CS 341-006, Spring 2021
Solutions for Midterm 2, Hybrid

1. (a) False. $\overline{A_{\rm TM}}$ is not Turing-recognizable by Corollary 4.23.

   (b) False. Theorem 4.11 shows that $A_{\rm TM}$ is undecidable, so no TM can decide $A_{\rm TM}$. The universal TM recognizes $A_{\rm TM}$ but doesn't decide it.

   (c) False. For example, for $\Sigma = \{0\}$, consider languages $A = \{0\}$ and $B = \{0, 00\}$, both of which are subsets of the universe Σ^*. Then $A \neq B$ and $\overline{A} \cap B = \{00\} \neq \emptyset$, but $A \cap \overline{B} = \emptyset$.

   (d) False. The set $\mathcal{N} = \{1, 2, 3, \ldots\}$ of natural numbers is infinite and countable.

   (e) True. Every finite language is context-free by slide 1-95, and every regular language is context-free by Corollary 2.32. Every context-free language is decidable by Theorem 4.9, and every decidable language is Turing-recognizable because the definition of Turing-recognizable is less restrictive than the definition of decidable (also see slide 4.55). Thus, every finite language is Turing-recognizable.

   (f) False. TM M can loop on $w \not\in L(M)$, so M never ends in $q_{\rm reject}$.

   (g) False. For any alphabet Σ, the set Σ^* is countable (just list the strings in string order). Define $\mathcal{L}$ as the set of languages over Σ, so $\mathcal{L}$ is the power set of Σ^*. But we know that $\mathcal{L}$ is uncountable (see slide 4-39).

   (h) True. This is part of Theorem 3.21.

   (i) False. Consider $\Sigma = \{a, b\}$, and for each $k = 1, 2, 3, \ldots$, define a language $L_k = \{a^k\}$. Each L_k is a regular language because $|L_k| = 1$, so L_k is decidable. Thus, there is a Turing machine M_k that decides L_k. There are infinitely many such languages L_k, so there are infinitely many corresponding TMs M_k.

   (j) False. Let $B = \Re$, which is the set of real numbers, and let $A = \{0\}$. Then $A \subseteq B$, but B is countable because it is finite.

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 $w \not\in L_1$. 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. $q_1abba\#baba$ $\quad xq_2bba\#baba$ $\quad xbq_2ba\#baba$ $\quad xbbq_2a\#baba$ $\quad xbbaq_2\#baba$ $\quad xbba\#q_4baba$ $\quad xbba\#bq_{\rm reject}aba$

4. (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.

5. (This is half of Theorem 3.21.) Suppose that A is Turing-recognizable, and we need to show that there is an enumerator that enumerates A. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}})$ be a Turing machine that recognizes A, and let $s_1, s_2, \ldots$ be an enumeration of all strings in Σ^*, e.g., in string order. We can construct an enumerator E for A as follows:

   $E =$ "Ignore the input.
   1. Repeat the following for $i = 1, 2, 3, \ldots$
   2. Run M for i steps on each of $s_1, s_2, \ldots, s_i$.
   3. If any computation accepts, print out the corresponding string s."

   The main issue is that we cannot sequentially run M on $s_1, s_2, s_3, \ldots$. The problem with doing this is that if M accepts some $s_j \in A$ but loops on $s_i \notin A$ for some $i < j$, then E will be stuck on s_i forever, so that s_j will never get printed. This is why Stage 2 runs M for only i steps on each of the first i strings.

6. The language of the decision problem is

   $$A = \{ \langle N \rangle \mid N \text{ is an NFA that accepts at least one string that has 101 as a substring} \}.$$

   For alphabet $\Sigma = \{0, 1\}$, consider the regular expression $R = (0 \cup 1)^* 101 (0 \cup 1)^*$, so $L(R)$ is the language of strings over Σ that have 101 as a substring. 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 $R = (0 \cup 1)^*101(0 \cup 1)^*$, 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*."