CS 341-007, Fall 2023, Hybrid Section
Solutions for Midterm 1

1. Multiple choice.

   1.1. Answer: (d).
   Because A is recognized by an NFA, A must be regular by Corollary 1.4. Because B has a regular expression, B must be regular by Kleene's theorem (Theorem 1.54).
   - We must then have that $A \circ B$ is regular by Theorem 1.47, so (a) is incorrect.
   - We must also then have that $A \cup B$ is regular by Theorem 1.45, so $A \cup B$ is recognized by a DFA, making (b) incorrect.
   - By the theorem on slide 1-34, $A \cap B$ must be regular, so Corollary 2.32 ensures that $A \cap B$ is also context-free, so (c) is incorrect. Also, (d) is correct by Theorem 2.20.

   1.2. Answer: (c).
   - 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 A has CFG with rules $S \to bSa \mid \varepsilon$, so (d) is incorrect.

   1.3. 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.

   1.4. 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 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.

   1.5. 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. 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.

1.6. Answer: (e).

- Suppose that $A = \emptyset$ for $\Sigma = \{0, 1\}$, so A is regular because it has a regular expression $\emptyset$. For every language B, we always have that $A \subseteq B$, and we know that there are languages B that are nonregular, regular, non-context-free, and context-free, showing that (a), (b), (c), and (d) are all incorrect.

1.7. Answer: (a)

- Suppose that $w \in A$ with $w \neq \varepsilon$. Then $w^n \in A^*$ for each $n \geq 0$, where $w^i \neq w^j$ for each $i \neq j$ because $w \neq \varepsilon$. Thus, A^* is infinite, so (a) is correct.
- To show that (b), (c), and (d) are incorrect, consider $A = \{b\}$. Then $A \circ A = \{bb\} \neq A$, so (b) is incorrect. Also, $A^+ = \{ b^n \mid n \geq 1 \} \neq \{ b^n \mid n \geq 0 \} = A^*$ because $\varepsilon \notin A^+$ but $\varepsilon \in A^*$, so (c) is incorrect. Note that $A^* \neq A$, so (d) is incorrect.

1.8. Answer: (c).

- $\varepsilon \notin \emptyset$, so (a) is incorrect.
- $\emptyset$ is the empty set, and ε is the empty string, so they aren't equal, making (b) incorrect.
- $\emptyset^* = \{\varepsilon\}$ making (c) correct, and (d) incorrect.

1.9. Answer: (c).

- The regular expression $(0 \cup 1)^*(01 \cup 10) \cup \varepsilon$ cannot generate the string $1 \in L$, so (i) is incorrect.
- The regular expression $(0 \cup 1)^*(\varepsilon \cup 0 \cup 1 \cup 01 \cup 10)$ generates the string $00 \notin L$, so (ii) is incorrect.
- A correct regular expression is $(0 \cup 1)^*(01 \cup 10) \cup 0 \cup 1 \cup \varepsilon$, which is essentially from HW 3, problem 4e.

1.10. Answer: (d).

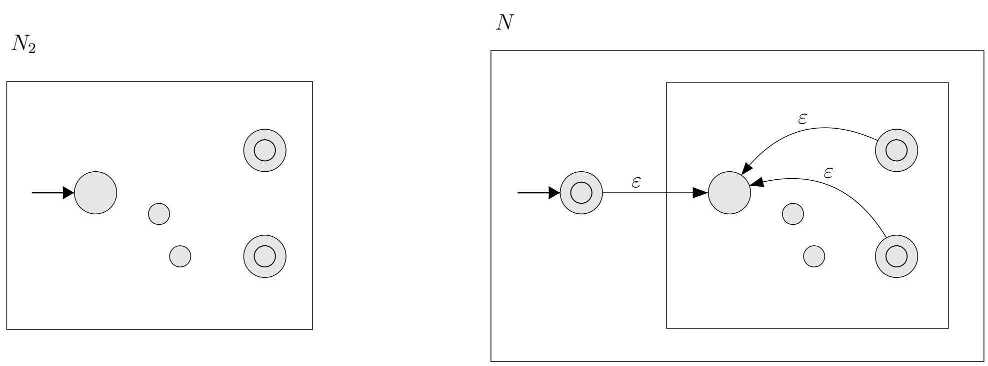
- For the alphabet $\Sigma = \{0, 1\}$, consider the language $A = \{ 0^n 1^n \mid n \geq 0 \} \subseteq \Sigma^*$, and slide 1-105 proves A is nonregular. Define B as the complement of A, so $B = \overline{A} = \Sigma^* - A$, which must also be nonregular. To see why, suppose that B is regular; then $\overline{B} = \Sigma^* - B$ 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.

2. (a) $((a \cup b)(a \cup b)(a \cup b))^*(a \cup b)(a \cup b)a$.
   There are infinitely many other correct regular expressions for the language, e.g., $((a \cup b)(a \cup b)(a \cup b))^*(a \cup b)(a \cup b)aa \cup ((a \cup b)(a \cup b)(a \cup b))^*(a \cup b)(a \cup b)ba$, or $aaa \cup aba \cup baa \cup bba \cup ((a \cup b)(a \cup b)(a \cup b))^*(a \cup b)(a \cup b)a$, or ....
   Some incorrect answers include
   - $((a \cup b)(a \cup b)a)^*$, which generates $\varepsilon \notin A$ and cannot generate $aabaaa \in A$;
   - $((a \cup b)(a \cup b)(a \cup b))^*a$, which generates $a \notin A$ and cannot generate $a^6 \in A$;
   - $(aaa \cup bbb)^*(aaa \cup aba \cup baa \cup bba)$, which can't generate $abaaba \in A$;
   - $(a \cup b)^n a$ with $n = 3k - 1$ for some $k \geq 1$, which is not a regular expression.

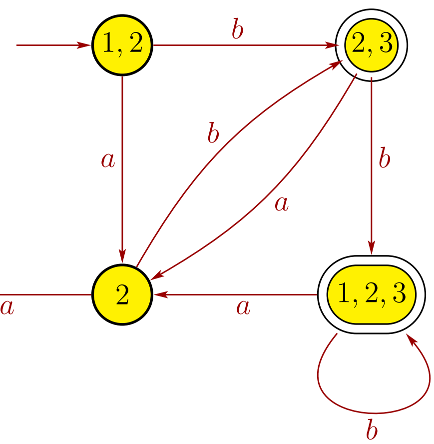
   (b) $(ba^*a \cup \varepsilon)b^*(a \cup \varepsilon)b^*$, or $(ba^*ab^* \cup b^*)(a \cup \varepsilon)b^*$, or .... There are infinitely many other correct regular expressions for this language.

   (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 3a.) A CFG for $A_3 = A_2 \cup A_1$ is $G_3 = (V_3, \Sigma, R_3, S_3)$ with $V_3 = V_1 \cup V_2 \cup \{S_3\}$, $S_3 \notin V_1 \cup V_2$, and $R_3 = R_1 \cup R_2 \cup \{ S_3 \to S_1 \mid S_2 \}$.

3. A DFA for C is below:



4. (a) For $\Sigma = \{a, b\}$, we have the language

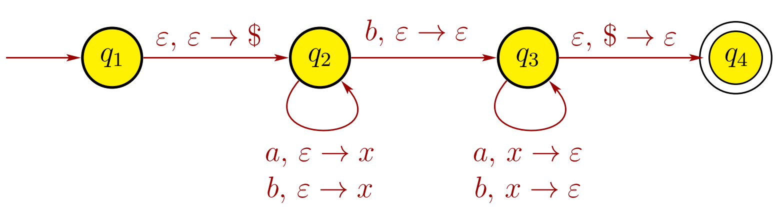
$$L = \{\, w \in \Sigma^* \mid |w| \text{ is odd, and the middle symbol of } w \text{ is } b \,\},$$

which is essentially the same as in HW 5, problem 1c. A CFG $G = (V, \Sigma, R, S)$ for L has $V = \{S\}$ with S the starting variable, $\Sigma = \{a, b\}$, and rules

$$S \to aSa \mid aSb \mid bSa \mid bSb \mid b$$

There are infinitely many other correct CFGs for L.

(b) There are infinitely many correct PDAs for L. Here is one, which we will call M_1:

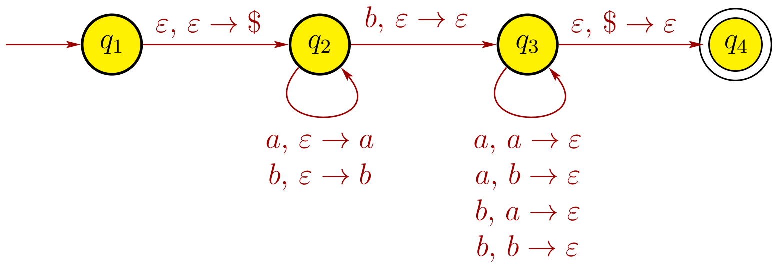


Each string w has length $n = 2k - 1$ for some $k \geq 1$, where the middle symbol of w is in position k. The number of symbols before the middle has to equal the number of symbols after the middle, but the language doesn't require matching specific symbols in corresponding positions before and after the middle. In the above PDA

- state q_2 pushes an x for each symbol read before the middle,
- the transition from q_2 to q_3 guesses that it is now reading the middle symbol in w, which has to be b, without matching it to anything,
- state q_3 reads in the symbols after the middle, popping an x for each one,

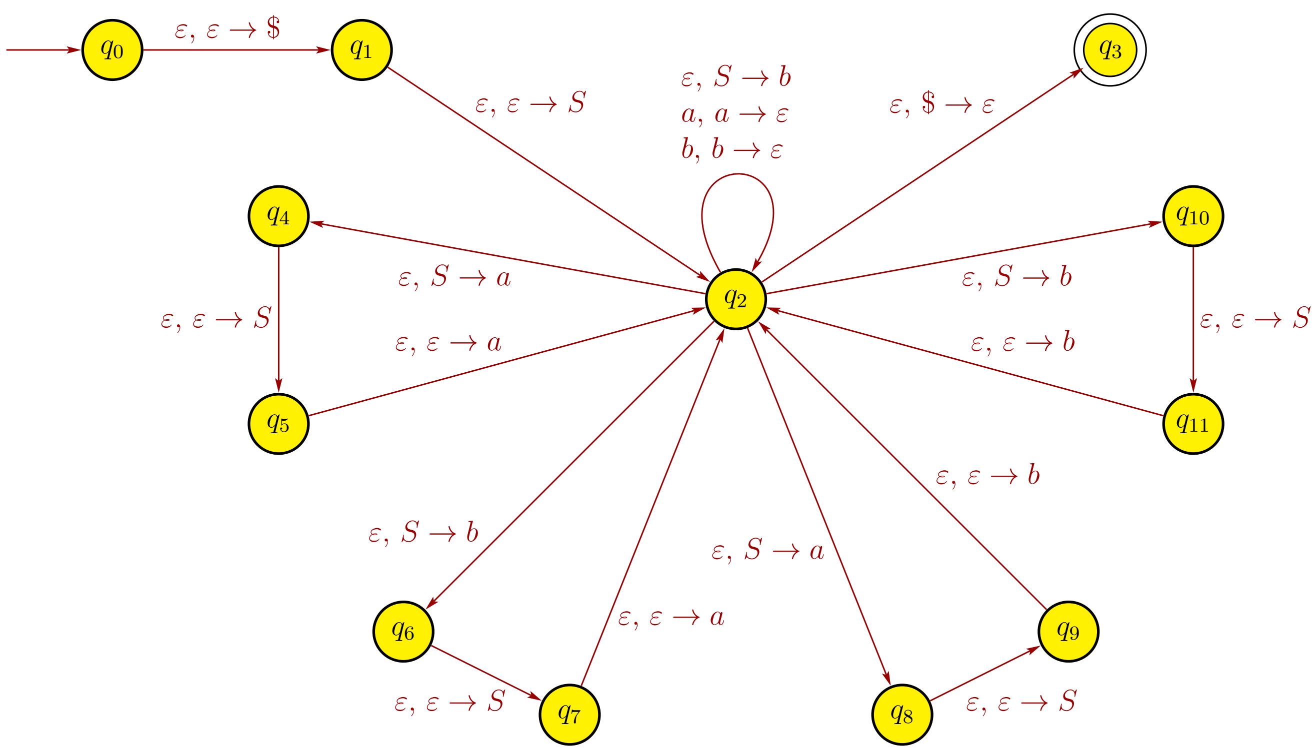
- the transition from state q_3 to q_4 pops \$ to make sure the stack is empty.

Another PDA M_2 for L is as follows:



For M_2, looping in state q_2 pushes the same symbol onto the stack that is read. But because strings in L don't have to match symbols before and after the middle, looping in state q_3 can pop anything on each symbol read from Σ.

Another approach to build a PDA for L uses the algorithm from Lemma 2.21 to convert the CFG in part (a) into a PDA, which we will call M_3.



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 $S \to aSb$, 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 $S \to bSa$, where the symbols on the right side of the rule are pushed in reverse order.
- The path $q_2 \to q_{10} \to q_{11} \to q_2$ corresponds to the rule $S \to bSb$, where the symbols on the right side of the rule are pushed in reverse order.

5. For $\Sigma = \{a, b\}$, the language

$$L = \{\, w \in \Sigma^* \mid |w| \text{ is odd, and the middle symbol of } w \text{ is } b \,\}$$

is nonregular. We prove this by contradiction. Suppose that A is a regular language. Let p be the "pumping length" of the pumping lemma for regular languages. Consider the string $s = a^p b a^p$, where $s \in A$ because $|s| = 2p + 1$ is odd, and the middle symbol of s is b. 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 of only 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

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

To see why $xy^2z \notin A$, there are two cases of y to consider to cover all possibilities: either $|y|$ is even or $|y|$ is odd.

- If $|y|$ is even, then $xyyz$ has odd length because $s = xyz$ has odd length since $s \in A$, so s has a middle symbol. But because $s = xyz = a^p b a^p$, the middle symbol of $xyyz = a^{p+k} b a^p$ is now to the left of b. Thus, the middle symbol is now a, implying that $xyyz \notin A$.

- If $|y|$ is odd, then because $s = xyz$ and $|s|$ is odd (since $s \in A$), we must then have that $xyyz$ has even length, so $xyyz \notin A$.

Thus, both cases have that $xy^2z \notin A$. Because the two cases cover all possibilities for y and each violates property (i) of the pumping lemma, we get a contradiction. Therefore, A is a nonregular language.