1. (Sipser 4.10) Let

\[ A = \{ \langle M \rangle | M \text{ is a DFA which doesn't accept any string containing an odd number of 1s} \} \]

Show that \( A \) is decidable.

2. (Sipser 4.18) Let \( A \) and \( B \) be two disjoint languages. Say that language \( C \) separates \( A \) and \( B \) if \( A \subseteq C \) and \( B \subseteq \overline{C} \). Show that any two disjoint co-Turing-recognizable languages are separable by some decidable language.

3. (Sipser 4.21) Let \( A \) be a Turing-recognizable language consisting of descriptions of Turing machines, \( \{ \langle M_1 \rangle, \langle M_2 \rangle, \ldots \} \), where every \( M_i \) is a decider. Prove that some decidable language is not decided by any decider \( M_i \) whose description appears in \( A \). (Hint: You may find it helpful to consider an enumerator for \( A \)).

4. (Sipser 4.13) Show that the problem of testing whether a CFG generates all strings in \( 1^* \) is decidable. In other words, show that

\[ \{ \langle G \rangle | G \text{ is a CFG over } \{0,1\}^* \text{ and } 1^* \subseteq L(G) \} \]

is a decidable language.