \{\}$
$\Gamma = \{a, b, c\}$	
$q_0 = q_0$	

δ :

i) for $S \to aSC_1$: $\delta(q_0, \wedge, S) \to (q_0, SC_1)$

ii) for $S \to bSC_2$: $\delta(q_0, \wedge, S) \to (q_0, SC_2)$

iii) for $S \to c$: $\delta(q_0, \wedge, S) \to (q_0, c)$

iv) for $C_1 \to a$: $\delta(q_0, \wedge, C_1) \to (q_0, a)$

v) for $C_2 \to b$: $\delta(q_0, \wedge, C_2) \to (q_0, b)$

for NT, a : $\delta(q_0, a, a) \to (q_0, \wedge)$

for NT, b : $\delta(q_0, b, b) \to (q_0, \wedge)$

for NT, c : $\delta(q_0, c, c) \to (q_0, \wedge)$

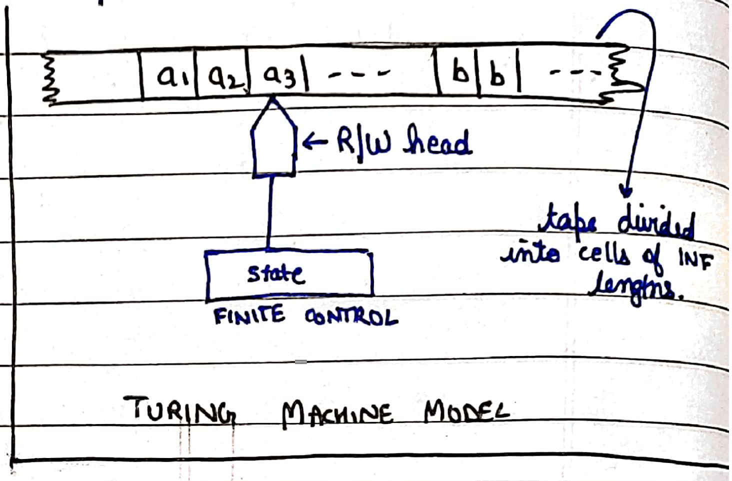
Date : / /
Page No.

UNIT 6 - Turing Machine

Q) What is Turing machine? Design a Turing Machine which accept the string over {a, b} containing even number of a's. 2019 F(old)

A Turing machine is a general purpose computer that can compute anything that is computable.

It can be thought of as finite control connected to a R/W head. It has one tape which is divided into a number of cells.



A turing machine M can be defined as a set of 7 tuples/parameters

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

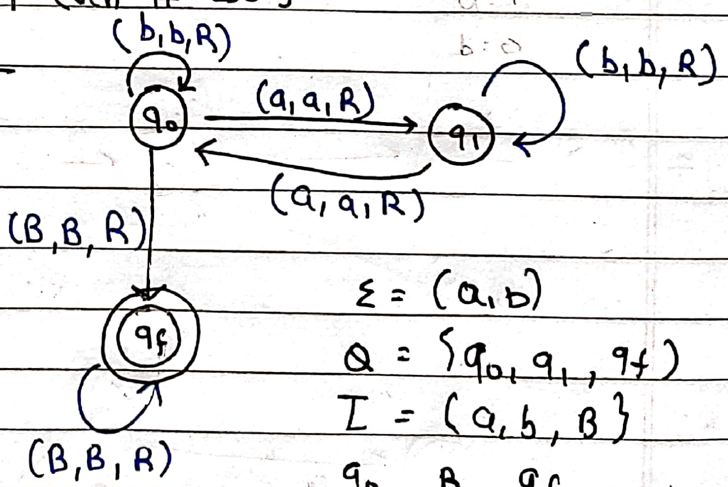
where,

Q = a finite non-empty set of states
Σ = finite non-empty set of input alphabets
Γ = tape symbol
δ = transition function : $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$
q_0 = initial state
b = blank, $b \in \Gamma$, $b \notin \Sigma$
F = final state, $F \subseteq Q$.

$$L = \{ (a,b)^* \mid \text{even } \# \ a\text{s} \}$$

δ:

Q\Σ	a	b	blank (B)
→ q_0	(q_0, b, R)	(q_1, a, R)	(B, B, R)
q_1	(q_1, b, R)	(q_0, a, R)	-
q_f	-	-	(q_f, B, R)



$\Sigma = (a, b)$
$Q = \{q_0, q_1, q_f\}$
$\Gamma = \{a, b, B\}$
q_0, B, q_f

Q.No. 5

A) Design a Turing machine(TM) that will recognize the language $L = \{0^n 1^n | n \geq 0\}$. 2018E

2X5=10

Page No.

B. What is Universal Turing Machine(UTM)? Design a Turing Machine(TM) that will recognize the language $L = \{a^n b^n \mid n \geq 1\}$. 2022E [CO#5]

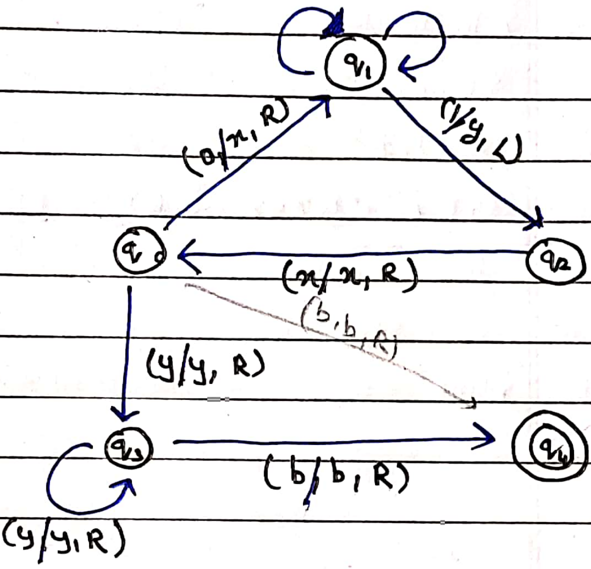
0 = a
1 = b

NOTE :

1. If leftmost symbol is 0, replace by x and move right till leftmost 1. Change it to y and move backwards.
2. Repeat (a) with leftmost 0. If we move back and forth and no 0 or 1 remains, move to final state
3. For strings not of form $0^n 1^n$, resulting state is non final

$\therefore M = (Q, \Sigma, T, \delta, q_0, b, F)$

$Q = \{q_0, q_1, q_2, q_3, q_f\}$
$F = \{q_f\}$
$\Sigma = \{0, 1\}$
$T = \{0, 1, x, y, b\}$



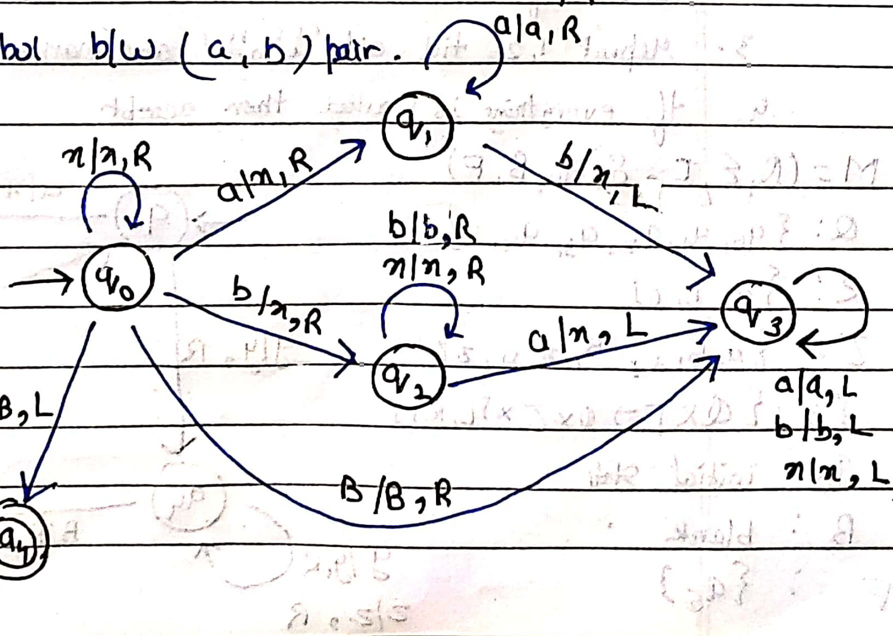
B) Design a Turing Machine(TM) to recognize strings having equal number of a's and b's. 2018 E

NOTE :

we iteratively search and mark pairs of a and b as x in both direction ⇄.

Dont touch symbol b/w (a, b) pair.

$\therefore M = (Q, \Sigma, T, \delta, q_0, B, F)$
$Q : \{q_0, q_1, q_2, q_3, q_4\}$
$\Sigma : \{a, b\}$
$T : \{a, b, B, x\}$
$\delta : Q \times T \rightarrow Q \times T \times \{L, R\}$
q_0 : initial
B : blank
$F : \{q_4\}$



Q.No. 2

2X7=14

A) Construct a Turing machine that will recognize the language $L = \{a^i b^j | i \geq j\}$ over the alphabet {a,b}. 2018 E

Date : / /

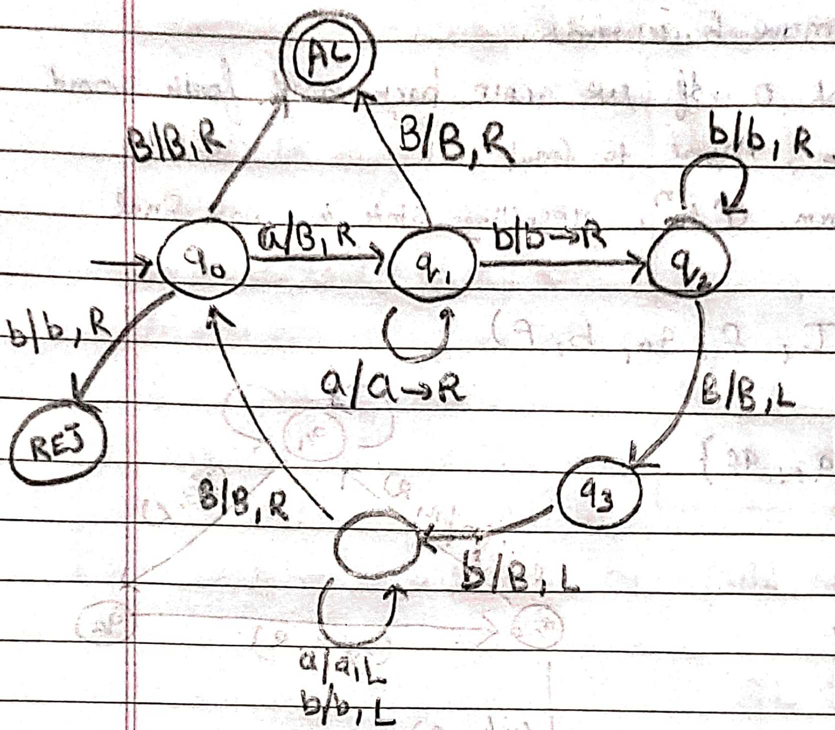
Page No

Q.No. 2

2X7=14

A) Construct a Turing machine that will recognize the language $L = \{a^i b^j | i \geq j\}$ over the alphabet {a,b}. 2018S

B → blank



~~a~~ ~~a~~ ~~a~~ b
B

~~a~~ ~~a~~ ~~b~~ ~~b~~
B

~~a~~ ~~a~~ b
B

Q.No. 2

2X5=10

A) Construct a Turing machine that will recognize the language $L = \{a^n b^n c^n | n \geq 1\}$ 2019E

B. Design a Turing Machine(TM) to recognize the language $L = \{0^n 1^n 2^n | n \geq 1\}$. 2022 E [CO#5]

Note:

1. Mark 'a' as x then right, 'b' as y then right, 'c' as z then left movement.
2. Come to left till x
3. Repeat 1,2 till all 'a','b','c' are marked.
4. If everything is marked then accept

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

$Q: \{q_0, q_1, q_2, q_3, q_4, q_5\}$

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

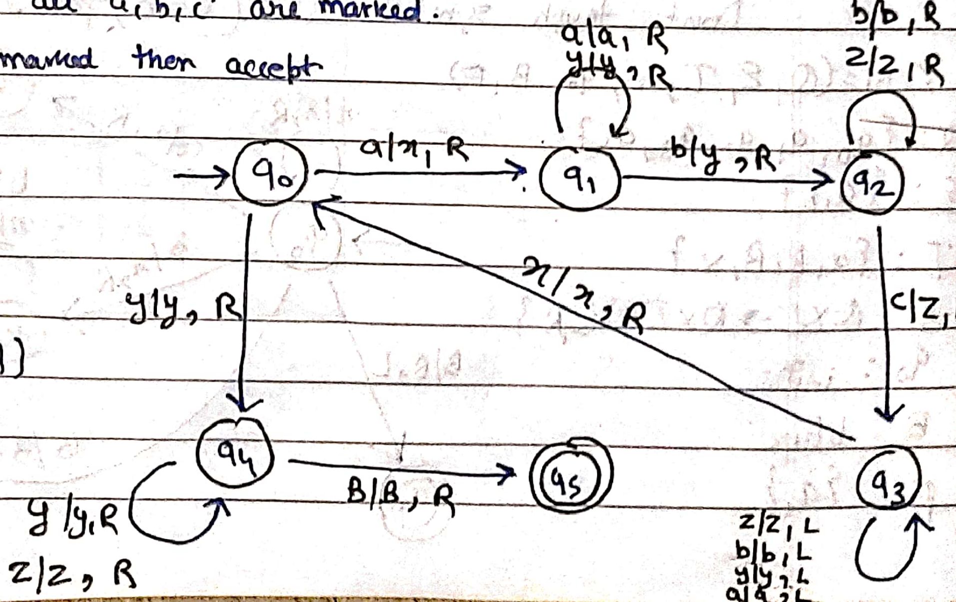
$\Gamma: \{a, b, c, B, x, y, z\}$

$\delta: \{Q \times \Gamma \to Q \times \Gamma \times \{L, R\}\}$

q_0 : initial state

B : blank

F : $\{q_5\}$



Date : / /
Page No.

B) Design a Turing Machine(TM), which can compute 2's complement of a given binary number string. 2019E(old)

NOTE:

1. move to right till blank, then left
2. move till you find 1st 1.
3. after that flip the 0 to 1 and 1 to 0 till blank.

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

Q: { q_0, q_1, q_2, q_f }
Σ : { 0, 1 }
Γ : { 0, 1, b }
q_0 : { q_0 }
F : { q_f }
b : blank

δ:

Q\Σ	0	1	b
q_0	$0Rq_0$	$1Rq_0$	bLq_1
q_1	$0Lq_1$	$1Lq_2$	—
q_2	$1Lq_2$	$0Lq_2$	bRq_f
q_f	—	—	—

B) Construct a Turing machine(TM) that will recognize the language $L = \{a^n b^{2n+1} | n \geq 0\}$. 2018S /2018E

Date : / /
Page No.

Q.No. 2 2X5=1

A) Design a Turing Machine which convert '11' to '10'. 2018E

NOTE:

1. We get first 1 in state q_0 and move right in q_1
2. we get second 1 in State q_1 and move right in q_0
3. otherwise move right in q_0.

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

Q: $\{q_0, q_1, q_f\}$
Σ: $\{0,1\}$
T: $\{0,1,b\}$
δ: $Q \times T \rightarrow Q \times T \times \{L,R\}$
q_0: q_0
q_f: q_f
b : blank state.

δ:

Q \ T	0	1	b
q_0	$q_0,0,R$	$q_1,1,R$	q_f,b,R
q_1	$q_0,0,R$	$q_0,0,R$	q_f,b,R
q_f	—	—	—

B) Design a Turing Machine which convert '111' to '011' in a binary input string. 2018E

If we get '0' at starting there is no effect on TM, aim is to determine consequtive three ones and replace by 011

$\therefore$ If we find '3' 1s, we go back 2 steps in another state and replace first '1' with '0'.

Q \ T	0	1	B
q_0	$q_0,0,R$	$q_1,1,R$	
q_1	$q_0,0,R$	$q_2,1,R$	
q_2	$q_0,0,R$	$q_3,1,L$	A/C
q_3	$q_0,0,R$	$q_4,1,L$	
q_4	—	$q_0,0,R$	

Date : / /
Page No.

Q) What is Universal Turing Machine? Design a Turing Machine that computes a function f (a, b) =a+b, i.e addition of two integers.

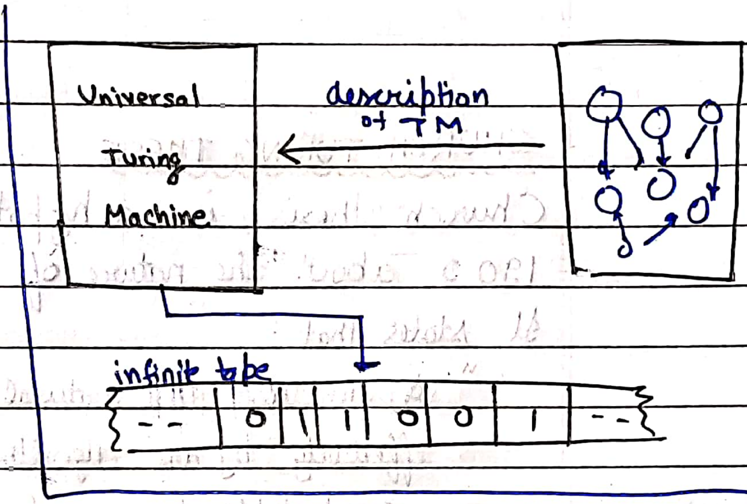
2019 E

A universal turing machine is a theoretical model of a computation proposed by Alon Turing (in 1936) which can simulate the behaviour of any other turing machine.

i.e. It is a turing machine which can compute any algorithmic computation that can be carried out by another turing machine.

It does so by:

1. reading description of another TM from its input tape
2. It creates a new tape to simulate the other machine's behaviour
3. It reads the i/p tape again and uses the simulated machine to compute o/p
4. The o/p is written on o/p tape
5. Stops when done



∴ The Language

$A_{TM} = \{ \langle M, w \rangle \mid M \text{ is a TM and } M \text{ accepts } w \}$ is turing recognizable.

So the UTM will

→ Simulate M

→ behave just like M would (accepts, reject, loop)

∴ UTM is a recognizer but not a decider for A_{TM}

$f(a,b) = a+b$

NOTE:

E.) TM will have $0^m 1 0^n$ on tape

It replaces 1 with 0 after scanning 0s, moves right and replaces rightmost 0 with B when it reaches a blank, halting with m+n 0s on tape.

Date : / /
Page No.

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

$Q : \{q_0, q_1, q_2, q_3\}$
$\Sigma : \{0, 1\}$
$\Gamma : \{0, 1, B\}$
$q_0 : q_0$
$F : \{q_3\}$
B : blank.

δ:

Q \ Σ	0	1	b
q_0	q_0 0 R	q_1 0 R	—
q_1	q_1 0 R	—	q_2, B, L
q_2	q_3 B R	—	—
q_3	—	—	—

6. Explain Church Turing thesis. Draw the Turing machine for copy machine i.e. convert Δab to ΔabΔab Assume tape starts with the blank symbol. 2018S (10 marks)

2018E III. Church's thesis

CHURCH TURING THESIS

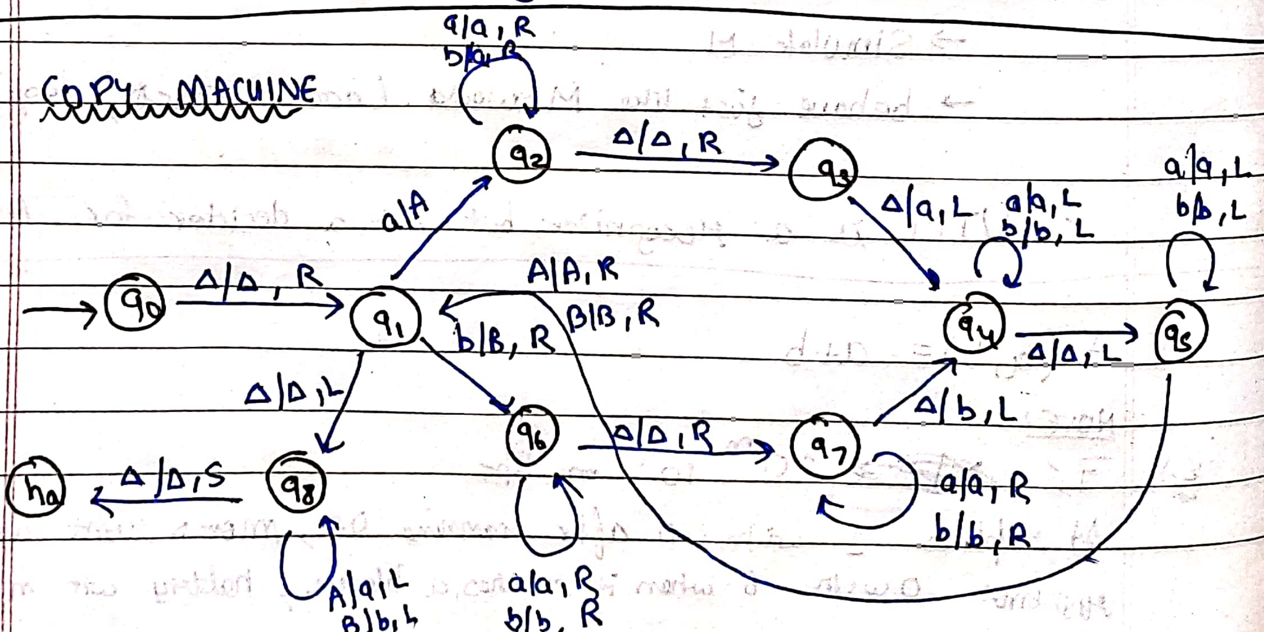
Church's thesis 2022E

Church's Thesis 2019E

Church thesis is a hypothesis given by Alonzo Church in 1930 about the nature of computable functions.
It states that :
" A function on natural numbers can be computed effectively by an algorithm if and only if it is computable by a turing machine "

∴ this cant be proved as such but it means that anything that can be done by current digital computer can also be done by a TM and currently there is no problem which can be solved by a digital computer and can not be solved by a TM.

COPY MACHINE



halt → dont go into INF loop

Date : / /
Page No.

and Halting problem 2018 E

"The halting problem of a TM is that :
" for a given program/algoritm, it is not possible to determine whether the turing machine will ever halt or not "

∴ we cant design a generalized algo which can appropreatly say a program will ever halt or not
⇒ It is an "Undecidable Problem"

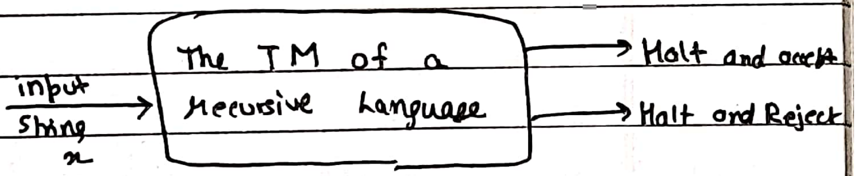
∴ The best solⁿ is to run and check if its halting or not!!

Q) Explain following terms with example
i Q.No. 6 and recursive enumerable languages. 2019E [5x2=10]

A. Explain following with suitable example [CO#6]
I. Recursive and recursively enumerable languages 2022E

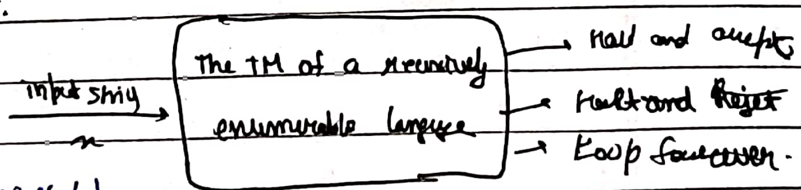
RECURSIVE LANGUAGES
Recursive languages are those whose strings are accepted by a total Turing machine that does either of the following:

→ Halt and accept
→ Halt and reject



A TM of these languages never goes into a loop and always halt in every case.
Hence, they are also called Turing Decidable Languages.

RECURSIVELY ENUMERABLE LANGUAGES
Recursively enumerable languages are those whose string are accepted by TM.
These strings may do :
→ Halt and accept
→ Halt and reject
→ loop forever (if not a part of L)



They are also called Turing Recognizable Languages

Date : / /
Page No.

iii. Post correspondence problem (PCP). 2018B

iii. Post correspondence problem (PCP) and Post Correspondence Problem(PCP) 2019E

ii. Post correspondence Problem(PCP) and 2022B

2018C ii. Post correspondence problem(PCP)

Post correspondence problem (PCP) is an undecidable decision problem that was introduced by Emil Post in 1946.
It is similar ~~to~~ than the halting problem and is used in proofs of undecidability.

It can be stated as follows :-
Given two sequences of 'n' strings on some alphabet 'Σ' say

$A = w_1, w_2, --- w_n$ and

$B = v_1, v_2, --- v_n$

we say that there exists a PC- solution for pair (A, B) if there is a non-empty sequence of integers i, j --- k such that

$w_i w_j --- w_k = v_i v_j --- v_k$

The post correspondance problem is to derive an algorithm that tells us, for any (A, B), whether or not there exists a PC- solution.

Total No. of Pages 1
V-SEMESTER
MID TERM EXAMINATION

Roll No…………
B.Tech.(CO)
Sept/Oct- 2023

CO-303 THEORY OF COMPUTATION

Time: 1:30 Hours

Max. Marks: 25

Note: Attempt **all** questions. Assume suitable missing data, if any

Q.No. 1 *[5] CO1*

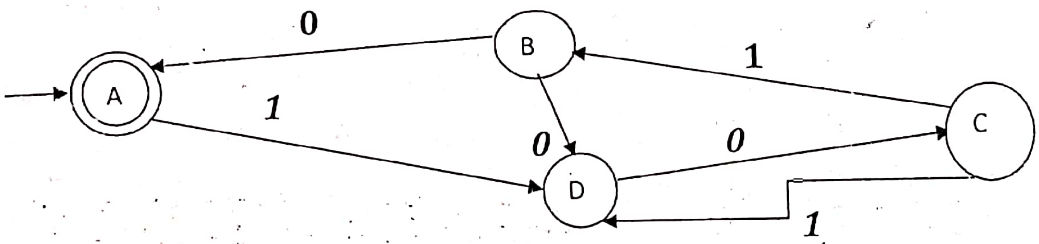
How DFA is different from NDFA? Explain with example and Design a DFA for strings of 0's and 1's having 101 or 110 as a substring.

Q.No. 2 *[5] CO1*

Design mealy machine for input from $(0+1+2)^*$ print the residue mod 5 of the input string treating it as ternary (base 3, with digits 0, 1,2) number.

Q.No. 3 *[5] CO2*

What is Arden's theorem? Find a regular expression (RE) corresponding to the following FA using Arden's theorem.



Q.No. 4 *[5] CO2*

Explain equivalence of two finite Automata and Construct a DFA to recognize the language $(a+b)^*(ab+bba)$

Q.No. 5 *[5] CO3*

Explain pumping lemma for regular expression and Prove that the language **L= $\{ 0^n 1^m 0^{m+n} \mid m \geq 1 \text{ and } n \geq 1 \}$** is not regular.

2K21/CO/262

Total No. of Pages 2
V-SEMESTER
END TERM EXAMINATION

Roll No...........
B.Tech.(CO)
Nov/Dec- 2023

CO-303 THEORY OF COMPUTATION

Time: 3:00 Hours Max: Marks: 50

Note: Attempt any five questions. Assume suitable missing data, if any

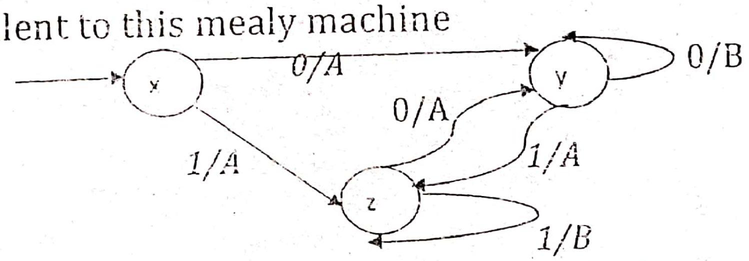
Q.No. 1 [5X2=10] CO1

A. How DFA is different from NDFA? Explain with example and Design a Finite automata (FA) for strings of 0's and 1's ending with 00(consecutive two zero's).

B. Prove that the regular languages are closed under both closure and Complementation.

Q.No. 2 [5X2=10] CO3

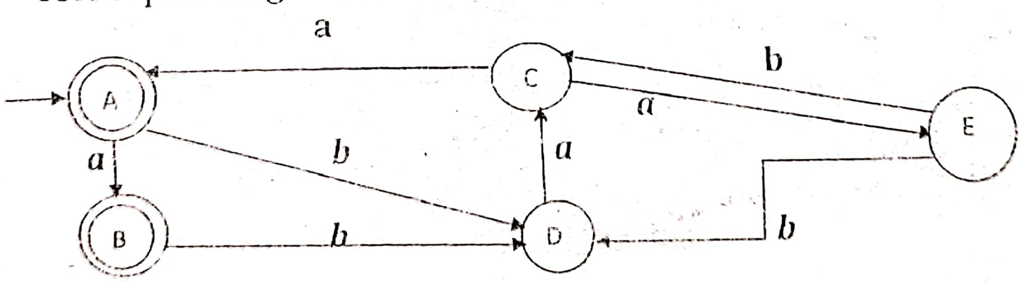
A. Consider following mealy machine, construct a Moore machine equivalent to this mealy machine



B. Construct a DFA accepting strings over {0,1}, whose every block of 4 consecutive symbol, contain at least 3 0's.

Q.No. 3 [5X2=10]

A. What is Arden's theorem? Find a regular expression (RE) corresponding to the following FA using Arden's theorem. CO1



B. What is the need of normal form? Explain CNF and GNF with suitable examples. CO2

Q.No. 4

A. *Design Moore and Mealy machine for binary input sequence if it ends in 110, output is 'X' if it ends in '101' output is 'Y' ,otherwise 'Z'.* *[5] CO3*

B. What is chomsky's classification for the grammar? Design a Context Free Grammar (CFG) for the language which do not contain 3-consecutive 'b's over Σ={a,b}. *[5] CO2*

Q.No. 5

A. Explain pumping lemma for context free language (CFL) and Prove that the language L= $\{0^i 1^j \mid j= i^2\}$ is not CFL. *[5] CO2*

B. Construct a Push down automata (PDA) for the accepting language L = $\{a^x b^y c^z \mid x+z=y\}$ *[5] CO4*

[5X2=10] CO5

Q.No. 6

A. What is universal turing machine (UTM)? Design a Turing Machine(TM) to recognize the language L = $\{a^n b^n \mid n \geq 1\}$.

B. Design Turing Machine(TM) which convert '111' to '011'

Q.No. 7

A. Construct the CFG generating the language accepted by the following PDA: M=({ q_0,q_1}, {0,1},{Z_0,X},δ,q_0, Z_0,ϕ) where δ is given below: *[5] CO4*

$$\delta(q_0, 1, Z_0) = \{(q_0, XZ_0)\}$$
$$\delta(q_0, 1, X) = \{(q_0, XX)\}$$
$$\delta(q_0, 0, X) = \{(q_1, X)\}$$
$$\delta(q_0, €, Z_0) = \{(q_0, €)\}$$
$$\delta(q_1, 1, X) = \{(q_1, €)\}$$
$$\delta(q_1, 0, Z_0) = \{(q_0, Z_0)\}$$

B. Explain following with suitable example *[5] CO6*

i. Recursive and recursively enumerable languages

ii. Post correspondence Problem(PCP) and Church's thesis