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 \( L_1, L_2, L_3, \ldots \) are 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 \( L_i = \{0, 1\}^* \setminus \{w_i\} \). By de Morgan’s law, \( \bigcap_{i=1}^{\infty} L_i = L \), which is not regular.

Alternatively, we could take \( L_i = \{0^k1^k \mid 1 \leq k \leq i\} \cup \{0^{k+1}\Sigma^*\} \) where \( \Sigma = \{0, 1\} \).

Then, \( \bigcap_{i=1}^{\infty} L_i = \{0^*1^n \mid n \geq 1\} \) is not regular.

**Remark:** Some students suggested taking \( L_i \) to be the language described by the regular expression \( 0^* \Sigma^* \), where \( \Sigma = \{0, 1\} \). This is not a valid counter-example, because \( \bigcap_{i=1}^{\infty} L_i = \emptyset \), which is regular. In particular, if we take any \( w \in \{0, 1\}^*, w \notin L_{|w|+1} \).

2. Fix \( n_1, n_2 \) to be positive integers. Show that there exists a constant \( N = f(n_1, n_2) \) that only depends on \( n_1 \) and \( n_2 \) with the following property: given any two DFAs \( M_1 \) and \( M_2 \) over \( \Sigma \) having \( n_1 \) and \( n_2 \) states respectively and such that \( L(M_1) \neq L(M_2) \), there is some string \( w \) in \( L(M_1) \triangle L(M_2) \) of length at most \( N \).

**Solution Outline:**

We prove the statement for \( N = f(n_1, n_2) = n_1n_2 \). Consider the DFA \( D \) with \( N \) states that run \( M_1 \) and \( M_2 \) in parallel, and accept if exactly one of \( M_1 \) and \( M_2 \) accepts the input.\(^1\) Now, if \( L(M_1) \neq L(M_2) \), then \( L(D) \neq \emptyset \). By applying the pumping lemma, \( L(D) \) must accept some \( w \) string of length at most \( N \). Then, \( w \in L(M_1) \triangle L(M_2) \).

It is also possible to prove the statement with \( f(n_1, n_2) = n_1 + n_2 \). (How?)

3. 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.

---

\(^1\)In particular, we can take \( Q_D = Q_1 \times Q_2, \delta_D((q_1, q_2), \sigma) = (\delta(q_1, \sigma), \delta(q_2, \sigma)) \) and \( F_D = F_1 \times F_2 \cup F_1 \times F_2 \).
Solution Outline:

(a) **Method I:** Let $D_i, i \geq 1$ be the DFA that recognizes the language $\{1^i\}$. Then, $\{(\langle D_i \rangle, \varepsilon)\}_{i \geq 1}$ 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 $\langle N \rangle, \varepsilon$. 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 $k_1$ 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, $\langle N, w \rangle \in L$. If we applying the pumping lemma $\langle N, w \rangle$ 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 $\langle D, w \rangle$ 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.

4. Show that any infinite, Turing-recognizable (recursively enumerable) language contains an infinite, decidable language.

**Solution Outline:**

Consider any infinite, Turing-recognizable language $L$ and its enumerator $E$. Consider the following enumerator $E'$:

i. Run $E$ and output the first string $w$ that $E$ outputs. Set $\ell = |w|$.

ii. Whenever $E$ outputs a string $w'$ of length longer than $\ell$, output $w'$ and set $\ell = |w'|$. Such a $w'$ always exists because $L$ is infinite.

iii. Return to ii.

$E'$ outputs an infinite subset of $L$ in lexicographical order, which constitutes an infinite, decidable subset of $L$. 

2