CS 341-452, Spring 2022, eLearning (online) Section
Solutions for Midterm 1

1. (a) False. The language with regular expression a^* is regular (by Kleene's theorem) but infinite.

   (b) True. Suppose that A is decided by a nondeterministic Turing machine. Then Corollary 3.19 implies that A is decidable, so A is decided by some deterministic Turing machine M. Thus, A is also recognized by the same deterministic TM M.

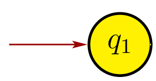
   (c) False. The DFA $M' = (Q, \Sigma, \delta, q_1, Q - F)$ recognizes $\overline{A}$, not A^*.

   (d) True. For any language A defined over an alphabet Σ, we always have that $A \cup \overline{A} = \Sigma^*$, which is regular. The fact that A is recognized by a 5-tape TM is irrelevant.

   (e) False. The CFG G gives the derivation $S \Rightarrow 1S0S \Rightarrow 10S \Rightarrow 101S0S \Rightarrow 1010S \Rightarrow 1010 \notin A$, so G does not generate A.

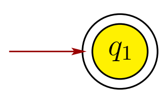
   (f) True. If A is finite, then it is also regular (see slide 1-95). Thus, $\overline{A}$ is also regular because the class of regular languages is closed under complementation (HW 2, problem 3). Corollary 2.32 then implies that $\overline{A}$ is context-free, so $\overline{A}$ has a CFG in Chomsky normal form by Theorem 2.9.

   (g) False. The following NFA

   recognizes the empty language $\emptyset$. But $\emptyset^* = \{\varepsilon\}$, which is finite.

   Alternatively, the following NFA

   recognizes the language $\{\varepsilon\}$. But $\{\varepsilon\}^* = \{\varepsilon\}$, which is finite.

   (h) False. The language $A = \{ w\#w \mid w \in \{0,1\}^* \}$ is recognized by the Turing machine described on slide 3-9. But by considering the string $s = 0^p1^p\#0^p1^p$, we can use Theorem 2.34 (the pumping lemma for context-free languages) to show that A is not context-free by slightly modifying the proof on slide 2-99 (which shows that the closely related language $D = \{ ww \mid w \in \{0,1\}^* \}$ is not context-free), so A does not have a CFG.

   (i) True. If A is regular, then so is $\overline{A}$ because the class of regular languages is closed under complementation (HW 2, problem 3). Because B is also regular, we then have that $\overline{A} \circ B$ is regular because the class of regular languages is closed under concatenation (Theorem 1.26).

   (j) False. The class of context-free languages is not closed under intersection (HW 6, problem 2a), so there are context-free languages A and B such that $A \cap B$ is not context-free. Thus, $A \cap B$ cannot have a PDA by Theorem 2.20.

2. (a) $b^*ba^*b^* \cup b^*ab^*$. Another regular expression for the language is $b^*(ba^* \cup a)b^*$. There are infinitely many correct regular expressions for the language.

(b) $\varepsilon, a, aa, ba, aaa$

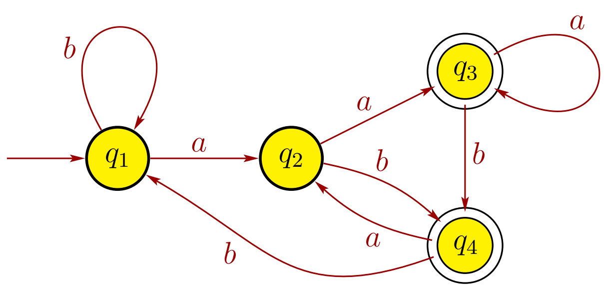
(c) After the one step of removing $X \to \varepsilon$, the CFG is then

$$\begin{aligned} S_0 &\to S \\ S &\to S1XS0 \mid S1S0 \mid \varepsilon \\ X &\to 0SX1X0 \mid 0S1X0 \mid 0SX10 \mid 0S10 \mid 0X1 \mid 01 \end{aligned}$$

(d) $G_3 = (V_3, \Sigma, R_3, S_3)$ with $V_3 = \{S_3, S_1, X_1, S_2, X_2\}$, S_3 is the start variable, $\Sigma = \{a, b\}$, and rules R_3 given by

$$\begin{aligned} S_3 &\to S_2S_1 \\ S_1 &\to aX_1aX_1S_1b \mid baa \\ X_1 &\to X_1bS_1aaX_1 \mid ab \\ S_2 &\to bX_2aS_2 \mid abb \\ X_2 &\to S_2bX_2S_2a \mid ba \end{aligned}$$

3. (a) This is a slight variation of HW 2, problem 2g. A DFA for $C = \{ w \in \Sigma^* \mid |w| \geq 2$, second-to-last symbol of w is $a \}$, with $\Sigma = \{a, b\}$, is below:



A 5-tuple description of the DFA above is $M = (Q, \Sigma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{a, b\}$
- The transition function $\delta : Q \times \Sigma \to Q$ is defined as

	a	b
q_1	q_2	q_1
q_2	q_3	q_4
q_3	q_3	q_4
q_4	q_2	q_1

- q_1 is the start state
- $F = \{q_3, q_4\}$

There are infinitely many other correct DFAs for C.

(b) A regular expression for C is $(a \cup b)^* a(a \cup b)$. Others are $(a \cup b)^*(aa \cup ab)$ and $(a \cup b)^* aa \cup (a \cup b)^* ab$. There are infinitely many other correct regular expressions for C.

4. This problem is a slight variation of HW 5, problem 1(f). A CFG for $D = \{\, b^i a^j b^k \mid i, j, k \geq 0,\ j = i + k \,\}$ is $G = (V, \Sigma, R, S)$ with $V = \{S, X, Y\}$ as the set of variables, where S is the start variable; $\Sigma = \{a, b\}$ is the set of terminals; and rules R given by

$$\begin{aligned} S &\rightarrow XY \\ X &\rightarrow bXa \mid \varepsilon \\ Y &\rightarrow aXb \mid \varepsilon \end{aligned}$$

We next prove the correctness of the CFG G, although the problem doesn't requiring providing such a proof. To see why the given CFG G works for D, we first claim that $D = A_1 \circ A_2$, where

$$\begin{aligned} A_1 &= \{\, b^i a^i \mid i \geq 0 \,\}, \\ A_2 &= \{\, a^k b^k \mid k \geq 0 \,\}. \end{aligned}$$

To prove that $D = A_1 \circ A_2$, we need to show that both $A_1 \circ A_2 \subseteq D$ and $D \subseteq A_1 \circ A_2$.

- To prove that $A_1 \circ A_2 \subseteq D$, we have to show that concatenating a string from A_1 with a string from A_2 always results in a string in D. This is true because concatenating a string $b^i a^i \in A_1$ with a string $a^k b^k \in A_2$ leads to $b^i a^i a^k b^k = b^i a^{i+k} b^k \in D$.
- Conversely, to show that $D \subseteq A_1 \circ A_2$, we need to show that every string in D can be expressed as a concatenation of a string from A_1 with a string from A_2. This is true because any string $s = b^i a^j b^k \in D$ has $j = i + k$, so $s = b^i a^{i+k} b^k = b^i a^i a^k b^k \in A_1 \circ A_2$.

A CFG G_1 for A_1 has rules $X \rightarrow bXa \mid \varepsilon$ with X as the starting variable. A CFG G_2 for A_2 has rules $Y \rightarrow aYb \mid \varepsilon$ with Y as the starting variable. As shown in HW 5, problem 3(b), the class of context-free languages is closed under concatenation, and the approach in that problem leads to the given CFG G for D.

There are infinitely many other correct CFGs for D.

5. Language $E = \{\, w \in \Sigma^* \mid w = w^{\mathcal{R}} \text{ and } w \text{ has odd length} \,\}$ with $\Sigma = \{c, d\}$ is nonregular. We prove this by contradiction. Suppose that E is a regular language. Let p be the "pumping length" of the Pumping Lemma. Consider the string

$$s = c^p d c^p.$$

Note that $s \in E$ because $s^{\mathcal{R}} = s$ and its length $|s| = 2p + 1$ is odd. Also, the length of s is $|s| = 2p + 1 > p$, so the Pumping Lemma will hold. Thus, there exists strings x, y, and z such that $s = xyz$ and

(i) $xy^i z \in E$ for each $i \geq 0$,

(ii) $|y| > 0$,

(iii) $|xy| \leq p$.

Since the first p symbols of s are all c's, the third property implies that x and y consist only of c's. So z will be the rest of the c's at the beginning, followed by dc^p. The second property states that $|y| > 0$, so y has at least one c. More precisely, we can then say that

$$\begin{aligned} x &= c^j \text{ for some } j \geq 0, \\ y &= c^k \text{ for some } k \geq 1, \\ z &= c^m dc^p \text{ for some } m \geq 0. \end{aligned}$$

Since $c^p dc^p = s = xyz = c^j c^k c^m dc^p = c^{j+k+m} dc^p$, we must have that

$$j + k + m = p, \text{ where } k \geq 1$$

by (ii). The first property implies that $xy^2 z \in E$, but

$$\begin{aligned} xy^2 z &= c^j c^k c^k c^m dc^p \\ &= c^{p+k} dc^p \notin E \end{aligned}$$

because $(c^{p+k} dc^p)^{\mathcal{R}} = c^p dc^{p+k}$ is not the same as $c^{p+k} dc^p$ since $k \geq 1$. Because the pumped string $xy^2 z \notin E$, we have a contradiction. Therefore, E is a nonregular language.

A string s that will not work for getting a contradiction is when s has only one type of symbol. For example, consider $s = c^{2p+1}$, where $s \in E$ because $|s| = 2p+1$ is odd, and $s^{\mathcal{R}} = s$. We claim that the pumping length p must be at least 2. (To see why, the proof of the pumping lemma shows that p is at least the number of states in the DFA for the language under consideration. A DFA with only 1 state can only recognize $\emptyset$ or Σ^*, neither of which is E, so we must have $p \geq 2$.) Because $|s| \geq p$, the pumping lemma will then apply. Then we could split $s = xyz$ with $x = \varepsilon$, $y = c^2$, and $z = c^{2p-1}$, which satisfy (ii) because $y \neq \varepsilon$, and (iii) because $|xy| = 2 \leq p$ as $p \geq 2$. For conclusion (i) of the pumping lemma, the pumped string $xy^i z = c^{2p+2i-1} \in E$ for each $i \geq 0$ because the pumped string is the same forwards and backwards, and has length $2(p+i) - 1$, which is odd, so there is no contradiction. There are many other strings that also won't work to get a contradiction.

6. (This is HW 7, problem 3.) The problem with the proof is that M on s_i might loop forever. If it loops forever, then E' doesn't print out s_i. More importantly, E' isn't going to move on to test the next string. Therefore, it won't be able to enumerate any other strings in the language L of the TM M. For this reason, we need to simulate M on each of the strings s_i for a fixed length of time so that no looping can occur.

7. $q_1 baba\#aab$ $\quad xq_3 aba\#aab$ $\quad xaq_3 ba\#aab$ $\quad xabq_3 a\#aab$ $\quad xabaq_3\#aab$ $\quad xaba\#q_5 aab$ $\quad xaba\#aq_{\text{reject}} ab$

8. Multiple answers

   (a) For the given statements, the following are true:

   - T is closed under union
   - R is a subset of N
   - D is closed under intersection
   - N is closed under complementation
   - N is a subset of R

   The rest are not true.

   - To show that R = P is not true, consider the language $A = \{ w\#w \mid w \in \{0,1\}^* \}$ is recognized by the Turing machine described on slide 3-9. But by considering the string $s = 0^p1^p\#0^p1^p$, we can use Theorem 2.34 (the pumping lemma for context-free languages) to show that A is not context-free by slightly modifying the proof on slide 2-99 (which shows that the closely related language $D = \{ ww \mid w \in \{0,1\}^* \}$ is not context-free), so A does not have a CFG. Thus, Theorem 2.20 implies that A does not have a PDA.

   (b) Language A is finite, so there is a DFA, NFA, PDA, Turing machine, k-tape Turing machine and nondeterministic Turing machine that will recognize A.

   (c) None of the other given statements is correct. To show that all of the other statements are not true, let $B = \{ a^nb^nc^n \mid n \geq 0\}$, which is not context-free, and consider the following choices for context-free A:

   - If $A = \{\varepsilon\}$, then $A \circ B = B$ is not context-free.
   - If $A = \emptyset$, then $A \circ B = A$ is context-free.
   - If $A = \emptyset$, then $A \cup B = B$ is not context-free.
   - If A has CFG Σ^* for $\Sigma = \{a, b, c\}$, then $A \cup B = A$ is context-free.
   - If A has CFG Σ^* for $\Sigma = \{a, b, c\}$, then $A \cap B = B$ is not context-free.
   - If $A = \emptyset$, then $A \cap B = A$ is context-free.

   (d) By Kleene's theorem, the class of languages having a regular expression is exactly the class of regular languages. So the class is closed under union (Theorem 1.25 or 1.45), concatenation (Theorem 1.26 or 1.47), Kleene star (Theorem 1.49), intersection (HW 2, problem 5), and complements (HW 2, problem 3).

   (e) The given PDA recognizes the language $A = \{ b^na^n \mid n \geq 1 \}$. Two of the given CFGs will generate A: rules

$$S \to bSa \mid ba$$

   and rules

$$S \to bSa \mid X$$
$$X \to bXa \mid ba$$

   None of the other CFGs are correct.