CS 373 Fall 2010
Quiz 4 Solutions

Lecture 1 - Mahesh

1. C. For any string \( w \), either \( w \) is in \( A \), \( w \) is in \( B \), or \( w \) is in neither (and it cannot be in both). The machines that decide \( A \) and \( B \) simulate the recognizers for \( A \), \( B \), and \( A \cup B \) in parallel on any input \( w \). One of them will accept in finite time, which identifies where \( w \) is.

2. C. Say \( L' = A_{TM} \). (a) is incorrect if \( L = \Sigma^* \times \Sigma^* \), and (b) is incorrect if \( L = \emptyset \). We can reduce \( L \) as follows: on input \( w \), we actually compute if \( w \in L \) (which can be done in finite time since \( L \) is decidable). If yes, \( f(w) = 0011 \). Else \( f(w) = 011 \). Thus \( w \in L \) iff \( f(w) \in L_{01n} \).

3. C. The reduction tells us that there is a function \( f \) such that \( w \in A \) iff \( f(w) \in B \). Thus \( w \in \tilde{A} \) iff \( w \notin A \) iff \( f(w) \notin B \) iff \( w \in \tilde{B} \) for the same function \( f \).

4. C. Since \( L_d \) is not RE, we learn this about \( L \).

5. C. (c) is a feature of a machine, not a language. Notice that (b) is actually a feature of a language, since a TM with an odd number of states can be in that set.

6. B. We can build a recognizer for \( L \) as follows: on input \( M \), dovetail all possible strings until 312929 are accepted. There is no decider though, since we cannot tell which strings are not in \( L(M) \) without running \( M \) on all of them.

Lecture 2 - Gul

1. C. For any string \( w \), either \( w \) is in \( A \), \( w \) is in \( B \), or \( w \) is in both (and it cannot be in neither). The machines that decide \( A \) and \( B \) work as follows: on input \( w \), put \( w \) in the \((A \cap B) \cup (A \cap B)\) decider. If it rejects, \( w \) is in \( A \cap B \) so accept \( w \). If it accepts, the \( A \) and \( B \) recognizers can be run in parallel to find out which set \( w \) is in.

2. C. Say \( L' = L_{d} \). (a) is incorrect if \( L = \Sigma^* \), and (b) is incorrect if \( L = \emptyset \). We can reduce \( L \) as follows: on input \( w \), pass \( f(w) = \langle M_L, w \rangle \) to a machine for \( A_{TM} \), where \( M_L \) is a machine for \( L \). Thus \( w \in L \) iff \( \langle M_L, w \rangle \in A_{TM} \).

3. C. The reduction tells us that there is a function \( f \) such that \( w \in A \) iff \( f(w) \in B \). Thus \( w \in \tilde{A} \) iff \( w \notin A \) iff \( f(w) \notin B \) iff \( w \in \tilde{B} \) for the same function \( f \).

4. B. Since \( A_{TM} \) is not decidable, we learn that \( L \) is also not decidable. We do not know whether \( L \) is recognizable or not.

5. A. This is a feature of \( M \)'s language, where the other two are features of the machine.
6. $\bar{L}$ is RE, so option (b) is impossible. Intuitively, we can see there is no machine for $L$. On input $M$, such a machine would have to check all strings to see that only 312929 are accepted by $M$. Thus $L$ is not RE, though its complement is.