Solutions to Practice Midterm 1

1. State whether each of the following statements is true. In addition, give a short proof (2-3 lines are sufficient) if the statement is true, and give a counterexample otherwise.

   (a) If $L_1, L_2, \ldots, L_{172}$ are all regular languages, then the language $\bigcap_{i=1}^{172} L_i$ is regular.

   (b) If $L_1, L_2, L_3, \ldots$ is an infinite sequence of regular languages, then the language $\bigcap_{i=1}^{\infty} L_i$ is regular.

   Solution Outline:

   (a) True. We know that the intersection of any 2 regular languages is regular. It follows by induction that the intersection of any finite collection of regular languages is regular.

   (b) False. Let $w_1, w_2, \ldots$ be the strings in the complement of some irregular language L over $\{0,1\}$, and let $L_i = \{0,1\}^* \backslash \{w_i\}$. By de Morgan's law, $\bigcap_{i=1}^{\infty} L_i = L$, which is not regular.
   Alternatively, we could take $L_i = \{0^k 1^k \mid 1 \leq k \leq i\} \cup \{0^{k+1}\Sigma^*\}$ where $\Sigma = \{0,1\}$. Then, $\bigcap_{i=1}^{\infty} L_i = \{0^n 1^n \mid n \geq 1\}$ is not regular.

2. Let

   $$L = \{(\langle D \rangle, w) \mid D \text{ is a DFA over the binary alphabet } \{0,1\} \text{ that accepts } w\}$$

   (Assume that the encoding of DFAs also uses the binary alphabet.)

   (a) Show that L is not regular.

   (b) Show that L is decidable.

   Solution Outline:

   (a) Method I: Let $D_i, i \geq 1$ be the DFA that recognizes the language $\{1^i\}$. Then, $\{(\langle D_i \rangle, \epsilon)\}_{i \geq 1}$ constitutes an infinite collection of distinguishable strings.

   Method II: Suppose on the contrary that L is regular. Then, let M be a DFA that recognizes L and k be the number of states in M. Let N be a DFA for some language $L(N)$ that requires a DFA with at least $k+1$ states (such a DFA exists because there are infinitely many distinct regular languages). Let q be the state of M that M ends up in upon reading input $(\langle N \rangle, \epsilon)$. Modify M to obtain a DFA M' whose start state is q. Then, it is easy to check that M' is a DFA for $L(N)$ with k states, a contradiction.

   Method III: Assume that the encoding of a DFA D starts with a string of k 1's, where k is the number of states in D, followed by a 0, and then some prefix-free encoding of binary representation of k, followed by two 0's, followed by some appropriate encoding of D. Now, assume on the contrary that L is decidable, and let p be the pumping length. Let N be a DFA for some language $L(N)$ that requires a DFA with at least $p+1$ states and w be some string in N. Then, $(\langle N \rangle, w) \in L$. If we applying the pumping lemma $(\langle N \rangle, w)$ and either pump up or pump down, we obtain an input that does not have a valid encoding of a DFA, a contradiction.

(b) We can construct a decider for L as follows. First, reject if the input is not correctly encoded; otherwise, parse the input as $(\langle D \rangle, w)$ where D is a DFA and $w \in \{0,1\}^*$. Then, simulate D on input w, and accept if D accepts w, and reject otherwise.

3. Consider the language

$$INT_{\mathsf{TM}} = \{\langle M_1, M_2 \rangle : L(M_1) \cap L(M_2) \neq \emptyset\}.$$

(Thus, INT_{TM} is the language associated with the problem of deciding whether, for two given Turing machines M_1 and M_2, there is some string that is accepted by both machines.)

(a) Show that INT_{TM} is Turing recognizable.

(b) Show that INT_{TM} is not decidable.

Solution Outline:

(a) We construct a Turing machine that recognizes INT_{TM} as in the construction of an enumerator for a Turing-recognizable language. On input $\langle M_1, M_2 \rangle$, for $i = 1, 2, \ldots$, for each string s of length at most i, simulate each M_1 and M_2 on input s for i steps, and accept if both M_1 and M_2 accept s.

(b) We shall present a mapping reduction from A_{TM} to INT_{TM}, and since A_{TM} is undecidable, it would follow that INT_{TM} is undecidable. The reduction is as follows: on input $\langle M, w \rangle$, first construct a machine M_w that on input x, check if $x = w$. If so, it simulates M on x and otherwise, reject. In addition, construct a machine M_{all} that accepts all inputs. Output $\langle M_w, M_{all} \rangle$. It is easy to see that M accepts w iff $\langle M_w, M_{all} \rangle \in INT_{\mathsf{TM}}$.

4. Let $S = \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \{\langle M \rangle\}\}$. Show that neither S nor $\overline{S}$ is Turing-recognizable.

Solution Outline: For this problem, we assume that a TM can recognize its own code[1]. We then show that $A_{TM} \leq_m S$ and $A_{TM} \leq_m \overline{S}$, which also imply $\overline{A_{TM}} \leq_m \overline{S}$ and $\overline{A_{TM}} \leq_m S$ respectively.

We first give the reduction from A_{TM} to S. Given an instance $\langle M, w \rangle$ of A_{TM}, we construct a machine M' which given an input x, rejects if $x \neq \langle M' \rangle$ and simulates M on w if $x = \langle M' \rangle$. Thus, $L(M') = \{\langle M' \rangle\}$ if M accepts w and $\emptyset$ otherwise. Similarly, for the reduction from A_{TM} to $\overline{S}$, we make M' accept if $x = \langle M' \rangle$ and simulate M on x otherwise. In this case, it gives $L(M') = \Sigma^*$ if M accepts w and $\{\langle M' \rangle\}$ otherwise.

[1] This assumption will be justified later in the class - apologies for using this here.