Problem 1. We saw in class that a language is Turing-recognizable if and only if some enumerator enumerates it. Why wasn’t the following simpler algorithm used for the forward direction of the proof? Recall, $s_1, s_2, \ldots$ is a list of all strings in $\Sigma^*$.

$$ E = \text{“Ignore the input:}$$

1. Repeat the following for $i = 1, 2, 3, \ldots$:
2. Run $M$ on $s_i$.
3. If $M$ accepts, print out $s_i$."

Problem 2. A Turing machine with stay put instead of left is similar to an ordinary Turing machine, but the transition function has the form $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{R, S\}$. At each point, the machine can move its head right or let it stay in the same position. Does such a Turing machine variant recognize the same class of languages as an ordinary Turing machine? Explain your answer.

Problem 3. Let $ALL_{\text{DFA}} = \{ \langle A \rangle \mid A$ is a DFA and $L(A) = \Sigma^* \}$. Show that $A$ is decidable.

Problem 4. Let $A = \{ \langle R, S \rangle \mid R$ and $S$ are regular expressions and $L(R) \subseteq L(S) \}$. Show that $A$ is decidable.

Problem 5. Let $A_{\epsilon_{\text{CFG}}} = \{ \langle G \rangle \mid G$ is a CFG that generates $\epsilon \}$. Show that $A_{\epsilon_{\text{CFG}}}$ is decidable.

Problem 6. Consider the problem of determining whether a DFA and a regular expression are equivalent. Express this problem as a language and show that it is decidable.

Problem 7. Show that the collection of decidable languages is closed under the operations of complement and intersection.