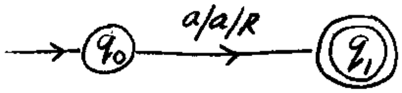
\# 2.



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

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

\# 3. aba :

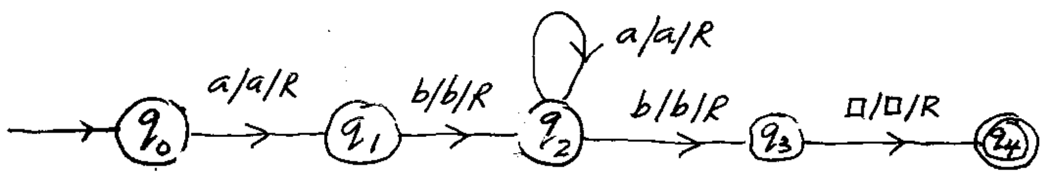
$q_0 aba \vdash x q_1 ba \vdash q_2 xya \vdash x q_0 ya \vdash xy q_3 a$

\# 4 No.

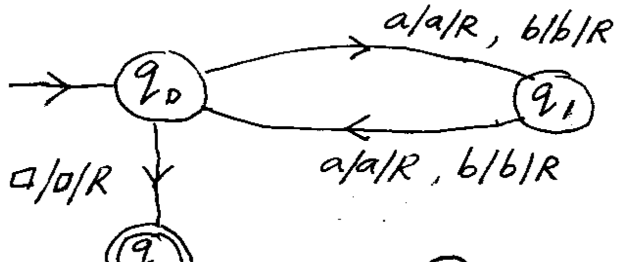
\# 5 $L(\underline{a} b^* + \underline{b}\, \underline{b}^* \underline{a} (\underline{a} + \underline{b})^*)$

\# 6. The Turing Machine halts in a non-accepting state

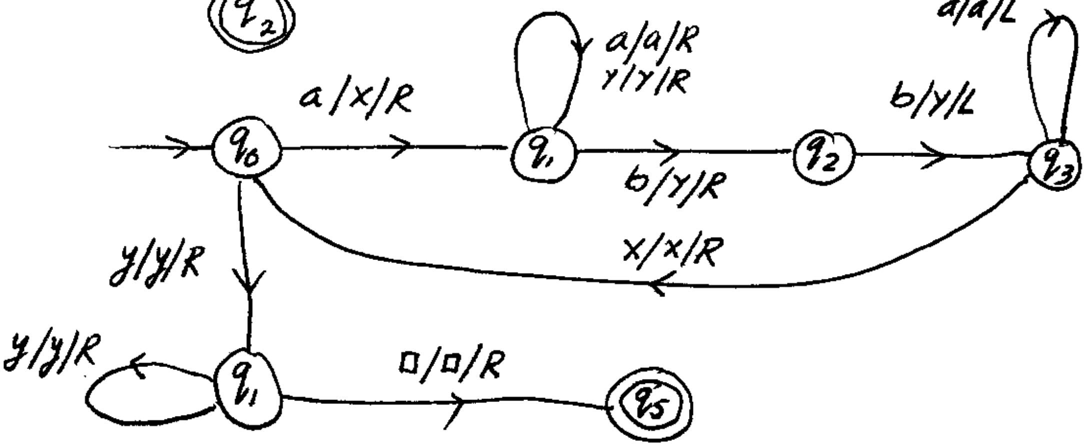
\# 7 (b)



(b)

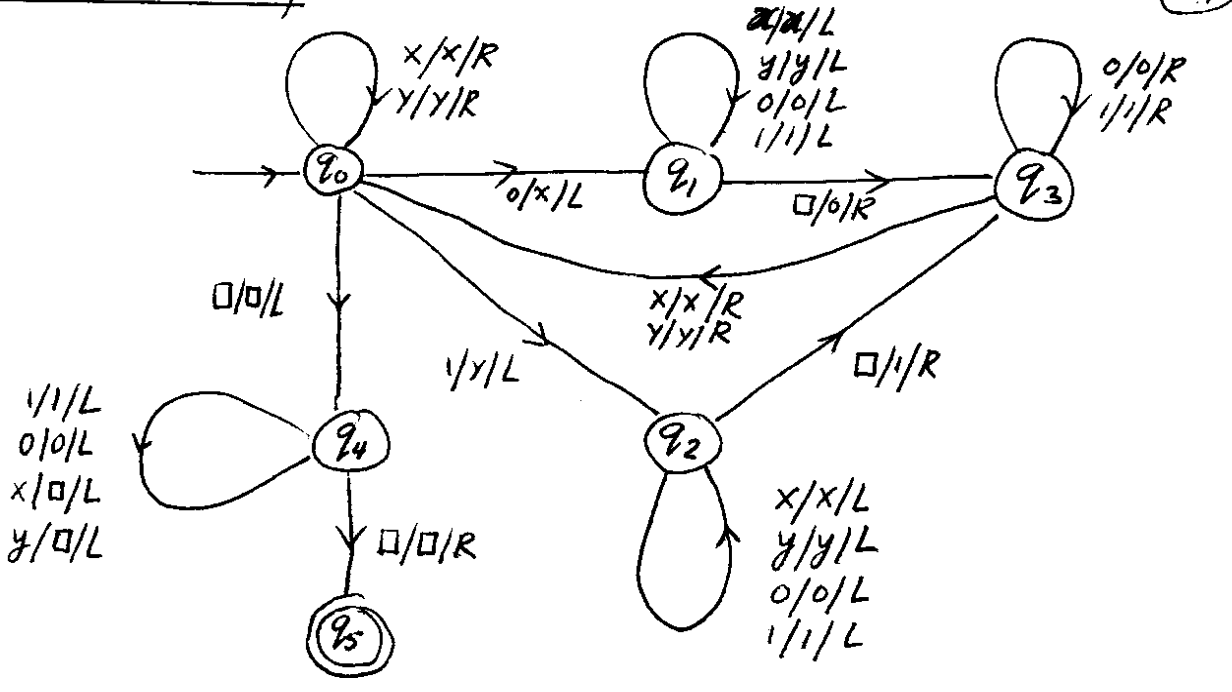


(g)

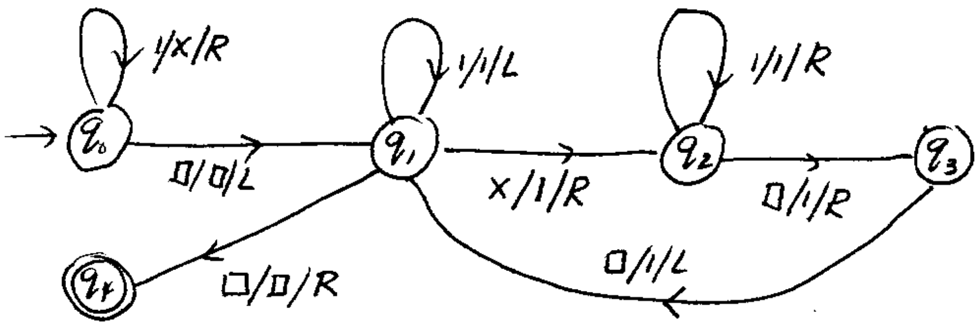


SECTION 9.1 p. 237

#9.



#11(a) Modify the machine in Example 9.10 to write two 1's for each x :



Section 11.1 #12 p. 282.

12 (a) If L_1 is recursive and L_2 is r.e., then $L_2 - L_1$ is necessarily r.e. because $\bar{L}$ is r.e., So $L_2 - L_1 = L_2 \cap \bar{L}_1$ = intersection of two r.e. sets = r.e

(b) If L_1 is recursive and L_2 is r.e., then $L_1 - L_2$ is not necessarily r.e. because if we take $L_1 = \Sigma^*$ and L_2 = a non-recursive r.e. subset of Σ^*, then $L_1 - L_2$ = a non- r.e. subset of Σ^*.

#2 The set of all r.e. languages is countable since every r.e. language is associated with a TM and the number of different TM's is countable. If there were countably many non-r.e. languages as well, then there would be countably many languages altogether. But the set of all languages on Σ is uncountable because $\mathcal{L}(\Sigma) = \mathcal{P}(\Sigma^*) \approx \mathcal{P}(\mathbb{N})$ which is uncountable. So there are uncountably many non-r.e. languages

#5. Suppose $\bar{L}$ is recursive. Then $\bar{\bar{L}}$ will also be recursive by the closure theorem for recursive languages. But $\bar{\bar{L}} = L$. So L is recursive. Hence L is r.e. But we were told that L was not r.e. So if L is not r.e., then $\bar{L}$ cannot be recursive.

#6 Suppose L_1 and L_2 are r.e. Let M_1 and M_2 be TM's such that $L(M_1) = L_1$ and $L(M_2) = L_2$. A third TM M can run both M_1 and M_2 together and enter and enter a final state if either one does (e.g., M can put M_1 & M_2 through their respective moves alternately). Any w which causes either M_1 or M_2 to enter a final state will cause M to enter a final state and vice versa $\therefore$ $L(M_1) \cup L(M_2) = L(M)$

$\therefore$ $L_1 \cup L_2$ will be r.e.

#7 Yes. Let M_1 and M_2 be as in problem 5 above. Another TM M can be designed to run both together and enter a final state if both M_1 and M_2 do. (For example, M can simulate M_1 and M_2 as follows. Put M_1 & M_2 alternately through their moves and if one them halts in a final state, then continue only with the other one until it halts in a final state.) Then $L(M) = L(M_1) \cap L(M_2)$. So $L_1 \cap L_2$ is also r.e.

#8 If L_1 & L_2 are recursive, then $\bar{L}_1$, $\bar{L}_2$, L_1 and L_2 will all be r.e. So by problems 5 and 6,

$L_1 \cup L_2$ will be r.e

and $\bar{L}_1 \cap \bar{L}_2 = \overline{L_1 \cup L_2}$ will be r.e.

Since $L_1 \cup L_2$ & $\overline{L_1 \cup L_2}$ are both r.e., it follows that $L_1 \cup L_2$ is recursive.

Similarly $L_1 \cap L_2$ & $\overline{L_1 \cap L_2} = \bar{L}_1 \cup \bar{L}_2$ will be both r.e. Hence $L_1 \cap L_2$ will be recursive.

Note: recursive sets are closed under $\cup$, $\cap$ and compliments. r.e. sets are not closed under compliments.

SECTION 13.1 p. 332

#1. (a)

$$\begin{aligned}
ADD(3,4) &= ADD(3,3+1) = ADD(3,3)+1 \\
&= ADD(3,2+1)+1 = (ADD(3,2)+1)+1 \\
&= (ADD(3,1+1)+1)+1 = ((ADD(3,1)+1)+1)+1 \\
&= ((ADD(3,0+1)+1)+1)+1 \\
&= (((ADD(3,0)+1)+1)+1)+1 \\
&= (((3+1)+1)+1)+1 = ((4+1)+1)+1 \\
&= (5+1)+1 = 6+1 = 7
\end{aligned}$$

(b)

$$\begin{aligned}
MULT(2,3) &= MULT(2,2+1) = MULT(2,2)+2 \\
&= MULT(2,1+1)+2 = (MULT(2,1)+2)+2 \\
&= (MULT(2,0+1)+2)+2 = ((MULT(2,0)+2)+2)+2 \\
&= ((0+2)+2)+2 = \cdots = 6
\end{aligned}$$

#3 Note $1 \dot{-} ((x \dot{-} y)+(y \dot{-} x)) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$

$\therefore$ equal (x,y)

$= \text{monus}(c_1(p_1(x,y)), \text{add}(\text{monus}(x,y), \text{monus}(p_2(x,y), p_1(x,y))))$

#4 (a) Note that $f(x,y) = x \dot{-} h(x,y)$ where

$$h(x,y) = \begin{cases} 0 & \text{if } x \neq y \\ x & \text{if } x = y \end{cases}$$

So $f(x,y) = x \dot{-} (x \cdot \text{equal}(x,y))$

$= \text{monus}(p_1(x,y), \text{mult}(p_1(x,y), \text{equal}(x,y)))$

(b) $f(0) = 1$

$f(y+1) = \text{mult}(y, f(y))$

From this we can see $f(y) = y!$ is primitive recursive by doing a little work.