Page No.
Date

Name : Kolani Karan Giyan.
Semester : 5 Subject : TCS.
Seat No : CS5A021 sign : K.G.Kalani
pg No : 1/9. of Q3

A)
i). Let A and B be two non empty lists of strings over Σ A and B are given as below :-

$A : \{x_1, x_2, x_3, \ldots x_k\}$.

$B : \{y_1, y_2, y_3, \ldots y_k\}$.

There is a post correspondence between A and B if there is sequence of one or more integers $i, J, k \ldots m$ such that :-

The string $x_i, x_J, \ldots x_m$ is equal to $y_i, y_J, \ldots y_m$

Example :- PCP with two lists :-

$A = \{a, aba^3, ab\}$ and $B = \{a^3, ab, b\}$.

have a solution ?.

To find a sequence using which when the elements of A and B are listed, will produce identical strings.

The required sequence is (2, 1, 1, 3)

~~A_2, A_1, A~~ $A_2 A_1 A_1 A_3$ = $aba^3 a a ab$ = $aba^6 b$

$B_2 B_1 B_1 B_3$ = $ab a^3 a^3 b$ = $aba^6 b$

Thus PCP has solution.

So accept the undecidability of post correspondence problem without proof.

Teacher's Sign.: ____________

Pg : 2/9 of Q3. sign K. G. Kalani

i). Example :
Determining the solution for following instance of PCP.

	List A	List B.
i	w_i	x_i
1	01	0
2	110010	0
3	1	1111
4	11	01

PCP has a solution. The required sequence is (1, 3, 2, 4, 4, 3)

~~w_1, w~~ $w_1 w_2 w_3 w_4 w_4 w_3$ = 01111001011111

$x_1 x_2 x_2 x_4 x_4 x_3$ = 01111001011111

iii). Universal Turing Machine.
A general purpose computer can be programmed to solve different types of problems. A TM can also behave like a general purpose computer. A general purpose computer solves a problem as given below :-
i) A program is written in a high level language and its machine code is obtained with help of compiler.
ii) Machine code is located in main memory.
iii) Input to the program can also be loaded in memory

P.T.O.

Pg :- 3/9 of Q3. sign :- K.G. Kalam.

(i) Program stored in memory is executed line by line. Execution involves reading a line of code pointed by IP, decoding the code and executing it.

We can follow a similar approach for a TM. Such a TM is known as Universal Turing machine. UTM can solve all sorts of problems.

A Turing Machine (M) is designed to solve a particular problem (P) can be specified as.

i. Initial state q_0 of TM. M.

ii. The transition function δ of M can be specified as given

If the current state of M is q_i and the symbol under head is a_i the machine moves to state q_j while changing a_i to a_j. The move of tape head may be :-

a) to left.

b) to right.

c) Neutral

Such a move of TM can be represented as Tuple

$(q_i, a_j, a_j, q_i, m_r)$: $q_i q_j \in$

Q :- $a_i . a_j \in$

r : m_r ~~∈ {To le~~

$\in$: {to left, to right, Neutral}.

Page No.:
Date:

Pg :- ~~8~~ 4/9 of Q3 Sign : K.G. Kalaw.

~~iii~~.

iii) UTM should be able to simulate every turing machine. Simulations of Turing will involve
a) Encoding behaviour of particular TM as program.
b) Execution of above program by UTM.

A move of form $(q_i, a_i ; q_j, a_j, m_f)$ can be represented as $10^{i+1}, 10^{1}, 10^{i+1}\ 10^{j}\ .10^{k}$, where

$k = 1$, if move to left
$K = 2$, if move to right
$K = 3$, if move is 'no move'

state q_0 is represented by 0
state q_1 is represented by 00.
State q_n is represented by 0^{n+1}.
First symbol can be represented by 0.
Second symbol can be represented by 00 and so on.
Two elements of tuple representing to move are seperated by 1.
Two moves are seperated by 1.

Execution of UTM :-
we can assume UTM as a 3-tape turing machine.

1. Input is written on first page.

PTO

sign :- K. G. Kalam

ii. Moves of TM is encoded form is written on Second page.

iii) The current state of TM is written on third page.

The control unit of UTM by counting numbers of 0's between 1's ~~or~~ can find out the current symbol under head. It can find current state from tape 3. Now it can locate the appropriate move based on current input and current state from tape 2.
Now the control unit can extract the following information from tape 2:-

i. Next state.

ii. Next symbol to be written.

iii. Move of head.

Based on information, the control unit can take the appropriate action.

Q.3.
B)
ii) Design PDA for $L = \{ a^{2n} b^n , n \geq 1 \}$.

step 1 :- CFG :- $S \to aaSb \mid aab$.

n=1 aab.

n=2 aaaabb.

Step 2 :- CFG to GNF

$S \to aC_1SC_2 \mid aC_1C_2$

$C_1 \to a$

$C_2 \to b$.

step 3 :- PDA $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = \{q_0\}$. $q_0 = q_0$

$\Sigma = \{a, b\}$. $Z_0 = S$

$\Gamma = \{S, C_1, C_2)$ $F = \{\}$.

$\delta =$

$S \to aC_1SC_2$	$\delta(q_0, a, S) = \{(q_0, C_1SC_2)\}$
$S \to aC_1C_2$	$\delta(q_0, a, S) = \{(q_0, C_1C_2)\}$
$C_1 \to a$	$\delta(q_0, a, C_1) = \{(q_0, \epsilon)\}$
$C_2 \to b$.	$\delta(q_0, b, C_2) = \{(q_0, \epsilon)\}$

step 4 :- (q_0, aab, S)

$\vdash (q_0, ab, C_1SC_2)$ $\vdash (q_0, ab, C_1C_2)$

$\vdash (q_0, b, SC_2)$ $\vdash (q_0, b, C_2)$

$\vdash (q_0, \epsilon, \epsilon)$

accept.

B

i) $\delta(q_0, a, S) = \{(q_0, S C_1 C_2)\}$.

$\delta(q_0, a, S) = \{(q_0, C_2 C_1 C_1 C_1)\}$.

$\delta(q_0, b, C_1) = \{(q_0, \epsilon)\}$.

$\delta(q_0, a, C_2) = \{(q_0, \epsilon)\}$.

$(q_0, aabbb, S)$

$\vdash (q_0, abbb, S C_1 C_2)$ $\quad\quad$ $\vdash (q_0, abbb, C_2 C_1 C_1 C_1)$

$\vdash (q_0, bbb, S C_1 C_1 C_1 C_1)$ $\quad$ $\vdash (q_0, bbb, C_2 C_1 C_1 C_1 C_1 C_1)$ $\quad$ $\vdash (q_0, bbb, C_1 C_1 C_1)$

$\vdash (q_0, bb, C_1 C_1)$

$\vdash (q_0, b, C_1)$

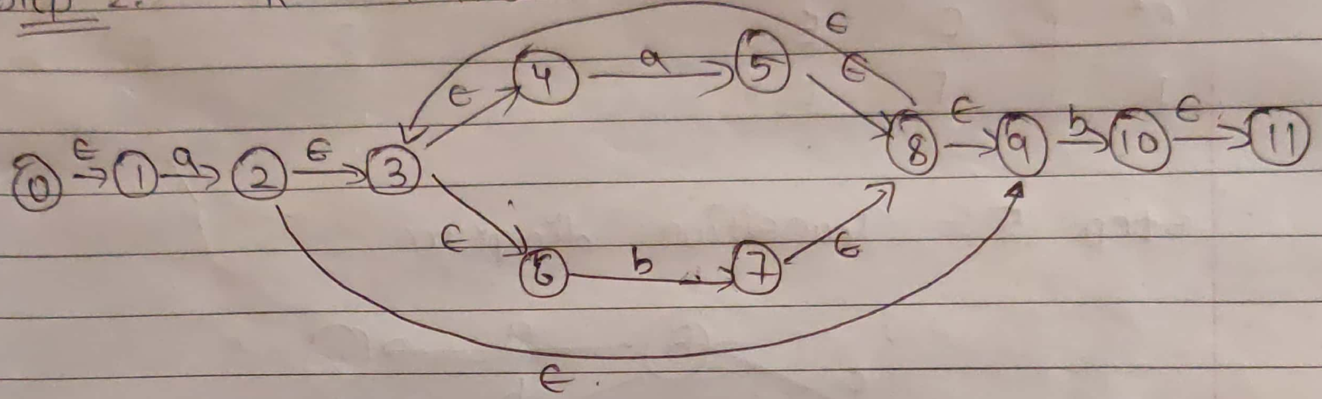
$\vdash (q_0, \epsilon, \epsilon)$

Accept.

pg :- 8/9 of Q3. sign : [signature]

B)
9). R.E = $a(a+b)^* b$ to minimized DFA.

Step 2 :- R = $a(a+b)^* \cdot b$.



M = $(Q, \Sigma, \delta, q_0, f)$

Q = is a set of stats.

Σ is a set of alphabets.

$q_0 \in q$ is the initial state.

$F \subseteq Q$ is set of final state and

transition function is a function time from $Q \times \Sigma$ to Q.

Step 3 :- Implementation.

	X.	Y = ∈ (closure of X)	δ(y, a)	δ(y, b).
A	{0}	{1}.	{2}	{ }
B	{2}	{2, 3, 4, 6, 9}.	{5}	{7, 10}
C	{5}	{3, 4, 5, 6, 8}	{5}	{7}
D	{7}	{3, 4, 6, 7, 8}.	{5}	{7}
E*	{7, 10}	{3, 4, 6, 7, 8, 9, 10, 11}	{5}	{7, 10}
F	{ }.	{ }	{ }	{ }

step 4 :-

Minimized Table.

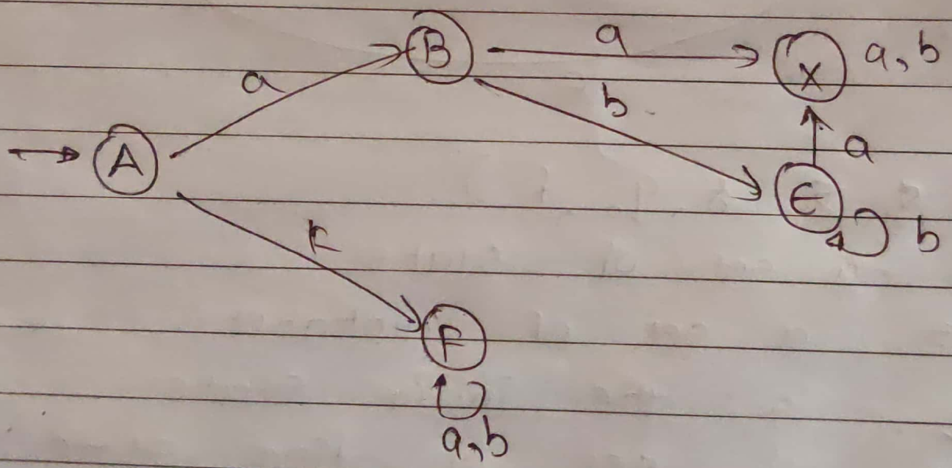
	a	b
A	B	F
B	C	E
C	C	D
D	C	D
E*	C	E
F	F	F

]-X

Minimized DFA.

	a	b
A	B	F
B	X	E
X	X	X
E	X	E
F	F	F

step 5 :- Transition diagram.



— x — x — x — x —