Fifth Semester B.E. Degree Examination, CBCS - June / July 2018

Automata Theory & Compatibility

Time: 3 hrs. Max. Marks: 8

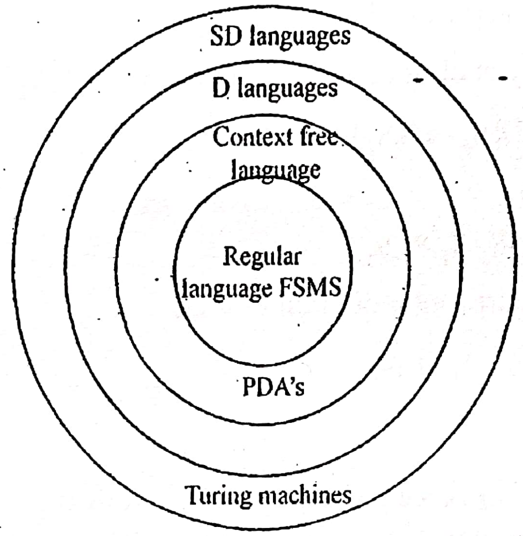
Note : *Answer any FIVE full questions, selecting ONE full question from each module.*

Module - 1

1. a. **With a neat diagram, explain a hierarchy of language classes in automata theor**

(04 Mark

Ans.



Grammar	Language	Automaton
Type - 0	Recursively enumerable	Turing machine
Type - 7	Context sensitive	Linear bounded Non - deterministic Turing machine
Type - 2	Context free	Non determinstic pushdown automata
Type - 3	Regular	finite state automaton

b. **Define deterministic FSM. Draw a DFSM to accept decimal strings which are divisible by 3.** **(06 Marks)**

Ans. Step 1 :- d = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} K = 3

Step 2 :- After dividing by 3, possible reminder are 0, 1, 2

Step 3 :- Compute transition

$$\delta(q_i, a) = q_j \quad \text{where } j = (r * i + d) \bmod K$$

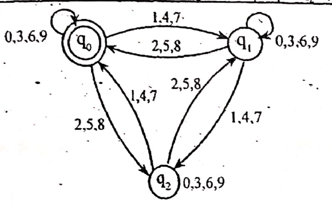
wih r = 10 and K = 3

{0, 3, 6, 9} leaves 0 as reminder)

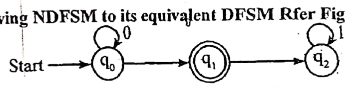
{1, 4, 7} leaves 1 as reminder)

{2, 5, 8} leaves 2 as reminder)

i = 0	0	(10 * 0 + 0) Mod 3 = 0	$\delta(q_0, 0) = q_0$	$\delta(q_0, \{0, 3, 6, 9\}) = q_0$
	1	(10 * 0 + 1) Mod 3 = 1	$\delta(q_0, 1) = q_1$	$\delta(q_0, \{1, 4, 7\}) = q_1$
	2	(10 * 0 + 2) Mod 3 = 2	$\delta(q_0, 2) = q_2$	$\delta(q_0, \{2, 5, 8\}) = q_2$
i = 1	0	(10 * 1 + 0) Mod 3 = 1	$\delta(q_1, 0) = q_1$	$\delta(q_1, \{0, 3, 6, 9\}) = q_1$
	1	(10 * 1 + 1) Mod 3 = 2	$\delta(q_1, 1) = q_2$	$\delta(q_1, \{1, 4, 7\}) = q_2$
	2	(10 * 1 + 2) Mod 3 = 0	$\delta(q_1, 2) = q_0$	$\delta(q_1, \{2, 5, 8\}) = q_0$
i = 2	0	(10 * 2 + 0) Mod 3 = 2	$\delta(q_2, 0) = q_2$	$\delta(q_2, \{0, 3, 6, 9\}) = q_2$
	1	(10 * 2 + 1) Mod 3 = 1	$\delta(q_2, 1) = q_0$	$\delta(q_2, \{1, 4, 7\}) = q_0$
	2	(10 * 2 + 2) Mod 3 = 0	$\delta(q_2, 2) = q_1$	$\delta(q_2, \{2, 5, 8\}) = q_1$



c. **Convert the following NDFSM to its equivalent DFSM Rfer Fig 1.c.**



Also write transition table for DFSM **(06 Marks)**

Ans. **Step 1 :-** Identify start state $Q_D = \{q_0\}$

Step 2 :- Identify alphabet $\Sigma = \{0, 1\}$

Step 3 :- Transitions

Input symbol = 0

$\delta_D = (\{q_0\}, 0) = \delta_N(\{q_0\}, 0) = \{q_0, q_1\}$

For state $\{q_0, q_1\}$

$\delta_D = (\{q_0, q_1\}, 0) = \delta_N(\{q_0, q_1\}, 0)$

$= \delta_N\{q_0, 0\} \cup \delta_N(q_1, 0)$

$= \{q_0, q_1\} \cup \{q_2\} = \{q_0, q_1, q_2\}$

Input symbol = 1

$\delta_D = (\{q_0\}, 1) = \delta_N(\{q_0\}, 1) = \{q_1\}$

$\delta_D = (\{q_0, q_1\}, 1) = \delta_N(\{q_0, q_1\}, 1)$

$= \delta_N\{q_0, 1) \cup \delta_N(q_1, 1\}$

$= \{q_1\} \cup \{q_2\} = \{q_1, q_2\}$

For state $\{q_1\}$

Input Symbol = 0

$\delta_D = (\{q_1\}, 0) = \delta_N(\{q_1\}, 0)$

Input Symbol = 1

$\delta_D = (\{q_1\}, 1) = \delta_N(\{q_1\}, 1)$

For state $\{q_0, q_1, q_2\}$

Input Symbol = 0

$\delta_D = (\{q_0, q_1, q_2\}, 0) = \delta_N(\{q_0, q_1, q_2\}, 0)$

$= \delta_N\{q_0, 0\} \cup \delta_N(q_1, 0) \cup \delta_N(q_2, 0)$

$= \{q_0, q_1\} \cup \{q_2\} \{\phi\} = \{q_0, q_1, q_2\}$

Input Symbol = 1

$\delta_D = (\{q_0, q_1, q_2\}, 1) = \delta_N(\{q_0, q_1, q_2\}, 1)$

$= \delta_N\{q_0, 1) \cup \delta_N(q_1, 1\} \cup \delta_N(q_2, 1\}$

$= \{q_1\} \cup \{q_2\} \cup \{q_2\} = \{q_1, q_2\}$

For state $\{q_1, q_2\}$

$\delta_D = (\{q_1, q_2\}, 0) = \delta_N(\{q_1, q_2\}, 0)$

$= \delta_N\{q_1, 0\} \cup \delta_N(q_2, 0)$

$= \{q_2\} \cup \phi = \{q_2\}$

$\delta_D = (\{q_0, q_1\}, 1) = \delta_N(\{q_0, q_1\}, 1)$

$= \delta_N\{q_1, 1) \cup \delta_N(q_2, 1\}$

$= \{q_2\} \cup \{q_2\} = \{q_2\}$

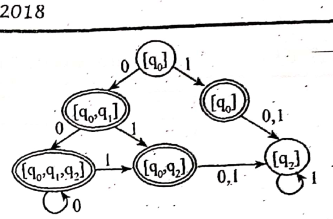
For state $\{q_2\}$

Input Symbol = 0

$\delta_D = (\{q_2\}, 0) = \delta_N(\{q_2\}, 0) = \{\phi\}$

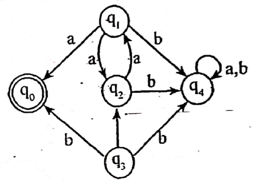
Input Symbol = 1

$\delta_D = (\{q_2\}, 1) = \delta_N(\{q_2\}, 1) = \{q_2\}$



OR

2. a. **Minimize the following finite automata (Refer Fig 2a.)** **(06 Marks)**



Ans. **Step 1 :-**

q_1				
q_2				
q_3				
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

Step 2 :-

δ	a	b
(p, q)	(r, s)	(r, s)
(q_0, q_1)	(q_1, q_2)	(q_3, q_4)
(q_0, q_2)	(q_1, q_1)	(q_3, q_4)
(q_0, q_3)	(q_1, q_2)	(q_3, q_4)
(q_1, q_2)	(q_2, q_1)	(q_4, q_4)
(q_1, q_3)	(q_2, q_2)	(q_4, q_4)
(q_2, q_3)	(q_1, q_2)	(q_4, q_4)

q_1	X			
q_2	X			
q_3	X			
q_4	X	X	X	X
	q_0	q_1	q_2	q_3

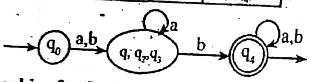
Step 3 :-

δ	a	b
(p, q)	(r, s)	(r, s)
(q_1, q_2)	(q_2, q_1)	(q_4, q_4)
(q_1, q_3)	(q_2, q_2)	(q_4, q_4)
(q_2, q_3)	(q_1, q_3)	(q_4, q_4)

None of the unmarked pairs (r, s) are marked in table.

Step 4 :- (q_0) (q_1 q_2 q_3) (q_4)

State	a	b
q_0	(q_1 q_2 q_3)	(q_1 q_2 q_3)
(q_1 q_2 q_3)	(q_1 q_2 q_3)	(q_4)
(q_4)	(q_4)	(q_4)



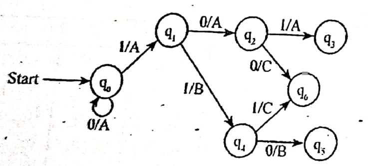
b. Construct a mealy machine for the following

i) Design a mealy machine for a binary input sequence. Such that if it has substring 101, the machine outputs A. if input has substring 110, the machine outputs. B otherwise it outputs C.

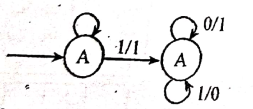
ii) Design a mealy machine that takes binary number as input and produces 2's complement of that number as input.

Assume the string is read from LSB to MSB and end carry is discarded. (06 Marks)

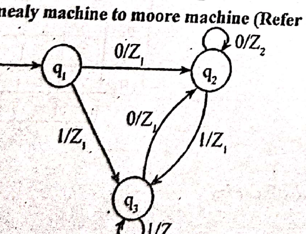
Ans. i)



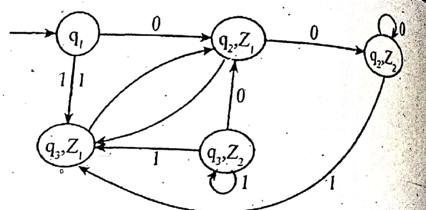
ii)



c. Convert the following mealy machine to moore machine (Refer fig 2.c.) (04 Marks)



Ans.



Module - 2

3. a. Define regular expression. Obtain a regular expression for the following langu

i) L = {a^n b^m | m + n is even}

ii) L = {a^n b^m | m ≥ 1 n ≥ 1, nm ≥ 3}

iii) L = { W : |W| mod 3 = 0 where W ∈ {a, b}*} **(08 Ma**

Ans. Definition :- Refer Q.No. 3.a. of MQP - I

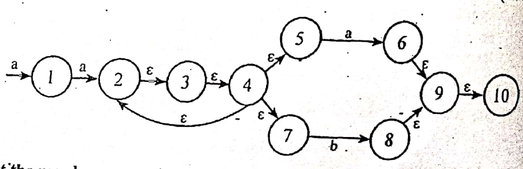
i) R∈ = ((a+b) (a + b) * or R∈ =(a a) * (b b) * + a(a a) * b(b b)*

ii) R∈ = aaaa*b + abbbb* + aaa* bbb*

iii) R∈ = ((a + b) (a + b) (a + b))*

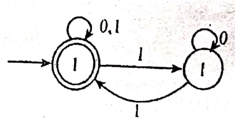
b. Design an NDFSM that accept the language L(aa*(a + b)) **(04 Ma**

Ans.



c. Convert the regular expression (0 + 1) * 1(0 + 1) to NDFSM **(04 Mar**

Ans.



OR

4. a. If the regular grammars define exactly the regular language, then prove that th class of languages that can be defined with regular grammars is exactly the regula languages. **(04 Marks**

Ans. We first show that any languages that can be defined with a regular grammar can b accepted by some FSM and so is regular. Then we must show that every regular languag can be defined with a regular grammar. Both proofs are by construction.

Regular grammar → FSM : The following algorithm construct an FSM M from a regula grammar G = (V, Σ, R,, S) and assures that LCM) = LCG) :

grammar to FSM (G : regular grammar) =
1. Create in M a separate state for each non terminal in V.
2. Make the state corresponding to S the start state.
3. If there are any rules in R of the form $x \rightarrow w$, for some $w \in \Sigma$, then create an additional state labeled #.
4. For each rule of the form $x \rightarrow wy$, add a transition from z to y labeled w.
5. For each rule of the form $x \rightarrow w$, add a transition from x to # labeled w.
6. For each rule of the form $x \rightarrow \varepsilon$, add a transition from x as accepting.
7. Mark state # as accepting.
8. If M is in complete, M requires a dead state. Add a new state D. For every (q, i) pair for which no transition has already been defined, create a transition from q to D labeled i. for every i in Σ, create a transition from D to D labeled i.

b. Prove that the regular language are closed under complement, intersection, difference, reverse and letter substitution. **(08 Marks)**

Ans. i) Under complement :- Let $M1 = (Q, \Sigma, \delta, q_0, F)$ be a DFA which accepts the language 1. Since the language is accepted by a DFA, the language is regular. Noe let us define the machine $M_2 = (Q, \Sigma, \delta, q_0, Q - F)$ which accepts I. Note that there in no difference between M_1 and M_2 except the final states.

The non - final states of M_1 = are th final state of M_2 and final stat of M_1 are the non final states of M_2 so the language which is rejected by M_1 is accepted by M_2 and vice versa. Thus we have a machine M_2 which accepts all those strings denoted by I that are rejected by machine M_1. So regular language is closed under complement.

ii) Intersection :-

$M_1 = (Q, \Sigma, \delta_1, q_1, F_1)$ which accepts L_1

$M_2 = (Q, \Sigma, \delta_2, q_2, F_2)$ which accepts L_2

$Q = Q_1 \times Q_2$

$q = (q_1, q_2)$ where q_1 and q_2 are the start states of machine M_1 and M_2 respectively.

$\hat{\delta}_1(q_1, w) \in F_1$ and $\hat{\delta}_2(q_2, w) \in F_2$.

i.e.., if and only if $w \in L_1 \cap L_2$. So the regular language is closed under intersection.

iii) Difference

$M_1 = (Q, \Sigma, \delta_1, q_1, F_1) \rightarrow L_1$

$M_2 = (Q, \Sigma, \delta_2, q_2, F_2) \rightarrow L_2$

$\hat{\delta}_1(q_1 q_2), w)$ is in F

$(\hat{\delta}_1(q_1, w) \in F_1$ and $(\hat{\delta}_2(q_2, w) \in F_2$

i.e., regular language is close under difference.

iv) Reversal and letter substitution

$$L(E^R) = (L(E))^R$$

Refer Q.No. 3b. of MQP - 2.

c. State and prove pumping lemma for regular language. **(04 Marks)**

Ans. Refer Q.No. 3b. of MQP - 1.

Module - 3

5. a. Define a context free grammar. Obtain the grammar to generate the language L= {W} $n_a(w) = n_b(w)$} **(04 Marks)**

Ans. $S \rightarrow \varepsilon$

$S \rightarrow asb$

$S \rightarrow bsa$

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

$V = \{S\}$

$T = \{a, b\}$

$P = \{\delta \rightarrow \varepsilon$

$\delta \rightarrow asb$

$\delta \rightarrow bsa$

}

S is start symbol

b. For the regular expression (011 + 1)* (01)* obtain the context free grammar. **(04 Marks)**

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

$V = \{S, A\}$

$T = \{0, 1\}$

P = {

$S \rightarrow \varepsilon \mid 011A \mid 01S$

$A \rightarrow 1S \mid \varepsilon$

}

's' is start symbol.

c. What is ambiguity? Show that the following grammar is ambigugous.

$S \rightarrow aB \mid bA$

$A \rightarrow as \mid bAA \mid a$

$B \rightarrow bs \mid aBB \mid b$. **(08 Marks)**

Ans. A grammar G is ambiguous if and only if there exists atleast one string $w \in T * \phi o \rho$ which two or more different parse trees exist by applying either LMD or RMD.

$S \Rightarrow aB$

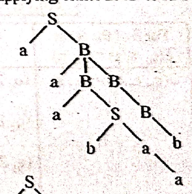
$S \Rightarrow aaBB$

$S \Rightarrow aabSB$

$S \Rightarrow aabbAB$

$S \Rightarrow aabbaB$

$S \Rightarrow aabbab$



$S \Rightarrow aB$

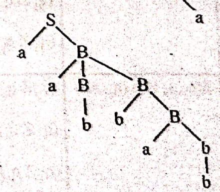
$S \Rightarrow aaBB$

$S \Rightarrow aabB$

$S \Rightarrow aabbS$

$S \Rightarrow aabbaB$

$S \Rightarrow aabbab$



Two parse tree for same string

So grammar is ambiguous.

OR

6. a. Define PDA. Obtain to accept the language $l(M) = \{w\ C\ w^R\} \mid w \in (a, b)8\}$, where w^R is reverse of w by a final state. **(08 Marks)**

Ans. Refer Q.No. 6.. of MQP - 1

b. For the grammar

$S \rightarrow aABB \mid aAA$
$A \rightarrow aBB \mid a$
$B \rightarrow bBB \mid A$
$C \rightarrow a$

Obtain the corresponding PDA. **(04 Marks)**

Ans. Refer Q.No. 6.c. of Dec 2017 / Jan 2018

c. Obtain a CFG for the FDA shown below:

$f(q_0, a, Z) = (q_0, AZ)$
$f(q_0, a, A) = (q_0, A)$
$f(q_0, b, A) = (q, \varepsilon)$
$f(q_1, \varepsilon, Z) = (q_2, \varepsilon)$. **(04 Marks)**

Ans.

For δ of the form $\delta(q_i, a, z) = (q_j, \varepsilon)$	Resulting production $(q_i\ Z\ q_j) \rightarrow a$
$\delta(q_0, a, A) = (q_3, \varepsilon)$ $\delta(q_0, b, A) = (q_1, \varepsilon)$ $\delta(q_1, \varepsilon, z) = (q_2, \varepsilon)$	$(q_0\ A\ q_3) \rightarrow a$ $(q_0\ A\ q_1) \rightarrow b$ $(q_1\ Z\ q_2) \rightarrow \varepsilon$

For δ of the form $\delta(q_i, a, z) = (q_j, \varepsilon)$	Resulting production $(q_i\ Z\ q_j) \rightarrow a$
$\delta(q_0, a, A) = (q_3, \varepsilon)$	$(q_0\ Z\ q_0) \rightarrow a(q_0\ A\ q_0)(q_0\ Z\ q_0) \mid a(q_0\ A\ q_1)(q_1\ Z\ q_0) \mid$ $a(q_0\ A\ q_2)(q_2\ Z\ q_0) \mid a(q_0\ A\ q_3)(q_3\ Z\ q_0)$ $(q_0\ Z\ q_1) \rightarrow a(q_0\ A\ q_0)(q_0\ Z\ q_1) \mid a(q_0\ A\ q_1)(q_1\ Z\ q_1) \mid$ $a(q_0\ A\ q_2)(q_2\ Z\ q_1) \mid a(q_0\ A\ q_3)(q_3\ Z\ q_1)$ $(q_0\ Z\ q_2) \rightarrow a(q_0\ A\ q_0)(q_0\ Z\ q_2) \mid a(q_0\ A\ q_1)(q_1\ Z\ q_2) \mid$ $a(q_0\ A\ q_2)(q_2\ Z\ q_2) \mid a(q_0\ A\ q_3)(q_3\ Z\ q_2)$ $(q_0\ Z\ q_3) \rightarrow a(q_0\ A\ q_0)(q_0\ Z\ q_3) \mid a(q_0\ A\ q_1)(q_1\ Z\ q_3) \mid$ $a(q_0\ A\ q_2)(q_2\ Z\ q_3) \mid a(q_0\ A\ q_3)(q_3\ Z\ q_3)$
$\delta(q_3, \varepsilon, z) = (q_0, Az)$	$(q_3\ Z\ q_0) \rightarrow (q_0\ A\ q_0)(q_0\ Z\ q_0) \mid (q_0\ A\ q_1)(q_1\ Z\ q_0) \mid$ $(q_0\ A\ q_2)(q_2\ Z\ q_0) \mid (q_0\ A\ q_3)(q_3\ Z\ q_0)$ $(q_3\ Z\ q_1) \rightarrow (q_0\ A\ q_0)(q_0\ Z\ q_1) \mid (q_0\ A\ q_1)(q_1\ Z\ q_1) \mid$ $(q_0\ A\ q_2)(q_2\ Z\ q_1) \mid (q_0\ A\ q_3)(q_3\ Z\ q_1)$ $(q_3\ Z\ q_2) \rightarrow (q_0\ A\ q_0)(q_0\ Z\ q_2) \mid (q_0\ A\ q_1)(q_1\ Z\ q_2) \mid$ $(q_0\ A\ q_2)(q_2\ Z\ q_2) \mid (q_0\ A\ q_3)(q_3\ Z\ q_2)$ $(q_3\ Z\ q_3) \rightarrow (q_0\ A\ q_0)(q_0\ Z\ q_3) \mid (q_0\ A\ q_1)(q_1\ Z\ q_3) \mid$ $(q_0\ A\ q_2)(q_2\ Z\ q_3) \mid (q_0\ A\ q_3)(q_3\ Z\ q_3)$

Module-4

7. a. Consider the grammar

$S \rightarrow 0A \mid 1B$
$A \rightarrow 0AA \mid 1S \mid 1$
$B \rightarrow 1BB \mid 0S \mid 0$

Obtain the grammar in CNF. **(08 Marks)**

Ans. $A \rightarrow 1$
$B \rightarrow 0$

Given production	Action	Resulting production
$S \rightarrow 0A \mid 1B$	Replace 0 by B_0 $B_0 \rightarrow 0$ Replace 1 by B_1 $B_1 \rightarrow 1$	$S \rightarrow B_0A \mid B_1B$ $B_0 \rightarrow 0$ $B_1 \rightarrow 1$
$A \rightarrow 0AA \mid 1S$	Replace 0 by B_0 $B_0 \rightarrow 0$ Replace 1 by B_1 $B_1 \rightarrow 1$	$A \rightarrow B_0AA \mid B_1S$ $B_0 \rightarrow 0$ $B_1 \rightarrow 1$
$B \rightarrow 1BB \mid 0S$	Replace 0 by B_0 $B_0 \rightarrow 0$ Replace 1 by B_1 $B_1 \rightarrow 1$	$B \rightarrow B_1BB \mid B_0S$ $B_1 \rightarrow 1$ $B_0 \rightarrow 0$

Consider $A \rightarrow B_0AA$ and $B \rightarrow B_1BB$

$A \rightarrow B_0AA \Rightarrow A \rightarrow B_0D_1$
$D_1 \rightarrow AA$

$B \rightarrow B_1BB \Rightarrow B \rightarrow B_1D_2$
$D_2 \rightarrow BB$

$G^1 = (V^1, T, P^1, S)$ is in CNF where

$V_1 = \{S, A, B, B_0, B_1, D_1, D_2\}$

$T_1 = \{0, 1\}$

$P_1 = \{$

$S \rightarrow B_0A \mid B_1B$
$A \rightarrow B_1S \mid 1 \mid B_0D_1$
$B \rightarrow B_0S \mid 1 \mid B_1D_2$
$B_0 \rightarrow 0$
$B_1 \rightarrow 1$
$D_1 \rightarrow AA$
$D_2 \rightarrow BB$
$\}$

'S' is the start symbol

b. Show that $L = \{a^nb^nc^n \mid n \geq 0\}$ is not context free. **(08 Ma**

Ans. Refer Q.No. 8.b. of MQP - 2

OR

8. a. **With a neat diagram, explain the working of a basic Turing machine.** **(04 Marks)**

Ans. Refer Q.No. 9.a. of MQP - 2

b. **Obtain a Turing machine to accept the language $L = \{0^n 1^n \mid \eta \geq 1\}$.** **(08 Marks)**

Ans. Refer Q.No. 9.a. of MQP - 1

c. **Briefly explain the techniques for TM construction.** **(04 Marks)**

Ans. Refer Q.No. 9.b.(i) of MQP - 1

Module-5

9. a. **Obtain a Turing machine to recognize the language $L = \{0^n 1^n 2^n \mid \eta \geq 1\}$.** **(08 Marks)**

Ans.

States	0	1	2	Z	Y	X	B
q_0	q_1, x, R				q_4, Y, R		
q_1	q_1, 0, R	q_2, Y, R			q_1, Y, R		
q_2		q_2, 1, R	q_3, Z, L	q_2, Z, R			
q_3	q_3, 0, L	q_3, 1, L		q_3, Z, L	q_3, Y, L	q_0, X, R	
q_4				q_5, Z, R	q_4, Y, R		
q_5				q_5, Z, R			q_6, B, R
q_6							

b. **Prove that $HALT_{TM}$ = {(M,W)| the Turing machine M halts on input W} is undecidable.** **(04 Marks)**

Ans. Refer Q.No. 10.a. of MQP - 1

c. **With example, explain the quantum computation.** **(04 Marks)**

Ans. Refer Q.No. 10.b.(ii) of MQP - 2

OR

10. **Write a short note on:**

a. **Multiple Turing machine**

b. **Non deterministic Turing machine**

c. **The model of linear bounded automaton**

d. **The post correspondence problem.** **(16 Marks)**

Ans. a) Refer Q.No. 9.b.(i) of MQP - 1

b) Refer Q.No. 9.b. of Dec 2017 / Jan 2018

c) Refer Q.No. 10.b. of MQP - 1

d) Refer Q.No. 10.b. of MQP - 2

OR

8. a. With a neat diagram, explain the working of a basic Turing machine. **(04 Marks)**

Ans. Refer Q.No. 9.a. of MQP - 2

b. Obtain a Turing machine to accept the language $L = \{0^n 1^n \mid n \geq 1\}$. **(08 Marks)**

Ans. Refer Q.No. 9.a. of MQP - 1

c. Briefly explain the techniques for TM construction. **(04 Marks)**

Ans. Refer Q.No. 9.b.(i) of MQP - 1

Module-5

9. a. Obtain a Turing machine to recognize the language $L = \{0^n 1^n 2^n \mid n \geq 1\}$. **(08 Marks)**

Ans.

States	0	1	2	Z	Y	X	B
q_0	q_1, x, R				q_4, Y, R		
q_1	q_1, 0, R	q_2, Y, R			q_1, Y, R		
q_2		q_2, 1, R	q_3, Z, L	q_2, Z, R			
q_3	q_3, 0, L	q_3, 1, L		q_3, Z, L	q_3, Y, L	q_0, X, R	
q_4				q_5, Z, R	q_4, Y; R		
q_5				q_5, Z, R			q_6, B, R
q_6							

b. Prove that $HALT_{TM}$ ={(M,W)| the Turing machine M halts on input W} is undecidable. **(04 Marks)**

Ans. Refer Q.No. 10.a. of MQP - 1

c. With example, explain the quantum computation. **(04 Marks)**

Ans. Refer Q.No. 10.b.(ii) of MQP - 2

OR

10 Write a short note on:

a. Multiple Turing machine

b. Non deterministic Turing machine

c. The model of linear bounded automaton

d. The post correspondence problem. **(16 Marks)**

Ans. a) Refer Q.No. 9.b.(i) of MQP - 1

b) Refer Q.No. 9.b. of Dec 2017 / Jan 2018

c) Refer Q.No. 10.b. of MQP - 1

d) Refer Q.No. 10.b. of MQP - 2

Ans.

Given productions	Action	Resulting production
$S \to 0A \mid 1B$	Replace 0 by B_0 and introduce production $B_0 \to 1$ Replace 0 by B_1 and introduce the production $B_1 \to 1$	$S \to B_0A \mid B_1B$ $B_0 \to 0$ $B_2 \to 1$
$A \to 0AA / 1S$	Replace 0 by B_0 and introduce $B_0 \to 0$ Replace 1 by B_1 and introduce $B_1 \to 1$	$A \to B_0AA \mid B_1S$ $B_0 \to 0$ $B_1 \to 1$
$B \to 1BB/0S$	Replace 0 by B_0 and introduce $B_0 \to 0$ Replace 1 by B_1 and introduce $B_1 \to 1$	$B \to B_1BB / B_0S$ $B_1 \to 1$ $B_0 \to 0$

$G=(V_1,T_1,P_1,S)$

$V_1=\{S,A,B,B_0,B_1\}$

$T_1=\{0,1\}$

$P_1=\{$

$S \to B_0A \mid B_1B$

$A \to B_0AA \mid B_1S \mid 1$

$B \to B_1BB \mid B_0S \mid 0$

$B_0 \to 0$

$B_1 \to 1$

$\}$

S is the start symbol

Finally

$G^1=(V^1,T^1,P^1,S)$ in CNF

$V_1=\{S,A,B,B_0,B_1,D_1,D_2\}$

$T_1=\{0,1\}$

$P_1=\{$

$S \to B_0A \mid B_1B$

$A \to B_1S \mid 1 \mid B_0D_1$

$B \to B_0S \mid 0 \mid B_1D_2$

$B_0 \to 0$

$B_1 \to 1$

$D_1 \to AA$

$D_2 \to BB$

$\}$

S is the start symbol

Module - 5

9. a. Obtain a turning machine to accept the language **(08 Marks)**

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

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

Where $Q=\{q_0,q_1,q_2,q_3\}$

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

$\Gamma=\{0,1,x,y,B\}$

$q_0 \in Q$ is the start state of machine

$B \in \Gamma$ is the blank symbol

$F=\{q_4\}$ is the final state

δ is shown below:

$\delta(q_0,0)=(q_1,X,R)$

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

$\delta(q_1,y)=(q_1,y,R)$

$\delta(q_1,1)=(q_2,Y,L)$

$\delta(q_2,Y)=(q_2,Y,L)$

$\delta(q_2,0)=(q_2,0,L)$

$\delta(q_2,X)=(q_0,X,R)$

$\delta(q_0,Y)=(q_3,Y,R)$

$\delta(q_3,Y)=(q_3,Y,R)$

$\delta(q_3,B)=(q_4,B,R)$

S	Tape symbols (Γ)				
States	0	1	X	Y	B
q_0	(q,X,R)	–		(q_3,Y,R)	
q_1	$(q_1,0,R)$	(q_2,Y,L)		(q_1,Y,R)	
q_2	(q_2,o,L)	–	(q_0,X,R)	(q_2,Y,L)	
q_3				(q_3,Y,L)	(q_4,B,R)
q_4					

b. Explain

I. turing machine as multi-stack machines

II. semi-infinite tape **(08 Marks)**

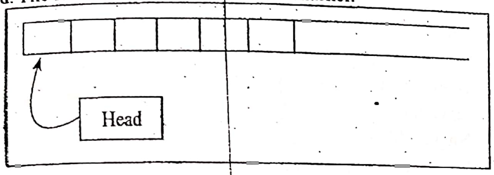
Ans. I. Turing machine as multi-stack machines:- Multi-stack/track Turing machines, a specific type of Multi-tape Turing machine, contain multiple tracks but just one tape head reads and writes on all tracks. Here, a single tape head reads n symbols from n tracks at one step. It accepts recursively enumerable languages like a normal single-track single-tape Turing Machine accepts.

A Multi-track Turing machine can be formally described as a 6-tuple $(Q, X, \Sigma, \delta, q_0, F)$ where

- Q is a finite set of states
- X is the tape alphabet
- Σ is the input alphabet
- δ is a relation on states and symbols where $\delta(Q_i, [a_1, a_2, a_3,....]) = (Q_j, [b_1, b_2, b_3,....],$ Leftshift or Rightshift)
- q_0 is the initial state
- F is the set of final states

Note - For every single-track Turing Machine S, there is an equivalent multi-track Turing Machine M such that **L(S) = L(M)**.

II. **Semi- Infinite tape:-A** Turing Machine with a semi-infinite tape has a left end but no right end. The left end is limited with an end marker.



It is a two-track tape -

Upper track - It represents the cells to the right of the initial head position.

Lower track - It represents the cells to the left of the initial head position in reverse order.

The infinite length input string is initially written on the tape in contiguous tape cells. The machine starts from the initial state qo and the head scans from the left end marker 'End'.

In each step, it reads the symbol on the tape under its head. It writes a new symbol on that tape cell and then it moves the head either into left or right one tape cell. A transition function determines the actions to be taken.

It has two special states called **accept state** and **reject state.** If at any point of time it enters into the accepted state, the input is accepted and if it enters into the reject state, the input is rejected by the TM. In some cases, it continues to run infinitely without being accepted or rejected for some certain input symbols.

Note - Turing machines with semi-infinite tape are equivalent to standard Turing machines.

OR

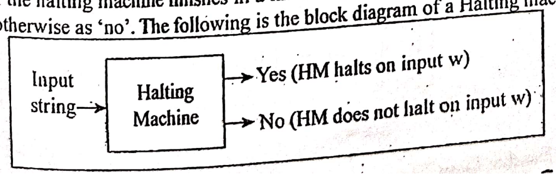
10. a.Explain Halting Problem **(08 marks)**

Ans. Halting Problem:

Input - A Turing machine and an input string w.

Problem - Does the Turing machine finish computing of the string win a finite number of steps? The answer must be either yes or no.

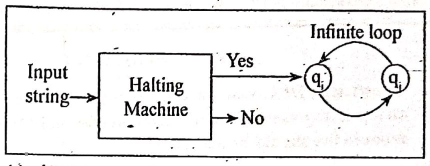
Proof - At first, we will assume that such a Turing machine exists to solve this problem and then we will show it is contradicting itself. We will call this Turing machine as a **Halting machine** that produces a 'yes' or 'no' in a finite amount of time. If the halting machine finishes in a finite amount of time, the output comes as 'yes', otherwise as 'no'. The following is the block diagram of a Halting machine -



Now we will design an **inverted halting machine (HM)'** as -

- If **H** returns YES, then loop forever.
- If H returns NO, then halt.

The following is the block diagram of an 'Inverted halting machine' -



Further, a machine $(HM)_2$ which input itself is constructed as follows -

- If $(HM)_2$ halts on input, loop forever.
- Else, halt.

Here, we have got a contradiction. Hence, the halting problem is **undecidable.**

b. Explain Linear bounded automata with respect to hiring machines (08 Marks)

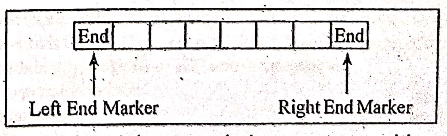
Ans. A linear bounded automaton is a multi-track non-deterministic Turing machine with a tape of some bounded finite length.

Length = function (Length of the initial input string, constant c) Here,
Memory information ≤ c * Input information

The computation is restricted to the constant bounded area. The input alphabet contains two special symbols which serve as left end markers and right end markers which mean the transitions neither move to the left of the left end marker nor to the right of the right end marker of the tape.

A linear bounded automaton can be defined as an 8-tuple (Q, X, X, Qo, ML, MR, δ, F) where -

- Q is a finite set of states
- X is the tape alphabet
- Σ is the input alphabet
- q_0 is the initial state
- M_L is the left end marker
- MR is the right end marker where M_R ^ M_L
- δ is a transition function which maps each pair (state, tape symbol) to (state, tape symbol, Constant 'c') where c can be 0 or +1 or -1
- F is the set of final states



A deterministic linear bounded automaton is always **context-sensitive** and the linear bounded automaton with empty language is **undecidable.**

$A_1 \to A_2A_3$

$A_2 \to A_3A_1 | b$

$A_3 \to bA_3A_2B_3 | aB_3 | bA_3A_2 | a$

$B_3 \to A_1A_3A_2 | A_1A_3A_2B_3$

Now all the production for A_3 are all in GNF

$A_1 \to A_2A_3$

$A_2 \to bA_3A_2B_3A_1 | aB_3A_1 | bA_3A_2A_1 | aA_1 | b$

$A_3 \to bA_3A_2B_3 | aB_3 | bA_3A_2 | a$

$B_3 \to A_1A_3A_2 | A_1A_3A_2B_3$

An equivalent grammar in GNF thus can be written as

$A_1 \to bA_3A_2B_3A_1A_3 | aB_3A_1A_3 | bA_3A_2A_1A_3 | A_1A_3 | bA_3$

$A_2 \to bA_3A_2B_3A_1 | aB_3A_1 | bA_3A_2A_1 | aA_1 | b$

$A_3 \to bA_3A_2B_3 | aB_3 | bA_3A_2 | a$

$B_1 \to bA_3A_2B_3A_1A_3A_3A_2 | bA_3A_2B_3A_1A_3A_3A_2B_3$

$aB_3A_1A_3A_3A_2 | aB_3A_1A_3A_3A_2B_3 | bA_3A_2A_1A_3A_3A_2 | bA_3A_2A_1A_3A_3A_2B_3 |$

$aA_1A_3A_3A_2 | aA_1A_3A_3A_2B_3 | bA_3A_3A_2 | bA_3A_3A_2B_3 |$

b. Show that L= {w|w∈{a,b,c}}* where $n_a(w) = n_b(w) = n_c(w)$} is not context free. **(08 Marks)**

Ans. The language $L_1 = \{a^n b^n c^n \mid n \geq 0\}$ is obtained by the intersection Land the regular language represented by the regular a*b*c* i.e.,

$\{a^nb^nc^n \mid n \geq 0\} = \{a^*b^*c^* \cap \{w \mid w \in \{a,b,c\}\}^* \text{ where } \eta_a(w) = \eta_b(w) = \eta_c(w)\}$

We know that intersection of context free language and regular language is also a context free. But its already known that

$L_1 = \{a^nb^nc^n \mid n \geq 0\}$ is not context free. Since L_1 is not context free, it implies that the given language

$L = \{w \mid w \in \{a,b,c\}^* \text{ where } \eta_a(w) = \eta_b(w) = \eta_c(w)\}$ is not context free and not context free grammer.

Module - 5

9. a. Explain the concept of turing machine in detail. **(08 Marks)**

Ans. A Turing Machine (TM) is a mathematical model which consists of an infinite length tape divided into cells on which input is given. It consists of a head which reads the input tape. A state register stores the state of the Turing machine. After reading an input symbol, it is replaced with another symbol, its internal state is changed, and it moves from one cell to the right or left. If the TM reaches the final state, the input string is accepted, otherwise rejected. A TM can be formally described as a 7-tuple

Next run MP on x. Accept iff MP does.
Note that C(M,w) accepts the same language as MP if M accepts w; C(M,w) accepts the empty language if M does not accept w.
Thus if M accepts w the Turing machine C(M,w) has the property P, and otherwise it doesn't. Feed the description of C(M,w) to B. If B accepts, accept the input (M,w); if B rejects, reject.

b. Define

i) Post Correspondence problem & ii) Quantum computation (08 Marks)

Ans. i) Post Correspondence problem

The Post Correspondence Problem (PCP), introduced by Emil Post in 1946, is an undecidable decision problem. The PCP problem over an alphabet £ is stated as follows - Given the following two lists, M and N of non-empty strings over £ ~

$M = (x_1, x_2, x_3, \quad , x_n)$

$N = (y_1, y_2, y_3, \quad , y_n)$

We can say that there is a Post Correspondence Solution, if for some $i_1, i_2 ...... i_k$ ik, where $1 < i_j < n$, the condition $X_{i1} ...... X_{ik} = Y_{i1} ...... Y_{ik}$.

Find whether the lists

M = (abb, aa, aaa) and N = (bba, aaa, aa) have a Post Correspondence Solution?

	X_1	X_2	X_3
M	Abb	aa	aaa
N	Bba	aaa	aa

Here, $X_2X_1X_3$ = 'aaabbaaa' and $y_2y_1y_3$ = 'aaabbaaa'

We can see that $X_2X_1X_3 = y_2y_1y_3$ Hence, the solution is $i = 2$, $j = 1$, and $k = 3$

ii) Quantum computation

Quantum computing studies theoretical computation systems (quantum computers) that make direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data. Quantum computers are different from binary digital electronic computers based on transistors. Whereas common digital computing requires that the data be encoded into binary digits (bits), each of which is always in one of two definite states (0 or 1), quantum computation uses quantum bits, which can be in superpositions of states. A quantum Turing machine is a theoretical model of such a computer, and is also known as the universal quantum computer. The field of quantum computing was initiated by the work of Paul Benioff and Yuri Man in in 1980, Richard Feynman in 1982, and David Deutsch in 1985. A quantum computer with spins as quantum bits was also formulated for use as a quantum spacetime in 1968.

As of 2017, the development of actual quantum computers is still in its infancy, but experiments have been carried out in which quantum computational operations were executed on a very small number of quantum bits. Both practical and theoretical research continues, and many national governments and military agencies are funding quantum computing research in an effort to develop quantum computers for civilian, business, trade, environmental and national security purposes, such as cryptanalysis

OR

6\. a. Is the following grammar ambiguous? (08 Marks)

S → aB | bA

A → as | bAA|a

B → bS|aBB|b

Ans. Left most derivation is

S ⇒ aB

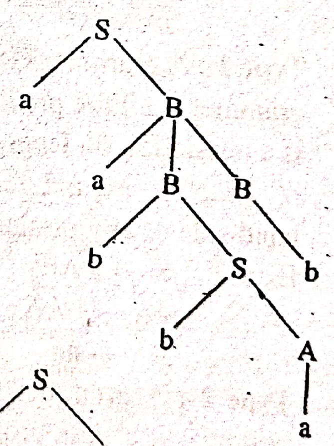
⇒ aaBB

⇒ aabSB

⇒ aabbAB

⇒ aabbaB

⇒ aabbab



Left most derivation

S ⇒ aB

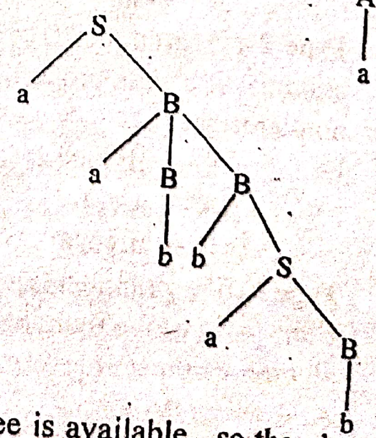
⇒ aaBB

⇒ aabB

⇒ aabbS

⇒ aabbaB

⇒ aabbab



For string aabbab, more than one parse tree is available , so the given grammar is ambiguous.

Sunstar Exam Scanner

b. Obtain a PDA to accept the language $L(m) = \{wCw^R \mid w \in (a+b)^*\}$. (08 Marks)

Ans. $M = (Q, \Sigma, \Gamma, \delta, Q_0, Z_0, f]$

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

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

$\Gamma = \{a, b, Z_0\}$

δ:{

$\delta(q_0, a, z_0) = (q_0, aZ_0)$

$\delta(q_0, b, Z_0) = (q_0, b, Z_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_0, ba)$

$\delta(q_0, a, b) = (q_0, ab)$

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

$\delta(q_0, c, a) = (q_1, a)$

$\delta(q_0, c, a) = (q_1, a)$

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

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

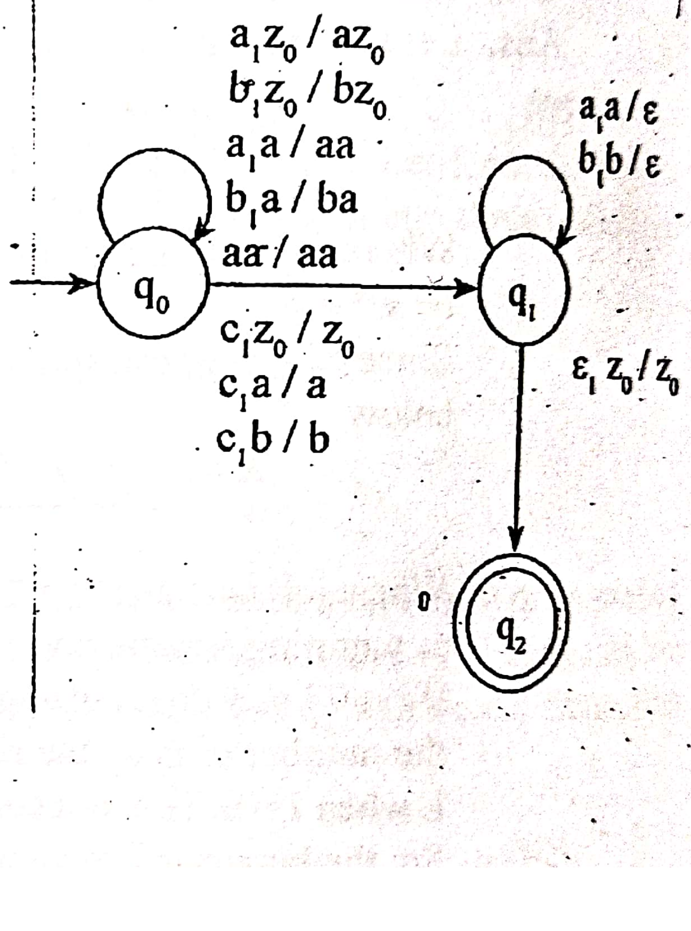
$\delta(q_1, \varepsilon, Z_0) = (q_1 Z_0)$

}

$q_0 \in Q$ is start state

$z_0 \in \Gamma$ is the initial stack symbol

$F = \{q_2\}$ is final state



b. State and prove pumping lemma for regular languages. **(08 Marks)**

Ans. Let $M = (Q,\Sigma,\delta,q_o,F)$ be an FA and has 'η' number of states. Let L be the regular language accepted by M. Let frequency string x can be broken into three substrings u,v and w such that

$$X = uvw$$

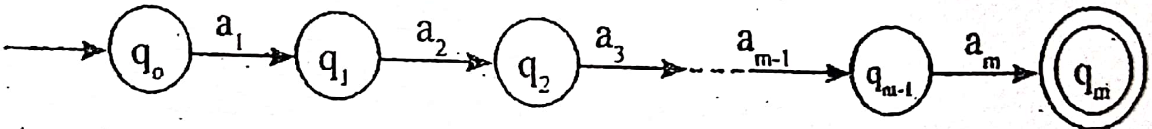
Satisfying the following constraints :

$V \neq \varepsilon$ ie., $|V| \geq 1$

$|uv| \leq \eta$

Then $uv^i w$ is in L for $i \geq 0$

Let $X = a_1a_2a_3 \ldots a_m$ where $m \geq n$ and each a_2 is in Σ. Here, n represent the states of DFA. Since we have m input symbols, naturally we should have m+1 states in the sequence $q_o, q_1, q_2 \ldots q_m$, where q_o will be the start state and q_m will be the final state as shown below.

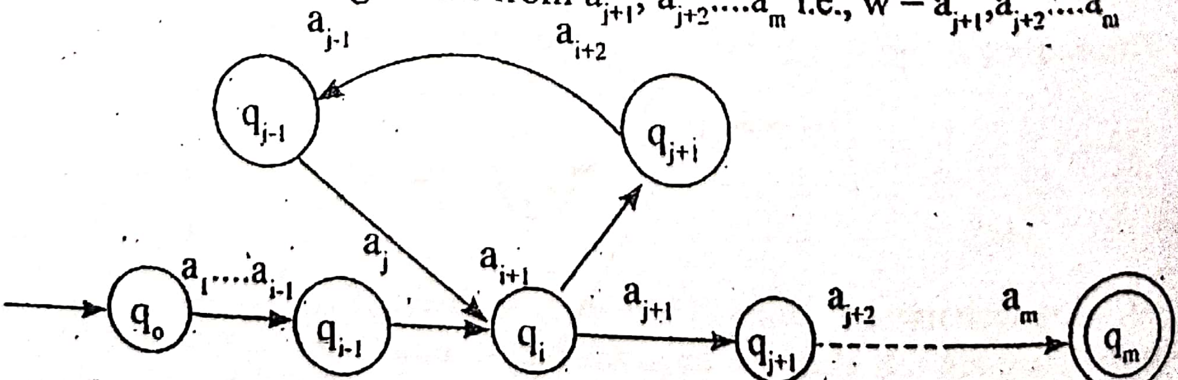


Since $|x| \geq n$, by the pigeon hole principle it is not possible to have distinct transitions. Once of the state can have a loop. Let the string x is divided into three substrings as shown below.

The first group is the string prefix from $a_1a_2 \ldots a_i$ i.e., $u = a_1a_2 \ldots a_i$

The second group is the loop string from $a_{i+1}, a_{i+2} \ldots a_{j-1}$ i.e., $V = a_{i+1}, a_{i+2} \ldots a_{j-1}a_j$

The third group is the string suffix from $a_{j+1}, a_{j+2} \ldots a_m$ i.e., $w = a_{j+1}, a_{j+2} \ldots a_m$



Observe from above figure that, the prefix string u taxes the machine from q_o to q_i, the loop string v takes the machine from q_i to q_i (Note $q_i = q_j$) and suffix string w toches the machine from q_j to q_f. The minimum string that can be accepted by the above FA is uw with $i = 0$.