CS 341, Fall 2019
Solutions for Midterm 2

1. (a) False, e.g., $\overline{A_{\rm TM}}$ is not Turing-recognizable.
   (b) 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 $(\overline{A} \cap B) \cup (A \cap \overline{B}) = \emptyset$.
   (c) True, by Theorem 4.5.
   (d) False, by Homework 9, problem 1.
   (e) False, by Theorems 3.13 and 3.16.
   (f) False. A TM M may loop on input w.
   (g) True, by Theorem 4.9.
   (h) True, by slide 4-38.
   (i) False, by Theorem 4.8.
   (j) False, by Theorem 4.11.

2. (a) Yes, because $f(x) \neq f(y)$ whenever $x \neq y$.
   (b) No, because nothing in D maps to $1 \in R$.
   (c) No, because f is not onto.
   (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. $q_1 1100\#0 \quad x q_3 100\#0 \quad x1q_3 00\#0 \quad x10q_3 0\#0 \quad x100q_3\#0 \quad x100\#q_5 0 \quad x100\#0q_r$

4. This is a slight modification of Theorem 4.17. For a proof by contradiction, suppose that A is countable. The set A is clearly infinite, so the assumption that A is countable means that we can define a correspondence $f : \mathcal{N} \to A$, where $\mathcal{N} = \{1, 2, 3, \ldots\}$ is the set of natural numbers, and let $a_n = f(n)$. In other words, we can enumerate the elements of A as a list $a_1, a_2, a_3, \ldots$, where

n	$f(n) = a_n$
1	$2.d_{11}d_{12}d_{13}\ldots$
2	$2.d_{21}d_{22}d_{23}\ldots$
3	$2.d_{31}d_{32}d_{33}\ldots$
$\vdots$	$\ddots$

For the nth number a_n in the list, its ith digit after the decimal point is a_{ni}. Now we construct a number $y \in A$ as $y = 2.b_1b_2b_3\ldots$, where for each $n = 1, 2, 3, \ldots$, the nth digit in y after the decimal point is $b_n = 3$ if $d_{nn} = 1$, and $b_n = 1$ if $d_{nn} \neq 1$.

The number y belongs to the set A, but for each $n = 1, 2, 3, \ldots$, the number y but does not equal the nth number in the list because they differ in the nth digit, i.e., $b_n \neq d_{nn}$. Therefore, we get a contradiction because the list was supposed to contain all elements of A, but the list does not include $y \in A$. We thus conclude that A is uncountable.

5. This is HW 7, problem 2b. For any two Turing-recognizable languages L_1 and L_2, let M_1 and M_2, respectively, be TMs that recognize them. We construct a TM M' that recognizes the union $L_1 \cup L_2$:

   M' = "On input string w:

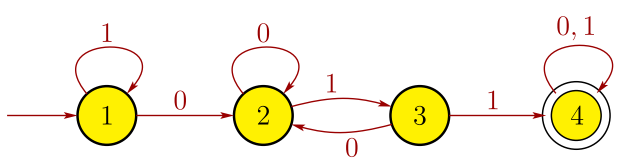
   1. Run M_1 and M_2 alternately on w, one step at a time.
   If either accepts, *accept*. If both halt and reject, *reject*.

   To see why M' recognizes $L_1 \cup L_2$, 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 $w \in L_2$, then M_2 accepts 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, neither M_1 nor M_2 accepts w, so M' will also not accept w. Hence, M' recognizes $L_1 \cup L_2$. Note that if neither M_1 nor M_2 accepts w and one of them does so by looping, then M' will loop, but this is fine because we only needed M' to *recognize* and not *decide* $L_1 \cup L_2$.

6. This is a slight modification of HW 8, problem 3. Let $\Sigma = \{0, 1\}$, and the language of the decision problem is

$$A = \{ \langle N \rangle \mid N \text{ is an NFA (with alphabet } \Sigma) \text{ that accepts at least one string } w \text{ having 011 as a substring, (i.e., } \exists \text{ string } w = x011y \text{ with } x, y \in \Sigma^*, \text{ and } N \text{ accepts } w) \}.$$

   Define the language $C = \{ w \in \Sigma^* \mid w \text{ has substring } 011 \}$. Note that C is a regular language with regular expression $(0 \cup 1)^*011(0 \cup 1)^*$ and is recognized by the following DFA D_C:



   Now consider any NFA N with alphabet Σ. If $L(N) \cap C \neq \emptyset$, then N accepts a string containing substring 011, so $\langle N \rangle \in A$. Conversely, if $L(N) \cap C = \emptyset$, then N does not accept any string containing substring 011, so $\langle N \rangle \notin A$. By Corollary 1.40, because $L(N)$ is recognized by the NFA N, the language $L(N)$ must be a regular language. Because C and $L(N)$ are regular languages, we see that $C \cap L(N)$ is regular as the class of regular languages is closed under intersection, as we saw in Chapter 1 (slide 1-34). Thus, $C \cap L(N)$ has some DFA $D_{C\cap L(N)}$. Theorem 4.4 shows that $E_{\text{DFA}} = \{ \langle B \rangle \mid B \text{ is a DFA with } L(B) = \emptyset \}$ is decidable, so there is a Turing machine H that decides E_{DFA}. We apply TM H to $\langle D_{C\cap L(N)} \rangle$ to determine if $C \cap L(N) = \emptyset$. Putting this all together gives us the following Turing machine T to decide A:

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

0. If $\langle N \rangle$ is not a proper encoding of an NFA, then *reject*.

1. Convert N into a DFA D_N using the algorithm in the proof of Theorem 1.39.

2. Construct a DFA $D_{C\cap L(N)}$ for language $C \cap L(N)$ from the DFAs D_C and D_N using the algorithm for DFA intersection.

3. Run TM H that decides E_{DFA} on input $\langle D_{C\cap L(N)} \rangle$.

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

7. This is Theorem 5.1, whose proof is given on slide 5-8. Specifically, suppose that $HALT_{\text{TM}}$ is decidable, and let R be a TM that decides $HALT_{\text{TM}}$. Thus, for any $\langle M, w \rangle$, which is an (encoded) pair of a TM M and string w, if $\langle M, w \rangle \in HALT_{\text{TM}}$ is the input to R, then R halts and accepts; if $\langle M, w \rangle \notin HALT_{\text{TM}}$ is the input to R, then R halts and rejects. To decide $HALT_{\text{TM}}$, the TM R cannot run M on w because M may loop on w, so R must use some other approach to decide $HALT_{\text{TM}}$. Now we build a TM S that decides A_{TM} using R as a subroutine.

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

1. Run TM R on input $\langle M, w \rangle$.

2. If R rejects, then *reject*.

3. If R accepts, then run M on input w.

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

Note that if M accepts w, then S accepts $\langle M, w \rangle$. If M does rejects w, then S rejects $\langle M, w \rangle$. If M loops on w, then S rejects $\langle M, w \rangle$ in stage 2. Thus, S decides A_{TM}, which is impossible because A_{TM} is undecidable by Theorem 4.11. Therefore, $HALT_{\text{TM}}$ is also undecidable.