Problem 1

(a) Show that the class of Turing-decidable languages is closed under complement. In particular, given a Turing machine \( M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}}) \) that is guaranteed to halt for any input (we refer to these Turing machines as TM-deciders), give a 7-tuple definition of a TM-decider \( M' \) accepting the complement of \( L(M) \).

(b) Briefly explain why the same construction cannot be used to show that Turing-recognizable languages are closed under complement. (In fact, this class of languages is not closed under the complement!)

Problem 2

(a) The class of Turing-decidable languages is closed under union. The construction could go as follows. Given TM-deciders \( M_1 \) and \( M_2 \), we construct a TM-decider \( M \) such that \( L(M) = L(M_1) \cup L(M_2) \). For simplicity, we will first construct a 2-tape TM, and then covert it to a single tape TM as described in class/book.

1. Copy the input \( w \) to the second tape.
2. Run \( M_1 \) on the first tape (using input \( w \)).
3. If \( M_1 \) accepts, then \( M \) accepts.
4. If \( M_1 \) rejects, then run \( M_2 \) on the second tape (using the copy of the input \( w \)).
5. If \( M_2 \) accepts, then \( M \) accepts.
6. If \( M_2 \) rejects, then \( M \) rejects.

Can the same construction be used to show that the class of Turing-recognizable languages is closed under union? Briefly reason your answer.

(b) The class of Turing-recognizable languages is closed under union. For TM \( M_1 \) and \( M_2 \), describe a construction of a TM \( M \) such that \( L(M) = L(M_1) \cup L(M_2) \).

(c) The class of Turing-decidable languages is closed under intersection. For TM-deciders \( M_1 \) and \( M_2 \), describe a construction of a TM-decider \( M \) such that \( L(M) = L(M_1) \cap L(M_2) \).

(d) Is the class of Turing-recognizable languages closed under intersection? Briefly reason your answer.
Problem 3

A *rewind TM* is a TM that is allowed to move its head to the right (R), stay in place (S), or rewind to the beginning (B).

(a) Can every standard TM be simulated by a rewind TM? Reason your answer. If you answered yes, describe the corresponding construction.

(b) Can every rewind TM be simulated by a standard TM? Reason your answer. If you answered yes, describe the corresponding construction.

Problem 4

Give a pseudo code of an algorithm that decides whether a given context-free grammar generates the empty language. Your algorithm takes a CFG $G$ as its input and it outputs YES if $L(G) = ∅$ and NO otherwise.

**Note:** If we represent a CFG $G$ by a string $⟨G⟩$ (e.g., listing all the production rules separated by special delimiter symbols), then the pseudo code implies that there is a TM-decider for the language $\{⟨G⟩ \mid G$ is a CFG and $L(G) = ∅\}$. 