CS 341-003, Fall 2021
Solutions for Midterm 2, Hybrid

1. (a) False, by Theorem 3.16.
   (b) True, by slide 4-25.
   (c) False, e.g., if $A = \{00, 11\}$ and $B = \{00, 11, 111\}$, then $A \cap \overline{B} = \emptyset$ but $A \neq B$. For A and B to be equal, we instead need $(\overline{A} \cap B) \cup (A \cap \overline{B}) = \emptyset$.
   (d) False. A TM M may loop on input w.
   (e) False. $\overline{A_{\text{TM}}}$ is not Turing-recognizable by Corollary 4.23.
   (f) True, because the definition of Turing-decidable is more restrictive than the definition of Turing-recognizable.
   (g) True, by Theorem 3.13.
   (h) False, because the set $\mathcal{N} = \{1, 2, 3, \ldots\}$ is countable.
   (i) True, because every regular language is context-free by Corollary 2.32, and every context-free language is decidable by Theorem 4.9.
   (j) True, by slide 4-38.

2. (a) Yes, because each element of D maps to a different element in R.
   (b) No, because nothing in D maps to $2 \in R$.
   (c) No, because f is not one-to-one.
   (d) A language L_1 that is Turing-recognizable is recognized by a Turing machine M_1 that may loop forever on a string $w \notin L_1$. A language L_2 that is Turing-decidable is recognized by a Turing machine M_2 that always halts.
   (e) An algorithm is a Turing machine that always halts.

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

4. (This is problem 2a from Homework 7.) For any two decidable languages L_1 and L_2, let M_1 and M_2, respectively be the TMs that decide them. We construct a TM M' that decides the union of L_1 and L_2:

   M' = "On input string w:
   1. Run M_1 on w. If it accepts, *accept*.
   2. Run M_2 on w. If it accepts, *accept*. Otherwise, *reject*."

   To see why M' decides $L_1 \cup L_2$, first consider $w \in L_1 \cup L_2$. Then w is in L_1 or in L_2 (or both). If $w \in L_1$, then M_1 accepts w, so M' will eventually accept w. Similarly, if $w \notin L_1$ but $w \in L_2$, then M_1 will reject w because M_1 is a decider (i.e., M_1 never loops), and M_2 will accept w, so M' will eventually accept w. On the other hand, if $w \notin L_1 \cup L_2$, then $w \notin L_1$ and $w \notin L_2$. Thus, both M_1 and M_2 reject w, so M' rejects $w \notin L_1 \cup L_2$. Hence, M' decides $L_1 \cup L_2$.

5. (From slides 4-39 and 4-40). Let $\mathcal{L}$ be the collection of languages over an alphabet Σ, and let $\mathcal{B}$ be the set of infinite binary strings, which we know is uncountable (by a diagonalization argument, on slide 4-39). We will show that there is a correspondence between $\mathcal{L}$ and $\mathcal{B}$, so they have the same size. Let $s_1, s_2, s_3, \ldots$ be an enumeration of the strings in Σ^*, e.g., the enumeration can list the strings in string order. Define mapping $\chi : \mathcal{L} \to \mathcal{B}$ such that for a language $A \in \mathcal{L}$, the nth bit of $\chi(A)$ is 1 if and only if the nth string $s_n \in A$. We now show χ is a correspondence.

   - To show that χ is one-to-one, suppose that $A_1, A_2 \in \mathcal{L}$ with $A_1 \neq A_2$. Then there is some string s_i such that s_i is in one of the languages but not the other. Then $\chi(A_1)$ and $\chi(A_2)$ differ in the ith bit, so χ is one-to-one.
   - To show that χ is onto, consider any infinite binary sequence $b = b_1 b_2 b_3 \ldots \in \mathcal{B}$. Consider the language A that includes all strings s_i for which $b_i = 1$ and does not include any string b_j for which $b_j = 0$. Then $\chi(A) = b$, so χ is onto.

   Since χ is one-to-one and onto, it is a correspondence. Thus, $\mathcal{L}$ and $\mathcal{B}$ have the same size, so $\mathcal{L}$ is uncountable because $\mathcal{B}$ is uncountable.

6. (This is half of Theorem 4.22.) Because A is Turing-recognizable, there is a TM M with $L(M) = A$. Because A is co-Turing-recognizable, $\overline{A}$ is Turing-recognizable, so there is a TM M' with $L(M') = \overline{A}$. Any string $w \in \Sigma^*$ is either in A or $\overline{A}$ but not both, so either M or M' (but not both) must accept w. Now build another TM D as follows:

   D = "On input string w:

   1. Alternate running one step on each of M and M', both on input w.

   2. If M accepts w, *accept*. If M' accepts w, *reject*.

   Because exactly one of M or M' will accept w, we see that D can't loop. Also, if $w \in A$, then M is the TM that will accept, so D accepts w. If $w \notin A$, then M' is the TM that will accept, so D rejects w. Hence, D decides A, so A is decidable.

7. The language of the decision problem is

   $$A = \{ \langle N \rangle \mid N \text{ is an NFA that accepts at least one string that begins with 01 } \}.$$

   For alphabet $\Sigma = \{0, 1\}$, consider the regular expression $R = 01(0 \cup 1)^*$, so $L(R)$ is the language of strings over Σ that begins with 01. Because $L(R)$ has a regular expression, it is regular. For any NFA N, its language $L(N)$ is regular by Corollary 1.40. Let T be a Turing machine that decides E_{DFA}, as in the proof of Theorem 4.4. For a given NFA N, we have that its encoding $\langle N \rangle \in A$ if and only if $L(N) \cap L(R) \neq \emptyset$, and we know that $L(N) \cap L(R)$ is regular because the class of regular languages is closed under intersection (slide 1-34). Thus, a Turing machine that decides A is as follows:

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

1. For the regular expression $R = 01(0 \cup 1)^*$,
construct DFA D that recognizes $L(N) \cap L(R)$,
which is possible because $L(N)$ and $L(R)$ are regular,
and the class of regular languages is closed under intersection.

2. Run TM T that decides E_{DFA} on input $\langle D \rangle$.
If T rejects $\langle D \rangle$, *accept*. Otherwise, *reject*."