CS 341, Spring 2012
Solutions for Midterm 2

1. (a) True, by Theorem 4.9.
   (b) True, by Theorem 4.9.
   (c) False, e.g., $\overline{A_{\text{TM}}}$ is not Turing-recognizable.
   (d) False. A TM M may loop on input w.
   (e) True. List the strings in lexicographic order.
   (f) False. Homework 9, problem 1.
   (g) True, by Theorems 3.13 and 3.16.
   (h) False, by Theorem 5.4.
   (i) True, by Theorem 4.5.
   (j) False, by Corollary 4.23.

2. (a) Yes, because each element in A maps to a different element in B.
   (b) No, because nothing in A maps to $4 \in B$.
   (c) No, because f is not 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) $q_1 10\#11 \quad xq_3 0\#11 \quad x0q_3\#11 \quad x0\#q_5 11 \quad x0q_6\#x1 \quad xq_7 0\#x1 \quad q_7 x0\#x1$
   $xq_1 0\#x1 \quad xxq_2\#x1 \quad xx\#q_4 x1 \quad xx\#xq_4 1 \quad xx\#x1q_{\text{reject}}$
   (b) $q_1 0\#0 \quad xq_2\#0 \quad x\#q_4 0 \quad xq_6\#x \quad q_7 x\#x \quad xq_1\#x \quad x\#q_8 x \quad x\#xq_8$
   $x\#x\sqcup q_{\text{accept}}$

4. This proof is given on slides 4-39 and 4-40. Write out the strings in Σ^* in lexicographic order as $s_1, s_2, s_3, \ldots$. Let $\mathcal{B}$ to be the set of infinite binary sequences, which we know is uncountable from HW 9, problem 1. Define the mapping $\chi : \mathcal{L} \to \mathcal{B}$ such that for any language $A \in \mathcal{L}$, we have $\chi(A)$ is the infinite binary sequence in which the ith bit is 1 if and only if $s_i \in A$. The mapping χ is one-to-one since if A_1 and A_2 are two different languages, then they differ by at least one string s_j, so then $\chi(A_1)$ and $\chi(A_2)$ differ in the jth bit. The mapping χ is onto since if $b = (b_1, b_2, b_3, \ldots) \in \mathcal{B}$ is an infinite binary sequence, then if we define the language A_b so that the string $s_i \in A_b$ if and only if $b_i = 1$, then $\chi(A_b) = b$. Since χ is both one-to-one and onto, χ is a correspondence. Thus, $\mathcal{L}$ is the same size as $\mathcal{B}$, so $\mathcal{L}$ is uncountable since $\mathcal{B}$ is uncountable.

5. Define the language as

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

Recall that the proof of Theorem 4.5 defines a Turing machine F that decides the language $EQ_{\text{DFA}} = \{\langle D, E\rangle \mid D \text{ and } E \text{ are DFAs with } L(D) = L(E)\}$. Then the following Turing machine T decides EQ_{REX}:

T = "On input $\langle R, S\rangle$, where R and S are regular expressions:

1. Convert R into an equivalent DFA D using Kleene's Theorem.

2. Convert S into an equivalent DFA E using Kleene's Theorem.

3. Run TM F from Theorem 4.4 on input $\langle D, E\rangle$.

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

6. This is Theorem 5.2, whose proof is given on slide 5-11. Specifically, suppose that E_{TM} is decidable by some TM R. Then define another TM S to decide A_{TM} as follows:

S = "On input $\langle M, w\rangle$, where M is a TM and w is a string:

1. Construct a TM M_1 from M and w as follows:

M_1 = "On input x:

1'. If $x \neq w$, *reject*.

2'. If $x = w$, run M on input w, and *accept* if M accepts."

2. Run TM R on input $\langle M_1\rangle$.

3. If R rejects, *accept*; if R accepts, *reject*."

Note that if M accepts w, then $L(M_1) = \{w\}$; if M does not accept w, then $L(M_1) = \emptyset$. Thus, $\langle M, w\rangle \in A_{\text{TM}}$ if and only if $\langle M_1\rangle \notin E_{\text{TM}}$, so S decides A_{TM}. Thus, we have reduced A_{TM} to E_{TM}. But since A_{TM} is undecidable, we must also have that E_{TM} is undecidable.