CS 341, Fall 2011
Solutions for Midterm 2

1. (a) False, e.g., $\overline{A_{\text{TM}}}$ is not Turing-recognizable.

   (b) False, e.g., if $A = \{00, 11, 111\}$ and $B = \{00, 11\}$, then $\overline{A} \cap B = \emptyset$ but $A \neq B$. For A and B to be equal, we instead need $(\overline{A} \cap B) \cup (A \cap \overline{B}) = \emptyset$.

   (c) False. TM M may loop on input w.

   (d) True, since every regular language is context-free, every context-free language is decidable, and every decidable language is Turing-recognizable.

   (e) False, by Theorem 4.11.

   (f) False, by Corollary 4.23.

   (g) False. Can decide this by the following TM:
   $M =$ "On input $\langle N, R\rangle$, where N is an NFA and R is a regular expression:
   1. Check if $\langle N, R\rangle$ is a proper encoding of NFA N and regular expression R; if not, *reject*.
   2. Convert N into equivalent DFA D_1 using algorithm in Theorem 1.39.
   3. Convert R into equivalent DFA D_2 using algorithms in Lemma 1.55 and Theorem 1.39.
   4. Run TM S for EQ_{DFA} on input $\langle D_1, D_2\rangle$. If S accepts, then *accept*; else, *reject*."

   (h) True, by Theorem 4.5.

   (i) False, by Theorem 3.13.

   (j) False, by Corollary 3.15.

2. (a) No, because $f(x) = f(y) = 1$.

   (b) No, because nothing in A maps to $3 \in B$.

   (c) No, because f is not one-to-one nor onto.

   (d) 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.

   (e) An algorithm is a Turing machine that always halts.

3. (a) $q_1 010\#1 \quad xq_2 10\#1 \quad x1q_2 0\#1 \quad x10q_2\#1 \quad x10\#q_4 1 \quad x10\#1q_{\text{reject}}$

   (b) $q_1 1\#1 \quad xq_3\#1 \quad x\#q_5 1 \quad xq_6\#x \quad q_7 x\#x \quad xq_1\#x \quad x\#q_8 x \quad x\#xq_8 \quad x\#x\sqcup q_{\text{accept}}$

4. Slides 4-39 and 4-40.

5. Define the language as

$$E_{\text{NFA}} = \{\langle N\rangle \mid N \text{ is an NFA with } L(N) = \emptyset\}.$$

Recall that the proof of Theorem 4.4 defines a Turing machine F that decides the language $E_{\text{DFA}} = \{ \langle D \rangle \mid D \text{ is a DFA with } L(D) = \emptyset \}$. Then the following Turing machine T decides E_{NFA}:

T = "On input $\langle N \rangle$, where N is an NFA:

1. Convert N into an equivalent DFA D using the algorithm in Theorem 1.39.

2. Run TM F from Theorem 4.4 on input $\langle D \rangle$.

3. If F accepts, *accept*. If F rejects, *reject*."

6. This is Theorem 5.1, whose proof is given on slide 5-8.