CS 341-009, Fall 2023, Hybrid Section
Solutions for Midterm 1

1. Multiple choice.

   1.1. Answer: (b).

   - HW 6, problem 2a, shows that the class of CFLs is not closed under intersection, so (a) is incorrect.
   - HW 5, problem 3c, shows that (b) is correct.
   - HW 6, problem 2b, shows that the class of CFLs is not closed under complementation, so (c) is incorrect.
   - The language $\{ a^n b^n c^n \mid n \geq 0 \}$ is not context-free by slide 2-96, so not all languages are context-free, so (d) is incorrect.
   - By Corollary 2.32, every regular language is also context-free, so (e) is incorrect.

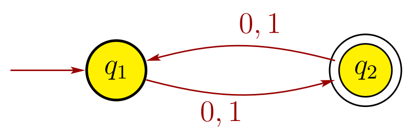
   1.2. Answer: (c).

   - The class of context-free languages is closed under union (Homework 5, problem 3a), so $B \cup C$ is context-free. Also, the class of context-free languages is closed under concatenation (Homework 5, problem 3b), ensuring that $A(B \cup C)$ is context-free, so (c) is correct.
   - We know that the class of context-free languages is *not* closed under complementation (Homework 6, problem 2b), so there exists some context-free language D whose complement $\overline{D}$ is not context-free. Also, let $B = C = \{\varepsilon\}$, which is finite, so B and C are regular (slide 1-95), making them also context-free (Corollary 2.32). Thus, $B \cup C = \{\varepsilon\}$, and let $A = D$, so $\overline{A}(B \cup C) = \overline{A} = \overline{D}$ is non-context free, making (a) incorrect.
   - Let A be any regular language, so A is also context-free (Corollary 2.32). As A is regular, $\overline{A}$ is also regular (Homework 2, problem 3), so $\overline{A}$ is also context-free (Corollary 2.32). The class of context free languages is closed under concatenation (Homework 5, problem 3b) and union (Homework 5, problem 3a), so in this case when A is regular, we have that $\overline{A}(B \cup C)$ is context-free, showing (b) is incorrect.
   - For $\Sigma = \{a, b\}$, let A be the language of all strings over Σ that don't begin with a. Now A has regular expression $\varepsilon \cup b(a \cup b)^*$, so Kleene's Theorem implies that A is a regular language, making A also context-free (Corollary 2.32). Also, $\overline{A}$ is the set of all strings over Σ that begin with a; e.g., $a \in \overline{A}$. Also, let $B = \{b\}$ and $C = \{b\}$, each of which are finite so also regular (slide 1-95) and context-free (Corollary 2.32). Also, $B \cup C = \{b\}$. Then, we have that $ab \in \overline{A}(B \cup C)$, but $ab \notin (B \cup C)\overline{A}$, making (d) incorrect.

   1.3. Answer: (b).

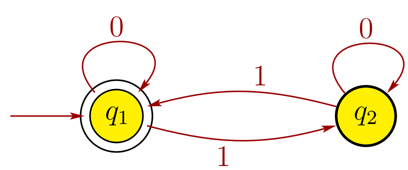
   - The regular expression $(10^*1)^* \cup ((0 \cup 1)(0 \cup 1))^*(0 \cup 1)$ cannot generate the string $0101 \in L$, so (i) is incorrect.

- The regular expression $((0 \cup 1)(0 \cup 1))^*(0 \cup 1) \cup (0^*10^*10^*)^*$ cannot generate the string $00 \in L$, so (iii) is incorrect.
- To understand the correctness of (ii), first express the language L as $L = L_1 \cup L_2$, where L_1 is the language of strings in Σ^* of odd length, and L_2 is the language of strings in Σ^* with an even number of 1's. Thus, if we have a regular express R_1 for L_1 and a regular expression R_2 for L_2, then a regular expression for $L = L_1 \cup L_2$ is $R = R_1 \cup R_2$. We can obtain regular expressions R_1 and R_2 by converting DFAs for L_1 and L_2 into regular expressions. A DFA M_1 for L_1 is



While we can use the algorithm in part of the proof of Kleene's theorem (Lemma 1.60) to convert the DFA M_1 into a regular expression R_1, the DFA is simple enough to be able to analyze it directly to obtain R_1. Specifically, note that every string accepted by M_1 has to be processed as follows:
  - start in q_1,
  - loop from q_1 back to q_1 zero or more times,
  - move from q_1 to q_2 and end in q_2.

Looping from q_1 back to q_1 corresponds to $(0 \cup 1)(0 \cup 1)$, so looping zero or more times yields $((0 \cup 1)(0 \cup 1))^*$. Moving from q_1 to q_2 happens on $(0 \cup 1)$. Thus, we get $R_1 = ((0 \cup 1)(0 \cup 1))^*(0 \cup 1)$.
A DFA M_2 recognizing L_2 is



To obtain a regular expression R_2 corresponding M_2, note that every string accepted by M_2 has to be processed as follows:
  - start in q_1,
  - loop from q_1 back to q_1 zero or more times, and end in q_1.

Looping from q_1 back to q_1 requires 0 or 10^*1, which corresponds to $0 \cup 10^*1 = 10^*1 \cup 0$, so looping zero or more times corresponds to $(10^*1 \cup 0)^*$, Thus, we get $R_2 = (10^*1 \cup 0)^*$.
Putting this all together gives $R = R_1 \cup R_2 = R_2 \cup R_1 = (10^*1 \cup 0)^* \cup ((0 \cup 1)(0 \cup 1))^*(0 \cup 1)$.

1.4. Answer: (d).
- Consider $A = \{ 0^n1^n \mid n \geq 0 \}$, which is nonregular (slide 1-105). Let $C =$

$\{ 0^{2n+1}1^{2n+1} \mid n \geq 0 \}$, so C is the set of strings in A with an odd number of 0s followed by exactly the same number of 1s, and C is infinite. Now let $B = A - C = \{ 0^{2n}1^{2n} \mid n \geq 0 \}$, so B is the set of strings in A with an even number of 0s followed by exactly the same number of 1s. We can show that B is nonregular by the pumping lemma, as follows. Suppose that B is regular, and consider $s = 0^{2p}1^{2p} \in B$, where p is the pumping length. Note that $|s| = 4p \geq p$, so the conclusions of the pumping lemma must hold. Splitting the string $s = xyz$ as in the pumping lemma leads to $x = 0^j$ for some $j \geq 0$, $y = 0^k$ for some $k \geq 1$, and $z = 0^m 0^p 1^{2p}$ for some $m \geq 0$, where $j+k+m = p$. But the pumped string $xyyz = 0^j 0^k 0^k 0^m 0^p 1^{2p} = 0^{2p+k}1^{2p} \notin B$, which is a contradiction. Thus, B is nonregular, showing (a) is incorrect.

- Consider $A = \{ 0^n 1^n \mid n \geq 0 \}$, which is nonregular (slide 1-105), and let $C = A$, which is infinite. Then $B = A - C = \emptyset$, which is regular (B has regular expression $\emptyset$, so B is regular by Kleene's theorem), so (b) is incorrect. Also, B then is also context-free (Corollary 2.32), so (c) is incorrect.

1.5. Answer: (b).

- The language A with regular expression b^* is infinite and regular, so (a) is incorrect.
- Corollary 2.32 shows that (b) is correct.
- Consider the language A with regular expression a^*b^*. Then $aab \in A$ and $abb \in A$, but their concatenation $aababb \notin A$, so (c) is incorrect. While the *class* of regular languages is closed under concatenation, an *individual* regular language may not be closed under concatenation, as the example shows.

1.6. Answer: (c).

- By Theorem 2.20, a language is context-free if and only if some PDA recognizes it, so we can answer the question by considering CFLs. The language $\{ a^n b^n \mid n \geq 0 \}$ is context-free but infinite, so (a) is incorrect.
- The language $\{\varepsilon\}$ is finite, so it is regular (slide 1-95), and Corollary 2.32 ensures it is also context-free, so (b) is incorrect.
- By Theorem 2.20, a language is context-free if and only if some PDA recognizes it, and Theorem 2.9 then guarantees that the language has a CFG in Chomsky normal form, so (d) is incorrect.

1.7. Answer: (c).

- Kleene's Theorem (Theorem 1.54) implies that L must be regular, so (a) is incorrect.
- Because L must be regular, Corollary 2.32 ensures L is also context-free, so (c) is correct and (b) is incorrect.
- For the language L with regular expression ab^*, we have that $x = ab \in L$ but $x^{\mathcal{R}} = ba \notin L$, so L is not closed under reversal, making (d) incorrect.

1.8. Answer: (c).

- The languages $L_1 = \{ a^n b^n c^n \mid n \geq 0 \}$ and $L_2 = \{ b^n a^n c^n \mid n \geq 0 \}$ are non-context-free languages (slide 2-96), with $L_1 \cap L_2 = \{\varepsilon\}$, which is regular because it is finite (slide 1-95). Thus, the intersection is also context-free by Corollary 2.32, making (a) incorrect.

- If $L_1 = L_2 = \{ a^n b^n c^n \mid n \geq 0 \}$, then $L_1 \cap L_2 = L_1$, which is non-regular and non-context-free, so (b) and (d) are incorrect.
- The previous two examples show that (c) is correct.

1.9. Answer: (e).

- Consider the language $A = \{ 0^n 1^n \mid n \geq 0 \}$, which we know is nonregular (slide 1-105). Now let $L = A^*$, which we can prove is also nonregular by the pumping lemma, which shows that (a) is incorrect. For an outline of the proof that L is nonregular, suppose that L is regular, and consider the string $s = 0^p 1^p \in L$, where p is the pumping length. Note that $|s| = 2p \geq p$, so the conclusions of the pumping lemma will hold. Thus, we can write $s = xyz$ with $x = 0^j$ for $j \geq 0$, $y = 0^k$ for $k \geq 1$, and $z = 0^m 1^p$ for $m \geq 0$, where $j + k + m = p$. But the pumped string $xyyz = 0^{p+k} 1^p$ cannot be written as a concatenation of zero of more strings from A. This contradicts the pumping lemma so L is nonregular, showing that (a) is incorrect. Also, let $B = A$, so $A \cap B = A$, which is nonregular, so (c) is also incorrect.
- For the same language $A = \{ 0^n 1^n \mid n \geq 0 \}$, let $B = \{\varepsilon\}$, so $A \circ B = A$, which we know is nonregular. Thus, (b) is incorrect.

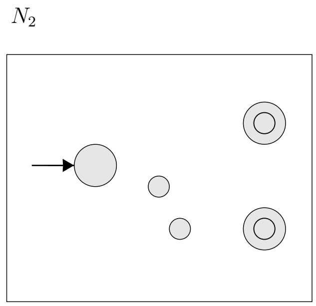
1.10. Answer: (d).

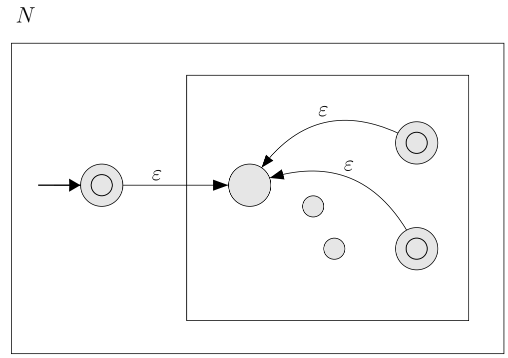
- HW 6, problem 4, shows that A is non-context-free, so (d) is correct.
- Because A is non-context-free, Theorem 2.20 shows that A cannot have a PDA, making (c) incorrect.
- Also, A being non-context-free implies that A is also not regular (Corollary 2.32), so (a) and (b) are incorrect. We can see that the regular expression $(00)^*(111)^*(0)^*$ in (a) is wrong because it generates the string $00 \notin A$.

2. (a) $((a \cup b)(a \cup b))^*(a \cup b)b$.
There are infinitely many other correct regular expressions for the language, e.g.,
$((a \cup b)(a \cup b))^* ab \cup ((a \cup b)(a \cup b))^* bb$, or
$ab \cup bb \cup ((a \cup b)(a \cup b))^*(a \cup b)b$, or $\ldots$.
Some incorrect answers include
  - $((a \cup b)b)^*$, which generates $\varepsilon \notin A$ and cannot generate $aaab \in A$;
  - $((a \cup b)(a \cup b)(a \cup b))^* b$, which generates $b \notin A$ and cannot generate $aaaaab \in A$;
  - $(aa \cup bb)^*(aa \cup ab)$, which can't generate $abab \in A$;
  - $(a \cup b)^n b$ for n odd, which is not a regular expression.

   (b) $a^*(ba \cup \varepsilon)b^*(a \cup \varepsilon)b^*$, or $(a^*ba \cup a^*)b^*(a \cup \varepsilon)b^*$, or $\ldots$. There are infinitely many other correct regular expressions for this language.

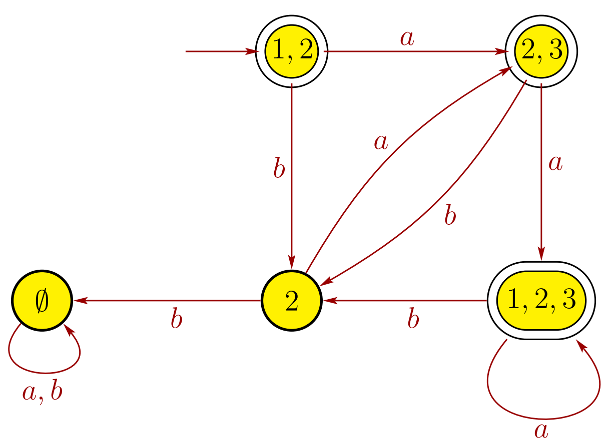
   (c) As on slide 1-63 of the notes, if A_1 is defined by NFA N_1 and A_2 is defined by NFA N_2, then an NFA N for $A_3 = A_2^*$ is as below:





(d) (Homework 5, problem 3b.) Assume that $S_3 \notin V_1 \cup V_2$, and $V_1 \cap V_2 = \emptyset$ is given. Then a CFG for $A_3 = A_2 \circ A_1$ is $G_3 = (V_3, \Sigma, R_3, S_3)$ with $V_3 = V_1 \cup V_2 \cup \{S_3\}$ and $R_3 = R_1 \cup R_2 \cup \{ S_3 \to S_2 S_1 \}$.

3. A DFA for C is below:

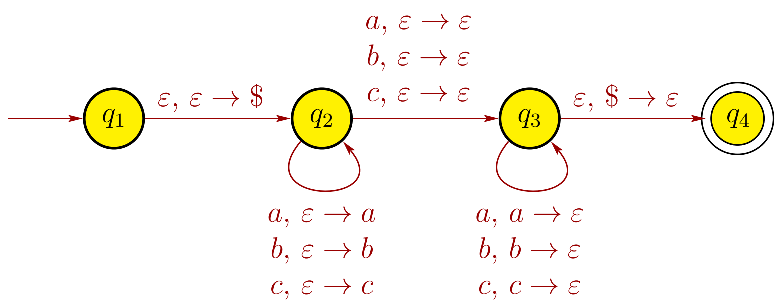


4. (a) For $\Sigma = \{a, b, c\}$, let $L = \{ w \in \Sigma^* \mid w = w^{\mathcal{R}}, |w| \text{ is odd} \}$ be the language, which is odd-length palindromes in Σ^*. A CFG $G = (V, \Sigma, R, S)$ for L has $V = \{S\}$ with S the starting variable, $\Sigma = \{a, b, c\}$, and rules

$$S \to aSa \mid bSb \mid cSc \mid a \mid b \mid c$$

There are infinitely many other correct CFGs for L.

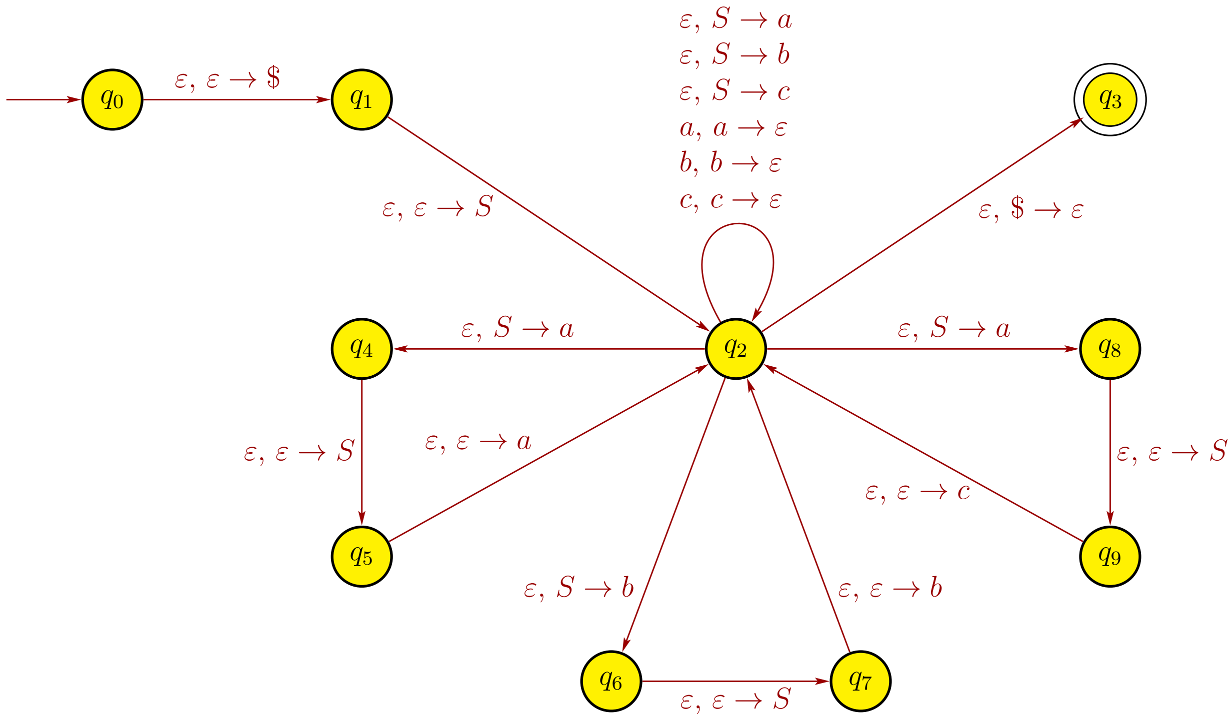
(b) There are infinitely many correct PDAs for L. Here is one:



The language consists of odd-length palindromes. Each string w has length $n = 2k+1$ for some $k \geq 0$, and the first k symbols are the reverse of the last k symbols, and the symbol in the middle is unmatched. In the above PDA

- state q_2 pushes an a for each a read, pushes an b for each b read, and pushes an c for each c read, for the first k symbols,
- the transition from q_2 to q_3 reads the middle symbol in w without matching it to anything,
- state q_3 reads in the last k symbols, matching them with the reverse of the first k symbols in the stack,
- the transition from state q_3 to q_4 pops \$ to make sure the stack is empty before accepting.

Another approach uses the algorithm from Lemma 2.21 to convert the CFG in part (a) into a PDA.



Note that

- The path $q_2 \to q_4 \to q_5 \to q_2$ corresponds to the rule $S \to aSa$, where the symbols on the right side of the rule are pushed in reverse order.
- The path $q_2 \to q_6 \to q_7 \to q_2$ corresponds to the rule $S \to bSb$, where the symbols on the right side of the rule are pushed in reverse order.
- The path $q_2 \to q_8 \to q_9 \to q_2$ corresponds to the rule $S \to cSc$, where the symbols on the right side of the rule are pushed in reverse order.

5. For $\Sigma = \{a, b, c\}$, the language $A = \{\, w \in \Sigma^* \mid w = w^{\mathcal{R}}, |w| \text{ is odd} \,\}$ is nonregular. We prove this by contradiction. Suppose that A is a regular language. Let p be the "pumping length" of the Pumping Lemma. Consider the string $s = a^p b a^p$, where $s \in A$ because $s = s^{\mathcal{R}}$ and $|s| = 2p + 1$ is odd. Also, we have that $|s| = 2p + 1 \geq p$, so the Pumping Lemma will hold. Thus, there exist strings x, y, and z such that $s = xyz$ and

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

   (b) $|y| > 0$,

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

   Because the first p symbols of s are all a's, the third property implies that x and y consist only of a's. So z will be the rest of the first set of a's (possibly none), followed by ba^p. The second property states that $|y| > 0$, so y has at least one a. More precisely, we can then say that

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

Because

$$a^p b a^p = s = xyz = a^j a^k a^m b a^p = a^{j+k+m} b a^p,$$

we must have that

$$j + k + m = p \quad \text{and} \quad k \geq 1.$$

The first property implies that the pumped string $xy^2z \in A$, but

$$\begin{aligned} xy^2z &= a^j a^k a^k a^m b a^p \\ &= a^{p+k} b a^p \notin A \end{aligned}$$

because it is not a palindrome since $k \geq 1$. This contradicts the first property of the pumping lemma. Therefore, A is a nonregular language.