1. This exercise concerns TM $M_1$ whose description and state diagram appear in the scanned example (from Sipser first edition), which is design to recognize the language $L = \{y \in \{0,1,\#\}^* \mid \exists w \in \{0,1\}^* \ y = w\#w\}$. In each of the parts, give the sequence of configurations that $M_1$ enters when started on indicated input string.

   1. 00.
   2. 0\#0.
   3. 0\#0.
   4. 00\#01.
   5. 01\#01.

2. Give state diagrams (pictures) for a Turing Machine that decides the following language over the alphabet $\{0,1\}$: $\{w \mid w$ contains twice as many 1s as 0s\}.

3. If $M_1$ and $M_2$ are two (not necessarily halting) Turing machines, then there exists a Turing machine $M$ such that $L(M) = L(M_1) \cup L(M_2)$. Prove the set equality, and be aware of infinite loops. Feel free to use multitape but do not use nondeterministic Turing Machines.

4. In Theorem 13 from the notes (various numbers in various Sipser editions) we showed that a language is Turing-recognizable if and only if some enumerator enumerates it. Why didn’t we use the following simpler algorithm for the ”only if” part of the proof? As in the proof, $s_1,s_2,\ldots$ is a list of all strings in $\Sigma^*$.

   A = “Ignore the input.
   1. Write down $w = s_1$
   2. Repeat the following for $i = 1, 2, 3\ldots$
   3. Run $M$ on $w$.
   4. If $M$ accepts, print out $w$.
   5. Obtain new $w$ from old $w$ by the TM that constructs $s_{i+1}$ from $s_i$”

5. A Turing machine with stay put instead of left is just like the normal TM except the transition function, which is:

   $$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, S\}.$$  

   The machine can only move its head right, or let it stay in the same position. Show that this type of Turing machine is not equivalent to the usual version; that is exhibit a language one type can recognize (give a state diagram here) and the other cannot (argue that the other type cannot recognize that language).