CS 341-005, Fall 2022, Face-to-Face Section
Solutions for Midterm 1

1. Multiple choice.

   1.1. 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 infinite, non-context-free languages, 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.

   1.2. Answer: (a).
   - By slide 1-95, L must be regular, making (a) correct, and (b) and (c) incorrect.
   - For the language $L = \{a, b\}$, note that $x = a \in L$ and $y = b \in L$, but $xy = ab \notin L$, so L is not closed under Kleene star, making (d) incorrect.

   1.3. Answer: (e).
   - The regular expression b^*a^* generates the string $bba \notin A$, so (a) is incorrect.
   - The regular expression $(ba)^*$ generates the string $baba \notin A$, so (b) is incorrect.
   - The language $L(G)$ of the given CFG G in part (c) is $L(G) = \emptyset$ (i.e., no strings at all) because derivations can never terminate: $S \Rightarrow bSa \Rightarrow bbSaa \Rightarrow bbbSaaa \Rightarrow \cdots$, so (c) is incorrect.
   - The language A has CFG with rules $S \to bSa \mid \varepsilon$, so (d) is incorrect.

   1.4. Answer: (c).
   - HW 6, problem 2a, shows that the class of CFLs is not closed under intersection, so (a) is incorrect.
   - HW 6, problem 2b, shows that the class of CFLs is not closed under complementation, so (b) is incorrect.
   - HW 5, problem 3c, shows that (c) is correct.
   - 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.

   1.5. Answer: (a).
   - Theorem 1.25 implies that $L_1 \cup L_2$ must be regular, whether each of L_1 and L_2 is finite or infinite. By Corollary 2.32, every regular language (even if it is infinite) is also context-free, so (a) is correct.
   - By Corollary 2.32, every regular language (even if it is infinite) is also context-free, so (b) and (c) are incorrect.
   - Theorem 1.25 implies that $L_1 \cup L_2$ must be regular, so (d) is incorrect.

1.6. Answer: (h).

- The regular expression $(00 \cup 11)^*$ cannot generate the string $1010 \in L$, so (i) is incorrect.
- The regular expression $(00 \cup 11 \cup (01 \cup 10)(01 \cup 10))^*$ cannot generate the string $010001 \in L$, so (ii) is incorrect.
- The regular expression $((01 \cup 10)(00 \cup 11)^*(01 \cup 10))^*$ cannot generate the string $11 \in L$, so (iii) is incorrect.

1.7. Answer: (d).

- The language $A = \{ a^n b^n c^n \mid n \geq 0 \}$ is non-context-free and infinite, so (a) is incorrect. In fact, if A is non-context-free language, A must be infinite.
- $A = \{ a^n b^n c^n \mid n \geq 0 \}$ is also nonregular, so (b) is incorrect.
- $A = \{ a^n b^n c^n \mid n \geq 0 \}$ is non-context-free, and $abc \in A$ but $(abc)^{\mathcal{R}} = cba \notin A$, so A is not closed under reversals, making (c) incorrect.

1.8. 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.

1.9. 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 $x = ab \in L$ and $y = abb \in L$, but $xy = ababb \notin L$, so L is not closed under concatenation, making (d) incorrect.

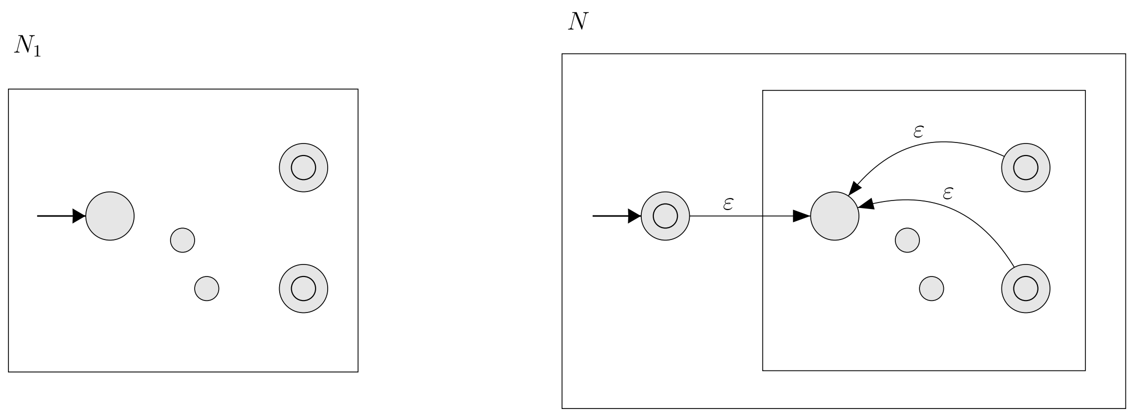
1.10. Answer: (b).

- HW 4, problem 5c, shows that (a) is incorrect, and that (b) is correct.
- Slightly modifying the proof on slide 1-105 shows that the language $L_1 = \{ a^n b^n \mid n \geq 1 \}$ is non-regular. Adding ε to L_1 leads to the language $L_2 = \{ a^n b^n \mid n \geq 0 \}$, which is context-free (with CFG having rules $S \rightarrow aSb \mid \varepsilon$), so (c) is incorrect.
- Slightly modifying the proof on slide 2-96 shows that the language $L_1 = \{ a^n b^n c^n \mid n \geq 1 \}$ is non-context-free, so it is also non-regular by Corollary 2.32. Adding ε to L_1 leads to the language $L_2 = \{ a^n b^n c^n \mid n \geq 0 \}$, which is non-context-free by the proof on slide 2-96, so (d) is incorrect.

2. (a) $a^*ba^*(a^*ba^*ba^*)^*$. This is essentially HW 2, problem 4b.
There are infinitely many other correct regular expressions for this language, such as $(a^*ba^*ba^*)^*a^*ba^*$
or $a^*b(a^*ba^*ba^*)^*a^*$
or $a^*ba^*(a^*ba^*ba^*)^*a^*$ or ....
Some incorrect regular expressions include $b(bb)^*$, $a^*b^*(a^*b^*a^*b^*a^*)^*a^*$, etc.

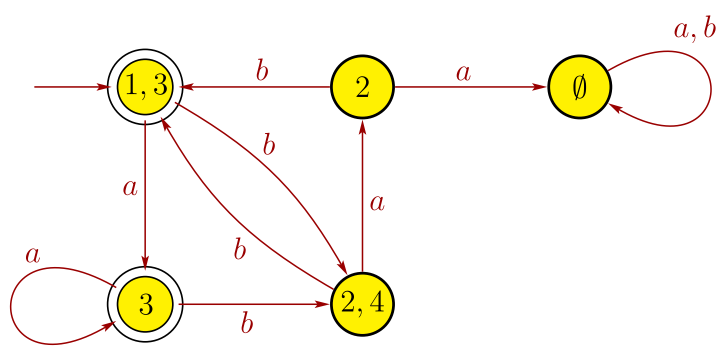
(b) $(ab^*(a \cup b) \cup b)(a \cup b)a^*$. Another regular expression is $ab^*(a \cup b)(a \cup b)a^* \cup b(a \cup b)a^*$. There are infinitely many correct regular expressions for this language.

(c) As on slide 1-66 of the notes, if A_1 is defined by NFA N_1, then an NFA N for A_1^* 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 $A_1 \cup A_2$ is $G_3 = (V_3, \Sigma, R_3, S_3)$ with $V_3 = V_1 \cup V_2 \cup \{S_3\}$ and $R_3 = R_1 \cup R_2 \cup \{ S_3 \to S_1 | S_2 \}$.

3. A DFA for C is below:



4. (a) This is essentially HW 5, problem 1f. Let $L = \{ c^i b^j a^k \mid i, j, k \geq 0 \text{ and } j = i + k \}$ be the language given in the problem, and define other languages

$$L_1 = \{ c^i b^i \mid i \geq 0 \},$$
$$L_2 = \{ b^k a^k \mid k \geq 0 \}.$$

Note that $L = L_1 \circ L_2$ because concatenating any string $c^i b^i \in L_1$ with any string $b^k a^k \in L_2$ results in a string $c^i b^i b^k a^k = c^i b^{i+k} a^k \in L$. Thus, if L_1 has a CFG $G_1 = (V_1, \Sigma, R_1, S_1)$, and L_2 has a CFG $G_2 = (V_2, \Sigma, R_2, S_2)$, we can construct a CFG for $L = L_1 \circ L_2$ by using the approach in HW 5, problem 3b. Specifically,

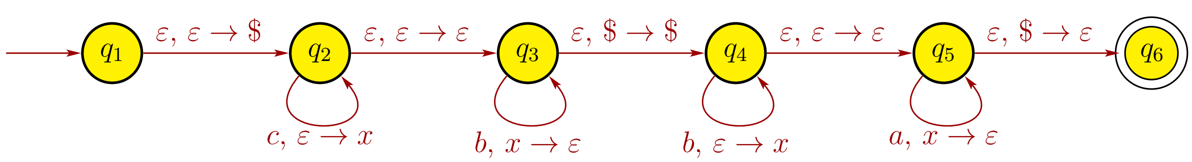
- L_1 has a CFG $G_1 = (V_1, \Sigma, R_1, S_1)$, with $V_1 = \{S_1\}$, $\Sigma = \{a, b, c\}$, S_1 as the starting variable, and rules $S_1 \to cS_1b \mid \varepsilon$ in R_1;
- L_2 has a CFG $G_2 = (V_2, \Sigma, R_2, S_2)$, with $V_2 = \{S_2\}$, $\Sigma = \{a, b, c\}$, S_2 as the starting variable, and rules $S_2 \to bS_2a \mid \varepsilon$ in R_2.

Even though $\Sigma = \{a, b, c\}$ for both CFGs G_1 and G_2, CFG G_1 never generates a string with c, and CFG G_2 never generates a string with b. Then a CFG $G_3 = (V_3, \Sigma, R_3, S_3)$ for L has $V_3 = V_1 \cup V_2 \cup \{S_3\} = \{S_1, S_2, S_3\}$ with S_3 the starting variable, $\Sigma = \{a, b, c\}$, and rules

$$\begin{aligned} S_3 &\to S_1S_2 \\ S_1 &\to cS_1b \mid \varepsilon \\ S_2 &\to bS_2a \mid \varepsilon \end{aligned}$$

There are infinitely many other correct CFGs for L.

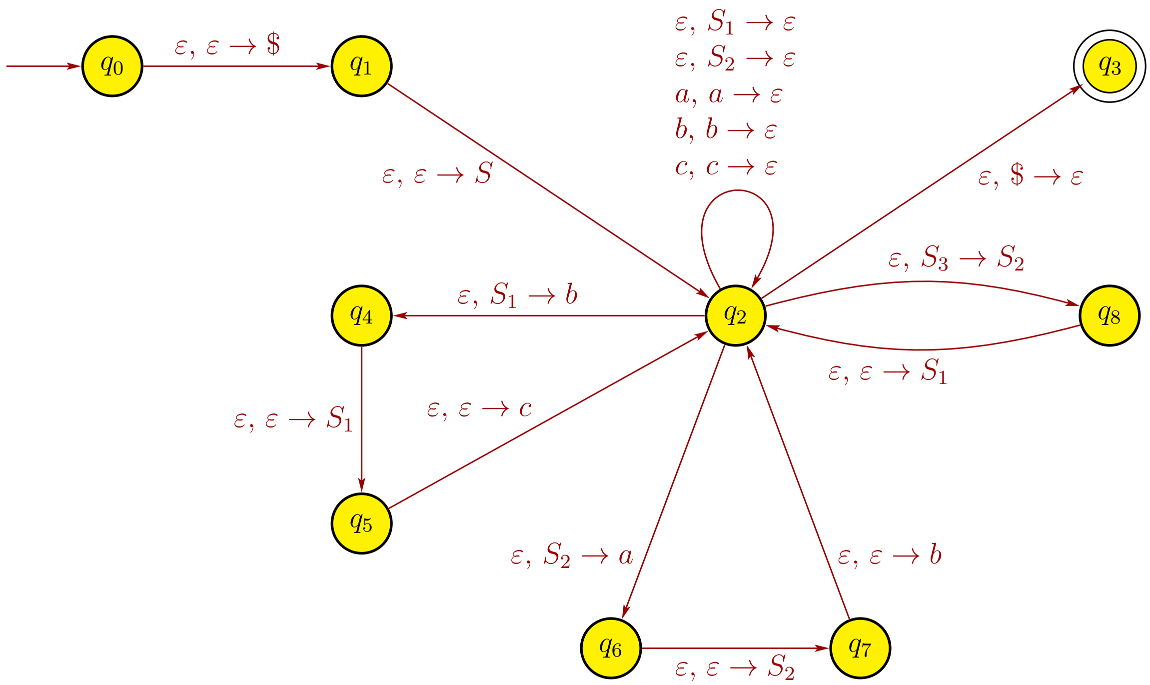
(b) This is essentially HW 6, problem 1g. There are infinitely many correct PDAs for L. Here is one:



In the above PDA,

- state q_2 pushes an x for each c read,
- state q_3 reads a b and pops an x for each c read in state q_2,
- the transition from q_3 to q_4 makes sure the stack is empty, and pushes another \$ on the stack,
- state q_4 pushes an x for each additional b read,
- state q_5 reads an a and pops an x to match each additional b read in state q_4.

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 $S_1 \to cS_1b$, 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 $S_2 \to bS_2a$, where the symbols on the right side of the rule are pushed in reverse order.
- The path $q_2 \to q_8 \to q_2$ corresponds to the rule $S_3 \to S_1S_2$, where the symbols on the right side of the rule are pushed in reverse order.

5. Language $A = \{\, c^i b^j a^k \mid i, j, k \geq 0 \text{ and } j = i + k \,\}$ 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 $s = c^p b^p$ Note that $s \in A$ because $s = c^i b^k a^k$ for $i = j = p$ and $k = 0$. Also, we have that $|s| = p > p$, so the Pumping Lemma will hold. Thus, there exist strings x, y, and z such that $s = xyz$ and

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

   (b) $|y| > 0$,

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

   Because 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 first set of c's (possibly none), followed by b^p. The second property states that $|y| > 0$, so y has at least one c. More precisely, we can then say that

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

Because

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

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

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

since $k \geq 1$, so in the pumped string, the number of b's does not equal the sum of the number of c's and a's. This contradicts the first property of the pumping lemma. Therefore, A is a nonregular language.

Another possible string that will result in a contradiction is $s = c^p b^{2p} a^p \in A$, where $|s| = 4p > p$. Then splitting $s = xyz$ satisfying properties (ii) and (iii) of the pumping lemma will lead to

$$
\begin{aligned}
x &= c^j \text{ for some } j \geq 0, \\
y &= c^k \text{ for some } k \geq 1, \\
z &= c^m b^{2p} a^p \text{ for some } m \geq 0,
\end{aligned}
$$

where $j + k + m = p$. Property (i) of the pumping lemma states that $xyyz \in A$, but $xyyz = c^{p+k} b^{2p} a^p \notin A$ because $2p \neq p + k + p = 2p + k$ since $k > 0$, giving a contradiction.