CS 341, Fall 2007
Solutions for Midterm 2

1. (a) True, see slide 4-54.
   (b) False, see slide 4-30.
   (c) False. A TM M could also loop on w.
   (d) False. The language $A = \{0^n 1^n 2^n \mid n \geq 0\}$ is decidable but not context-free.
   (e) True. Homework 8, problem 2.
   (f) False. For example, if A recognizes Σ^* and B recognizes $\emptyset$, then $\overline{L(A)} \cap L(B) = \emptyset$, but A and B are not equivalent. DFAs A and B are equivalent if and only if $[L(A) \cap \overline{L(B)}] \cup [\overline{L(A)} \cap L(B)] = \emptyset$.
   (g) False, since $\overline{A_{\text{TM}}}$ is not Turing-recognizable by Corollary 4.23.
   (h) True, by Theorem 4.4.
   (i) True, see slide 4-38.
   (j) False, by Theorems 3.16 and 3.13.

2. (a) A language L_1 that is Turing-recognizable has a Turing machine M_1 that may loop forever on a string $w \notin L_1$. A language L_2 that is Turing-decidable has a Turing machine M_2 that always halts.
   (b) The informal notion of an algorithm corresponds exactly to a Turing machine that always halts.
   (c) If $x, y \in A$ with $x \neq y$, then $f(x) \neq f(y)$. Equivalently, if $f(x) = f(y)$, then $x = y$.
   (d) For all $y \in B$, there exists $x \in A$ such that $f(x) = y$.

3. This is basically Homework 7, problem 1.

   (a) $q_1 00 \quad \sqcup q_2 0 \quad \sqcup x q_3 \sqcup \quad \sqcup q_5 x \quad q_5 \sqcup x \quad \sqcup q_2 x \quad \sqcup x q_2 \sqcup \quad \sqcup x \sqcup q_{\text{accept}}$

   (b) $q_1 000 \quad \sqcup q_2 00 \quad \sqcup x q_3 0 \quad \sqcup x 0 q_4 \sqcup \quad \sqcup x 0 \sqcup q_{\text{reject}}$

4. Theorem 4.17.

5. This problem is a slight variation of Homework 8, problem 1. The equivalence problem for regular expressions can be expressed as the language

$$EQ_{\text{REX}} = \{ \langle R_1, R_2 \rangle \mid R_1 \text{ and } R_2 \text{ are regular expressions with } L(R_1) = L(R_2) \}.$$

The language EQ_{DFA} is decidable by Theorem 4.5, and let M be a TM that decides EQ_{DFA}. The following TM S decides EQ_{REX}:

S = "On input $\langle R_1, R_2 \rangle$, where R_1, R_2 are regular expressions:

1. Check if $\langle R_1, R_2 \rangle$ is a proper encoding. If not, *reject*.

2. Convert R_1 and R_2 into equivalent DFAs D_1 and D_2 by first using the algorithm in Kleene's theorem for converting a regular expression into an equivalent NFA (Lemma 1.55), and then converting the NFA into an equivalent DFA (Theorem 1.39).

3. Run TM M on input $\langle D_1, D_2 \rangle$, where M is the TM that decides EQ_{DFA}

4. If M accepts, *accept*. If M rejects, *reject*.

6. See Theorem 5.4.