Name: Sindhuja Yerramalla

UUID: U00839259

mailid : syrrmlla@memphis.edu.

Foundations of Computing HW4

References:- Referred to many websites, Text book, lecturenotes Chegg & worked with few classmates.

Problem 1: Problem 10, sec 9.1 from Linz.

Sol Given $f(w) = w^R$, where $w \in \{0,1\}^*$

let $M = (\{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}, \{0,1\}, \{0,1,\square\}, \delta, q_0, \square, \{q_{11}\})$.

where $\delta$ is defined as:

$\delta(q_0, 0) = (q_1, x, R)$ $\quad$ $\delta(q_1, 0) = (q_1, 0, R)$ $\quad$ $\delta(q_1, \square) = (q_3, \square, L)$

$\delta(q_0, 1) = (q_2, y, R)$ $\quad$ $\delta(q_1, 1) = (q_1, 1, R)$ $\quad$ $\delta(q_2, 0) = (q_2, 0, R)$

$\delta(q_0, x) = (q_0, x, L)$ $\quad$ $\delta(q_1, x) = (q_3, x, L)$ $\quad$ $\delta(q_2, 1) = (q_2, 1, R)$

$\delta(q_0, y) = (q_0, y, L)$ $\quad$ $\delta(q_1, y) = (q_3, y, L)$ $\quad$ $\delta(q_2, x) = (q_4, x, L)$

$\delta(q_2, y) = (q_4, y, L)$ $\quad$ $\delta(q_4, 1) = (q_6, y, L)$ $\quad$ $\delta(q_6, 1) = (q_6, 1, R)$

$\delta(q_2, \square) = (q_4, \square, L)$ $\quad$ $\delta(q_4, y) = (q_9, y, L)$ $\quad$ $\delta(q_6, y) = (q_0, y, R)$

$\delta(q_3, 0) = (q_7, x, L)$ $\quad$ $\delta(q_5, 0) = (q_5, 0, R)$ $\quad$ $\delta(q_7, 0) = (q_8, 0, R)$

$\delta(q_3, 1) = (q_8, x, L)$ $\quad$ $\delta(q_5, 1) = (q_5, 1, R)$ $\quad$ $\delta(q_7, 1) = (q_8, 1, R)$

$\delta(q_3, x) = (q_9, x, L)$ $\quad$ $\delta(q_5, y) = (q_0, x, R)$ $\quad$ $\delta(q_7, x) = (q_0, x, R)$

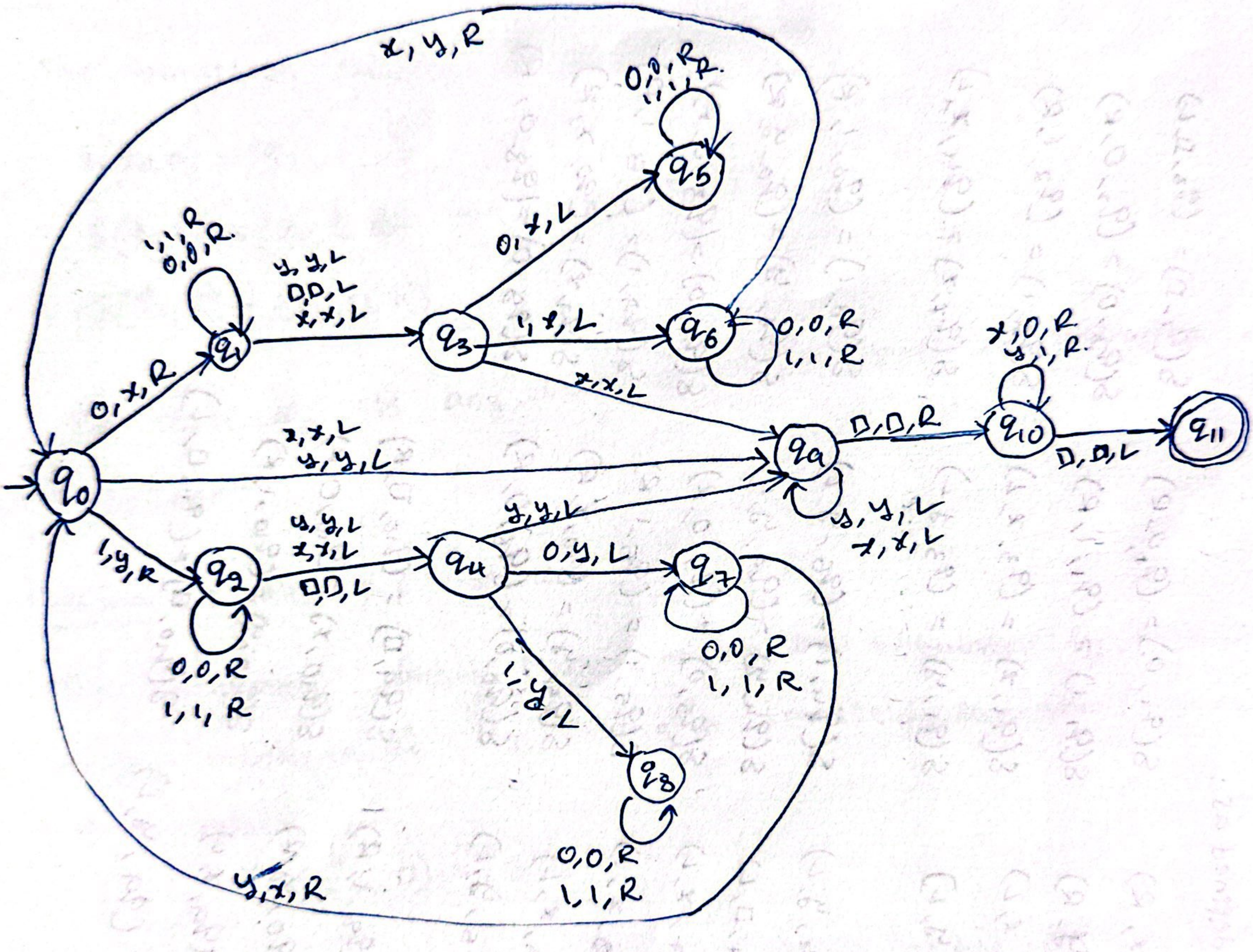
$\delta(q_4, 0) = (q_5, y, L)$ $\quad$ $\delta(q_6, 0) = (q_6, 0, R)$ $\quad$ $\delta(q_8, 0) = (q_8, 0, R)$

$\delta(q_8, 1) = (q_8, 1, R)$ $\quad$ $\delta(q_9, \square) = (q_{10}, \square, R)$

$\delta(q_8, x) = (q_0, y, R)$ $\quad$ $\delta(q_{10}, x) = (q_{10}, 0, R)$

$\delta(q_9, x) = (q_9, x, L)$ $\quad$ $\delta(q_{10}, y) = (q_{10}, 1, R)$

$\delta(q_9, y) = (q_9, y, L)$ $\quad$ $\delta(q_{10}, \square) = (q_{11}, \square, L)$


x, y, R
0,0,R
1,1,R
1,1,R
0,0,R
y,y,L
□,□,L
x,x,L
0,x,L
1,x,L
0,0,R
1,1,R
x,x,L
0,x,R
x,x,L
y,y,L
□,□,R
x,0,R
y,1,R
□,□,L
y,y,L
x,x,L
1,y,R
y,y,L
x,x,L
□,□,L
y,y,L
0,y,L
0,0,R
1,1,R
1,y,L
0,0,R
1,1,R
0,0,R
1,1,R
y,x,R
$q_0$
$q_1$
$q_2$
$q_3$
$q_4$
$q_5$
$q_6$
$q_7$
$q_8$
$q_9$
$q_{10}$
$q_{11}$

Problem 2 : Problem 8 , Sec 9.2 from linz.

Given macroinstruction:

searchleft$(a, q_i, q_j)$

$M = (\{q_0, q_i, q_j\}, \{a\}, \{a, b, \square\}, \delta, q_0, \square, \{q_0\})$ is the turing machine

The transition function $\delta$ is defined as

$\delta(q_0, a) = (q_i, a, L)$

$\delta(q_0, b) = (q_0, b, L)$ for all $b \in \Gamma - \{a\}$

$\delta(q_0, \square) = (q_j, \square, R)$.

The state $q_0$ is any state where the searchleft instruction may be applied.

Problem 3 : Problem 6 , Sec 10.1 from linz.

The non-erasing Turing Machine M can be simulated by Standard turing machine $\hat{M}$ by adding a transition function (restriction condition) given below.

$$\hat{\delta}(\hat{q}_i, \square) = (\hat{q}_j, \square, L \text{ or } R)$$

Thus the machine $\hat{M}$ on seeing a blank symbol leaves it unchanged and converts it into the input symbol only if it is not a blank symbol

∴ No generality is lost by making such restriction on a non erasing Turing Machine.

Problem 4 : Problem 9, Sec 10.1 from linz

A Turing machine that always writes a symbol other than the one it reads is defined by the transition function given below

$\delta(q_i, a) = (q_j, b, L \text{ or } R)$ where a,b are different.

A standard Turing machine can make the following transition,

$\delta(q_i, a) = (q_j, a, L \text{ or } R)$.

That is, it can rewrite the old symbol before moving the read-write head left or right.

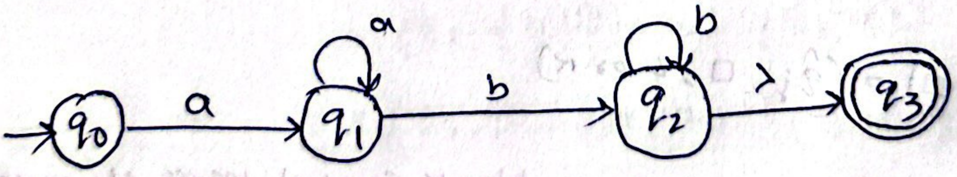
The above discussed Turing machine cannot make such a transition. Therefore such a Turing machine is less powerful when compared to the standard Turing machine.

Problem 5 : Problem 12, Sec 10.1 from linz

Transition function for turing machine for $L(aa^* bb^*)$

$\delta(q_i, \{a, b\}) = (q_j, c, R)$ i.e move if current symbol is neither a nor b

NFA for language.




Turing machine :- $Q = \{q_0, q_1, q_2, q_3, q_f\}$ $\Sigma = \{a, b\}$ $\Gamma \{a, b, \square\}$

$F = \{q_f\}$

$\delta \rightarrow \{(q_0, a) = (q_1, a, R)$

$(q_1, a) = (q_1, a, R)$

$(q_1, b) = (q_2, b, R)$

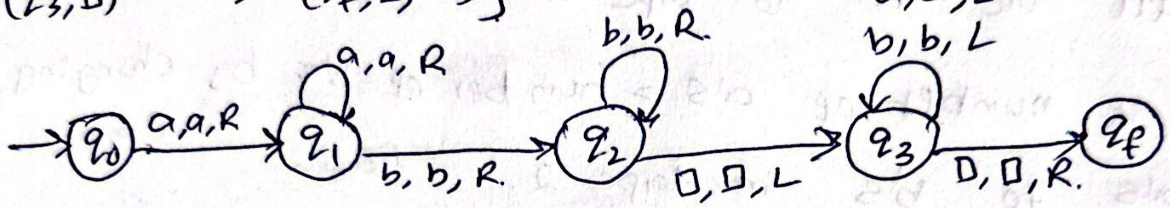
$(q_2, b) = (q_2, b, R)$

$(q_2, \square) = (q_3, \square, L)$

$(q_3, b) = (q_3, b, L)$

$(q_3, a) = (q_3, a, L)$

$(q_3, \square) = (q_f, \square, R)\}$




To implement with the transactions like previous exercise.

$\delta = \{(q_0, \{b, \square\}) = (q_1, a, R)$

$(q_1, \{b, \square\}) = (q_1, a, R)$

$(q_1, \{a, \square\}) = (q_2, b, R)$

$(q_2, \{a, \square\}) = (q_2, b, R)$

$(q_2, \{a, b\}) = (q_3, \square, L)$

$(q_3, \{a, \square\}) = (q_3, b, L)$

$(q_3, \{b, \square\}) = (q_3, a, L)$

$(q_3, \{a, b\}) = (q_f, \square, R)$

Scanned with CamScanner

Problem 6: Problem 1(e), Sec 10.2 from linz.

We can implement $L = \{n_a(w) = n_b(w) = n_c(w)\}$ using two-tape turing machine by implementing tape's as they accept half or stay along with move left & Right.

We have 3 steps in implementing this

1. Copy all the a's to tape 2.
2. check if number of a's = number of b's by changing all the a's to b's in tape 2.
3. check if $n_b(w) = n_c(w)$ like the way it is explained

Program for turing machines:-

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

$\delta = \{(q_0, a, \square) = (q_0, a, a, R, L)$

$(q_0, b, \square) = (q_1, b, \square, S, R)$

$(q_1, b, a) = (q_1, b, a, R, R)$

$(q_1, c, \square) = (q_2, c, \square, S, L)$

$(q_2, c, b) = (q_3, c, b, R, L)$

$(q_3, \square, \square) = (q_f, \square, \square, S, S)\}$