CS 341-006, Spring 2023, Face-to-Face Section
Solutions for Midterm 1

1. Multiple choice.

   1.1. Answer: (d).

   - For the alphabet $\Sigma = \{0, 1\}$, consider the language $A = \{ 0^n1^n \mid n \geq 0 \} \subseteq \Sigma^*$, and slide 1-105 proves A is nonregular. Define B as the complement of A, so $B = \overline{A} = \Sigma^* - A$, which must also be nonregular. To see why, suppose that B is regular; then $\overline{B} = \Sigma^* - B$ is regular because the class of regular languages is closed under complementation (Homework 2, problem 3). But $\overline{B} = A$, making A regular, which is a contradiction because A is nonregular. Now $A \cup B = \Sigma^*$, which is regular by Kleene's theorem since it has a regular expression, so (d) is correct, and (c) is incorrect. Also, for this example, $A \cap B = \emptyset$, which is regular by slide 1-95 because $\emptyset$ is finite, so (a) is incorrect.
   - To show that choice (b) is incorrect, suppose that A and B are nonregular. Now A must be infinite because if it were finite, then it would be regular by slide 1-95. We always have that $A \subseteq A \cup B$, so $|A \cup B| \geq |A| = \infty$, proving that $A \cup B$ must be infinite.

   1.2. 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 $\{ a^nb^n \mid n \geq 0 \}$ 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. Combining this item and the previous shows that choice (c) is correct.
   - Theorem 2.9 guarantees that a context-free language has a CFG in Chomsky normal form, so (d) is incorrect.

   1.3. Answer: (e).

   - Suppose that $B = \Sigma^*$ for $\Sigma = \{0, 1\}$, so B is regular because it has a regular expression, and B is also context-free by Corollary 2.32. We next give examples of various languages A with $A \subseteq B$ to show that (a), (b), (c), and (d) are incorrect.
     - Consider $A = \{ 0^n1^n \mid n \geq 0\} \subseteq B$, where A is nonregular by slide 1-105, showing that (a) is incorrect. Also, A is context-free (slide 2-5), so (d) is incorrect.
     - Consider $A = \emptyset \subseteq B$, where A is regular since it has regular expression $\emptyset$, making (b) incorrect. Also, A is context-free by Corollary 2.32, so (d) is incorrect.
     - Consider $A = \{ ww \mid w \in \Sigma^* \} \subseteq B$, where A is non-context-free by slide 2-99, so (c) is incorrect.

   1.4. 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 $S \to aSb \mid \varepsilon$), 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.

1.5. 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 given CFG G in part (c) has language $L(G) = \emptyset$ (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.

1.6. 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 3b, 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, making (d) incorrect.
- By Corollary 2.32, every regular language is also context-free, so (e) is incorrect.

1.7. Answer: (b).
- Suppose that A has regular expression $(aa)^*a$, so A is the set of strings of a's of odd length. Because A has a regular expression, it is regular by Kleene's Theorem. Note that $a \in A$ and $aaa \in A$, but their concatenation $aaaa \notin A$, so A is not closed under concatenation, showing that (a) is incorrect, so (d) is also incorrect. (While this example shows that a *particular* regular language may not be closed under concatenation, the *class* of regular languages *is* closed under concatenation.) Also, the same language A is infinite, showing that (c) is incorrect.
- Corollary 2.32 shows that A must be context-free, so (b) is correct.

1.8. Answer: (c)
- Homework 1, problem 5 shows that (c) is correct.
- The language $A = \{\varepsilon\}$ has $A^* = \{\varepsilon\}$, which is finite, so (a) is incorrect.
- The language $A = \{\varepsilon, a\}$ has $A \circ A = \{\varepsilon, a, aa\} \neq A$, making (b) incorrect. The same language A has $A^* = \{ a^n \mid n \geq 0\} \neq A$, so (d) is incorrect.

1.9. Answer: (a).

- Slide 1-95 shows that L must be regular, so (a) is correct, and (b) and (c) are incorrect.
- For the language $L_1 = \{a\}$, the string $aa \in L_1$ but $aa \notin L_1$, making (d) incorrect.

1.10. Answer: (h).

- The regular expression $(0 \cup 1)^*(01 \cup 10)$ cannot generate the string $1 \in L$, so (i) is incorrect.
- The regular expression $(0 \cup 1)^*(0 \cup 1 \cup 01 \cup 10)$ generates the string $00 \notin L$, so (ii) is incorrect.
- The regular expression $(0 \cup 1)^*(01 \cup 10) \cup 0 \cup 1$ cannot generate the string $\varepsilon \in L$, so (iii) is incorrect.

2. (a) $(a \cup b)^* bab$

There are infinitely many other correct regular expressions for this language, such as $\Sigma^* bab$
or $(\varepsilon \cup a \cup b)^* bab$
or $bab \cup (a \cup b)(a \cup b)^* bab$
or $bab \cup abab \cup bbab \cup (a \cup b)^* bab$
or $(a^* b^*)^* bab$ or ....

There are also infinitely many incorrect regular expressions. For example, the regular expression $(bab)^*$ is wrong because it can't generate $abab \in A$ nor $bbab \in A$; it also incorrectly generates $\varepsilon \notin A$. Also, $a^* b^* aba$ is wrong it can't generate $bababa \in A$.

(b) $aa^* \cup aab^* aa^* \cup bb^* aa^*$

There are infinitely many other correct regular expressions for this language, such as $(a \cup aab^* a \cup bb^* a)a^*$
or $(a \cup (aa \cup b)b^* a)a^*$
or $(a(\varepsilon \cup ab^* a) \cup bb^* a)a^*$ or ...

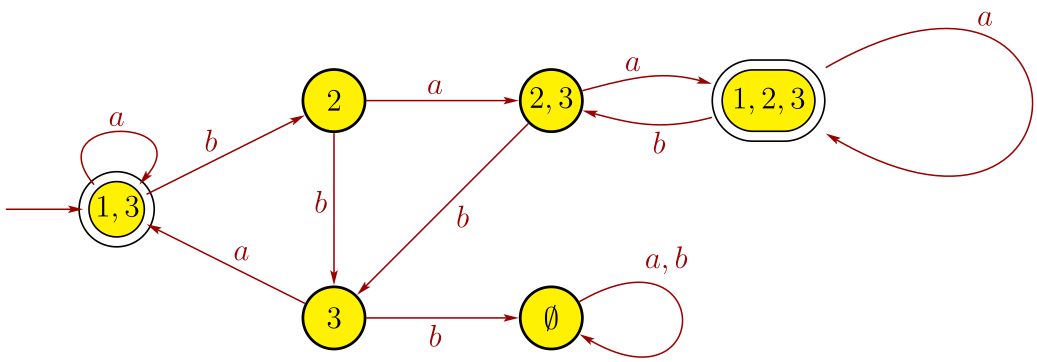
(c) After performing the one step, the CFG is then

$$
\begin{aligned}
S_0 &\rightarrow S \\
S &\rightarrow A0AS \mid 0AS \mid A0S \mid 0S \mid 0AS1S \mid 0S1S \mid \varepsilon \\
A &\rightarrow 1A0S \mid 10S
\end{aligned}
$$

(d) (Slides 1-32 and 1-33) Given a DFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ for language A_1 and a DFA $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ for language A_2, the language $A_3 = A_1 \cup A_2$ is recognized by the DFA $M_3 = (Q_3, \Sigma, \delta_3, q_3, F_3)$, with

- $Q_3 = Q_1 \times Q_2$,
- $\delta_3((x, y), \ell) = (\delta_1(x, \ell), \delta_2(y, \ell))$ for $(x, y) \in Q_3$ and $\ell \in \Sigma$,
- $q_3 = (q_1, q_2)$, and
- $F_3 = (F_1 \times Q_2) \cup (Q_1 \times F_2) = \{ (x, y) \in Q_3 \mid x \in F_1 \text{ or } y \in F_2 \}$.

3. A DFA for C is below:



4. (a) $G = (V, \Sigma, R, S)$ with set of variables $V = \{S, W, X, Y, Z\}$, where S is the start variable; set of terminals $\Sigma = \{a, b, c\}$; and rules

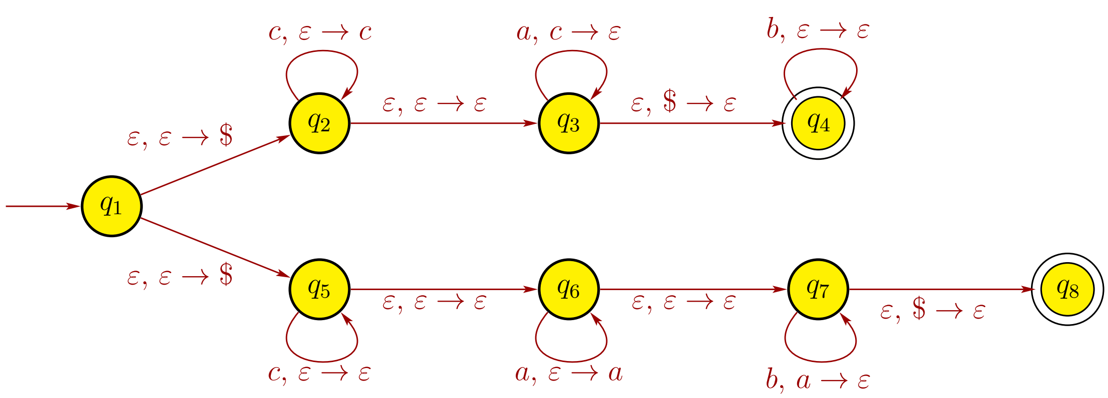
$$\begin{aligned} S &\to WX \mid YZ \\ W &\to cWa \mid \varepsilon \\ X &\to bX \mid \varepsilon \\ Y &\to cY \mid \varepsilon \\ Z &\to aZb \mid \varepsilon \end{aligned}$$

Starting from variable W, the derived string will be in $A_1 = \{ c^n a^n \mid n \geq 0 \}$. Starting from variable X, the derived string will be in $A_2 = L(b^*)$. So if $S \Rightarrow WX$ is the first step taken in a derivation, then the resulting string will be in the language $B_1 = A_1 \circ A_2 = \{ c^n a^n b^k \mid n \geq 0, k \geq 0 \}$.

Similarly, starting from the variable Y, the derived string will be in $A_3 = L(c^*)$. Starting from the variable Z, the derived string will be in $A_4 = \{ a^n b^n \mid n \geq 0 \}$. So if $S \Rightarrow YZ$ is the first step taken in a derivation, then the resulting string will be in the language $B_2 = \{ c^i a^n b^n \mid i \geq 0, n \geq 0 \}$. Finally, we get $L = B_1 \cup B_2$.

There are infinitely many other correct CFGs for L.

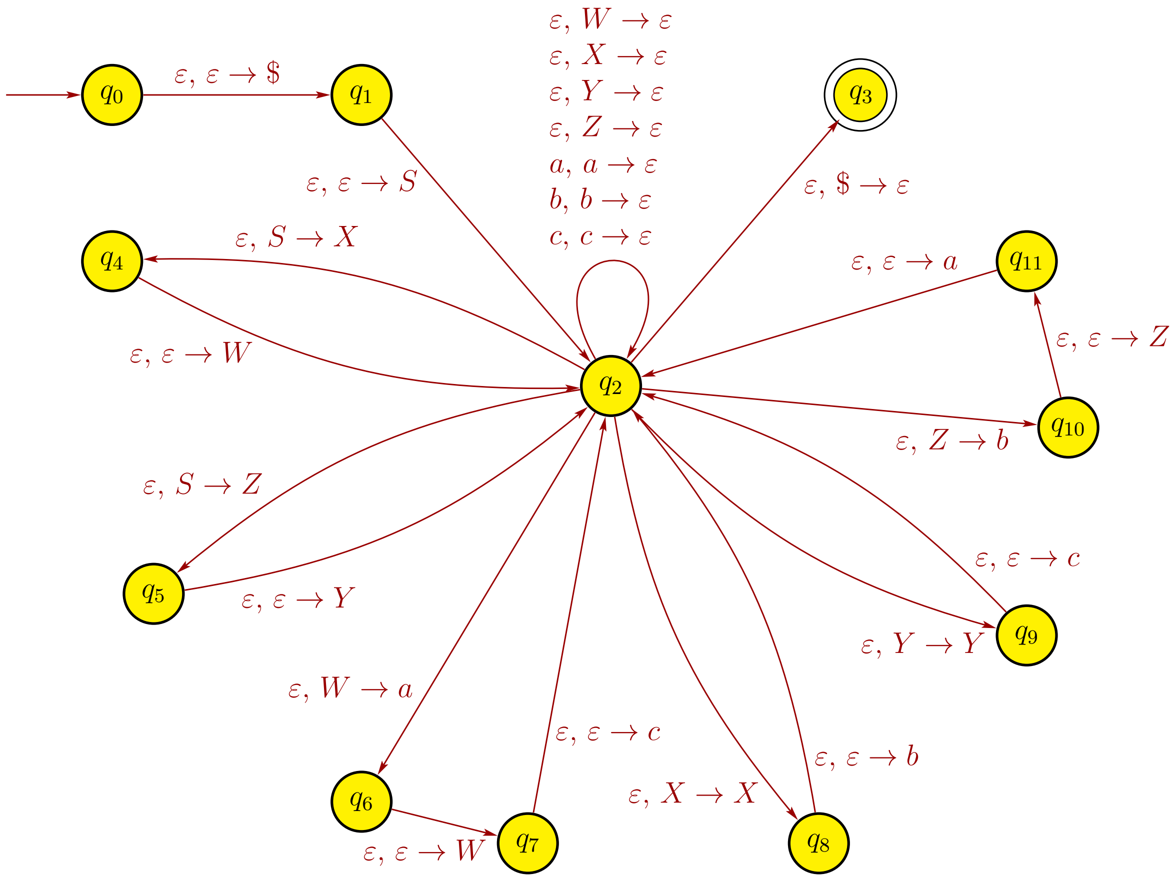
(b) There are infinitely many correct PDAs for L. Here is one:



The PDA has a nondeterministic branch at q_1.

- If the string is $c^i a^j b^k$ with $i = j$, then the PDA can accept the string by first taking the branch from q_1 to q_2.
- If the string is $c^i a^j b^k$ with $j = k$, then the PDA can accept the string by first taking the branch from q_1 to q_5.

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



Note that

- The path $q_2 \to q_4 \to q_2$ corresponds to the rule $S \to WX$.
- The path $q_2 \to q_5 \to q_2$ corresponds to the rule $S \to YZ$.
- The path $q_2 \to q_6 \to q_7 \to q_2$ corresponds to the rule $W \to cWa$.
- The path $q_2 \to q_8 \to q_2$ corresponds to the rule $X \to bX$.
- The path $q_2 \to q_9 \to q_2$ corresponds to the rule $Y \to cY$.
- The path $q_2 \to q_{10} \to q_{11} \to q_2$ corresponds to the rule $Z \to aZb$.

5. Language $A = \{ c^i a^j b^k \mid i, j, k \geq 0, \text{ and } i = j \text{ or } j = k \}$ 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 $s = c^p a^p$. Note that $s \in A$ because the numbers of c's and a's are equal, and $|s| = 2p \geq p$, so the Pumping Lemma will hold. Thus, there exists strings x, y, and z such that $s = xyz$ and

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

(b) $|y| > 0$,

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

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, followed by a^p. The second property states that $|y| > 0$, so y has at least one c. More precisely, we can then say that

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

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

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

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

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

since $p + k > p$ because $j + k + m = p$ and $k \geq 1$, so the number of c's in the pumped string xy^2z doesn't match the number of a's, and the number of a's doesn't match the number of b's (none). Because the pumped string $xy^2z \notin A$, we have a contradiction. Therefore, A is a nonregular language.

Note that if you instead chose the string $s = c^p a^p b^p \in A$, you would not get a contradiction. This is because pumping up or down leads to the number of c's changing, but the number of a's and b's remain the same and equal. Thus, the pumped string is still in the language, so there is no contradiction.

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

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

where $j + k + m = p$. Property (i) of the pumping lemma states that $xyyz \in A$, but $xyyz = a^{p+k} b^p \notin A$ because $k \geq 1$, giving a contradiction.