Guidelines:

Please submit your assignment to the TA/lecturer at the beginning of the appropriate lecture. Solutions should be typed, drawings can be done by hand.

You are encouraged to work in groups of up to three people, however we ask that you dedicate enough time to think about each problem by yourself before consulting others. Moreover, you must write up each problem solution by yourself without assistance. You are asked to identify your collaborators on problem sets. If you did not work with anyone, please write “Collaborators: none”.

You may consult the course notes and the text (Sipser). When this problem set references a section or a problem, we mean the corresponding section or problem of the textbook Introduction to the Theory of Computation (3rd Edition) by Michael Sipser.

The assignment questions may be similar to questions on problem sets from past offerings of this course or courses at other universities. Using any preexisting solutions from these or any other sources is strictly prohibited.

Each of the Questions 1-6 is worth 10 points, please attempt all problems. Please provide full proofs, unless stated otherwise.

In all that follows, you may assume that all strings encode some Turing machine and some automaton, thus any string is an input of the correct format to all the languages described below.

Question 1.

1. (6 points) Consider the following definition:

   **Definition:** A language $L \subseteq \Sigma^*$ is called RE-complete if the following holds:
   
   (a) (membership): $L \in RE$.
   
   (b) (hardness): For every $L' \in RE$, it holds that $L' \leq_m L$.

   Prove that $A_{TM}$ is RE-complete.

2. (4 points) Let

   $A_{DFA} = \{ \langle B, w \rangle : B \text{ is a DFA that accepts } w \}$. 

   3-1
Recall that we have shown in class that $A_{DFA}$ is decidable.
Consider the following (false) claim and proof. Find the bug in the proof.

**Claim:** $A_{DFA}$ is undecidable.

**Proof:** We show $A_{TM} \preceq_m A_{DFA}$. Given an input $\langle M, w \rangle$ for $A_{TM}$ we construct an input $f(\langle M, w \rangle) = \langle B, w' \rangle$ for $A_{DFA}$ as follows: it holds that either $M$ accepts $w$ or $M$ does not accept $w$. In the first case, we output $\langle B, w' \rangle$ where $B$ is a DFA such that $L(B) = \Sigma^*$ (e.g an accepting self-loop) and $w' = \varepsilon$, and in the second case we output $\langle B, w' \rangle$ such that $L(B) = \phi$ (e.g a rejecting self-loop) and $w' = \varepsilon$. It is easy to see that $M$ accepts $w$ iff $B$ accepts $w'$.

Clearly constructing these automata $B$ is computable. Also, we only need to show that there exists a reduction, and not compute the reduction itself, and since we know that one of the two cases above always holds, then one of the outputs we defined is correct, so the reduction exists. □

**Question 2.**

1. (4 points) Use a diagonalization argument to show that $HALT_{TM}$ is undecidable.
2. (3 points) Show that $\overline{HALT}_{TM} \preceq_m HALT_{TM}$.
3. (3 points) Read Section 6.3 in the textbook about Turing reducibility (pages 260-261). Show that $\overline{HALT}_{TM} \preceq_T HALT_{TM}$.

**Remark:** We have seen that if $A \preceq_m B$, and $B$ is undecidable, than $A$ is undecidable. Theorem 6.21 shows that the same holds for Turing reductions. For mapping reductions it also holds that if $A \preceq_m B$, and $B$ is unrecognizable, than $A$ is unrecognizable. This part of the question shows that a similar claim in not true for Turing reductions, as $\overline{HALT}_{TM}$ is unrecognizable, but $HALT_{TM}$ is recognizable.

**Question 3.**

A *useful state* in a deterministic Turing machine is one that is entered during the machine’s computation on at least one input string. Consider the language

$$USEFUL_{TM} = \{ \langle M, q \rangle : M \text{ is a TM and } q \text{ is a useful state of } M \}.$$  

1. (3 points) Prove that $USEFUL_{TM}$ is recognizable.
2. (5 points) Prove that $USEFUL_{TM}$ is undecidable.
3. (2 point) Is $USEFUL_{TM}$ recognizable?
**Question 4.**

Let $ALL_{TM}$ be the set of Turing machines that accept every input string. That is,

$$ALL_{TM} = \{ \langle M \rangle : M \text{ is a TM with input alphabet } \Sigma \text{ and } L(M) = \Sigma^* \}.$$ 

1. (5 points) Prove that $\overline{ALL}_{TM}$ is unrecognizable.

2. (5 points) Prove that $ALL_{TM}$ is unrecognizable.

*Hint:* Prove $\overline{A}_{TM} \preceq_m ALL_{TM}$. Note that if $\langle M, x \rangle \in \overline{A}_{TM}$, then $M$ halts and accepts $w$ after $n$ steps, for some $n \in \mathbb{N}$. Let $\langle N \rangle$ be the input for $ALL_{TM}$ constructed by the reduction. What should $N$ do given $x \in \Sigma^n$?

**Question 5.**

Consider the problem $WW_{PDA}$ of determining if a given pushdown automaton (Chapter 2.2 in the textbook) accepts some string of the form $ww$. Formally,

$$WW_{PDA} = \{ \langle P \rangle : P \text{ is a PDA and } L(P) \cap \{ ww : w \in \{0,1\}^* \} \neq \emptyset \}.$$ 

Use the computation history method (pages 220-226 in the textbook) to show that $WW_{PDA}$ is undecidable.

*Hint:* The input to the constructed PDA should be of the form $CH_M,x$ where $CH_M$ is the computational history of $M$ on $x$. Recall from Problem Set 1, Question 6, that there is a PDA that computes the language $L = \{ w \in \Sigma^* : w = w^R \}$. Therefore, some of the configurations in $CH_M$ should be written in reversed order, to allow the PDA to compare configurations.

**Question 6.**

Let $\prec$ be the lexicographical order on strings in $\{0,1\}^*$. In other words, $s_1 \prec s_2$ iff either $|s_1| < |s_2|$, or $|s_1| = |s_2|$ and $\text{bin}(s_1) < \text{bin}(s_2)$, where $\text{bin}(s)$ is the binary number represented by the string $s \in \{0,1\}^*$.

Furthermore, a sequence of strings $s_1, s_2, s_3, \ldots$ is monotone increasing iff for all $k$, $s_k \prec s_{k+1}$. For example: $1, 00, 01, 000$ is a monotone increasing sequence as $1 \prec 00$, $00 \prec 01$, and $01 \prec 000$.

1. (2 points) Prove that every infinite sequence of strings contains a subsequence which is infinite and monotone increasing.

2. (4 points) Use this idea to prove that for every infinite recognizable language $L$ in $\{0,1\}^*$, there is an infinite decidable language $L' \subseteq L$. 
Hint: The heading “Enumerators” in Chapter 3 of the textbook (pages 180-181) should be useful here.

Remark: This would, for example, imply that a non-looping Turing machine exists that can accept infinitely many (though not all) instances of the Halting Problem, while rejecting all other strings.

3. (4 points) Prove that for every infinite decidable language \( L \) in \( \{0,1\}^* \), there is an infinite undecidable language \( L' \subseteq L \).

Hint: Filter the elements of \( L \) using an undecidable language.