CS 341-010, Spring 2024, 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 non-context-free languages (slide 2-96), with $L_1 \cap L_2 = \{\varepsilon\}$, which is regular because it is finite (slide 1-95), so (d) is incorrect. 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) is incorrect.
   - The previous two examples show that (c) is correct.

   1.2. Answer: (c).

   - 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 will show To prove that L is nonregular, suppose for contradiction that L is regular. Let p be the pumping length, and consider the string $s = 0^p 1^p \in L$. 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.
   - For alphabet $\Sigma = \{0, 1\}$, let $A = \{ 0^n 1^n \mid n \geq 0 \}$, which is nonregular (slide 1-105). Then $\overline{A} = \Sigma^* - A$, which we now prove is nonregular. Suppose for contradiction that $\overline{A}$ is regular. Then the complement $\overline{\overline{A}}$ of $\overline{A}$ must be regular since the class of regular languages is closed under complementation (HW 2, problem 3). But $\overline{\overline{A}} = A$, which is nonregular, giving a contradiction, so $\overline{A}$ is nonregular, making (b) incorrect.
   - For any language A over any alphabet Σ, we have that $\overline{A} = \Sigma^* - A$, and $\overline{A} \cup A = \Sigma^*$, which is regular by Kleene's Theorem (Theorem 1.54) because it has a regular expression Σ^*. Thus (c) is correct.

   1.3. Answer: (d).

   - Slightly modifying 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 $(000)^*(11)^*(0)^*$ in (a) is wrong because it generates the string $000 \notin A$.

1.4. Answer: (d).

Because A is recognized by an NFA, A must be regular by Corollary 1.40. 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 because the class of regular languages is closed under concatenation (Theorem 1.47), so (a) is incorrect.
- We must also then have that $A \cup B$ is regular because the class of regular languages is closed under union (Theorem 1.45), so $A \cup B$ is recognized by some DFA, making (b) incorrect.
- By HW 2, problem 5, $A \cap B$ 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.

1.5. Answer: (d).

- Consider $A = \Sigma^*$ for $\Sigma = \{a, b\}$. Because A has regular expression $(a \cup b)^*$, Kleene's theorem (Theorem 1.54) ensures that A is regular, so Corollary 2.32 implies that A is a CFL. Let $C = \{ a^n b^n \mid n \geq 0 \}$, which is infinite and nonregular (slide 1-105). Remove C from A to get B, so $B = A - C = \Sigma^* - C = \overline{C}$, which we now show is nonregular. For a contradiction, suppose that $\overline{C}$ is regular. Then the complement $\overline{\overline{C}}$ of $\overline{C}$ must be regular because the class of regular languages is closed under complements (HW 2, problem 3). But $\overline{\overline{C}} = C$, which is nonregular, giving a contradiction. So we have an example where removing an infinite number of strings from a context-free language A results in a nonregular language B, showing (a) is incorrect.
- Consider $A = \Sigma^*$ for $\Sigma = \{a, b\}$, so A is regular and context-free, and let $C = A$, which is infinite. Then $B = A - C = \emptyset$, which is regular because it is finite (slide 1-95), so (b) is incorrect. Also, B then is also context-free (Corollary 2.32), so (c) is incorrect.

1.6. 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 (Theorem 1.54). 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 the *class* of regular languages *is* closed under concatenation by Theorem 1.47, this example shows that a *particular* regular language may not be 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, and the class of context-free languages is closed under concatenation (HW 5, problem 3b), so (b) is correct.

1.7. Answer: (a).

- The class of languages recognized by NFAs is the same as the class of regular languages by Corollary 1.40. Thus, this class of languages is closed under complementation (HW 2, problem 3). Hence, option (a) is correct.

- Since the class of regular languages is closed under intersection (HW 2, problem 5), option (b) is incorrect.
- The language $\{ 0^n 1^n \mid n \geq 0 \}$ is nonregular (slide 1-105), so it is not recognized by any NFA (Corollary 1.40), making option (c) incorrect. Also, this language is context-free (slide 2-5), making (d) incorrect.

1.8. Answer: (b).

- The language A has regular expression $1^* 0^*$, so Kleene's Theorem (Theorem 1.54) implies that A is regular. Thus, (b) is correct, and (c) is incorrect. If a language is non-context-free, then it must also be non-regular (Corollary 2.32), so (d) is incorrect.
- Each string $1^i \in A$ for $i \geq 0$, so A is infinite, making (a) incorrect.

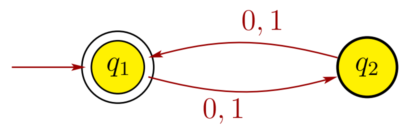
1.9. Answer: (d).

- 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 Kleene-star (Homework 5, problem 3c), so A^* is context-free. Finally, the class of context-free languages is closed under concatenation (Homework 5, problem 3b), so A^* is context-free, ensuring that $A^*(B \cup C)$ is context-free, so (d) 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). Also, $B^* = \{\varepsilon\}$, so $B^* \cup C = \{\varepsilon\}$. 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 because the class of regular languages is closed under complementation (Homework 2, problem 3), so $\overline{A}$ is also context-free (Corollary 2.32). The class of CFLs is closed under Kleene-star (Homework 5, problem 3c), so C^* is context-free. The class of CFLs is closed under union (Homework 5, problem 3a), so $B \cup C^*$ is context-free. The class of CFLs is closed under concatenation (Homework 5, problem 3b), 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 (Theorem 1.54) 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, $B \cup C = \{b\} = C \cup B$. Then, we have that $ab \in \overline{A}(B \cup C)$, but $ab \notin (C \cup B)\overline{A}$, so $\overline{A}(B \cup C) \neq (C \cup B)\overline{A}$, making (c) incorrect.

1.10. Answer: (f).

For $\Sigma = \{0, 1\}$, let A be the language of all strings over Σ that have even length or an odd number of 1's. Note that $A = A_1 \cup A_2$, where A_1 is the language of strings in Σ^* of even length, and A_2 is the language of strings in Σ^* with an odd number of 1's. Thus, if we have a regular express r_1 for A_1 and a regular expression r_2 for A_2, then a regular expression for $A = A_1 \cup A_2$ is $R = r_1 \cup r_2$.

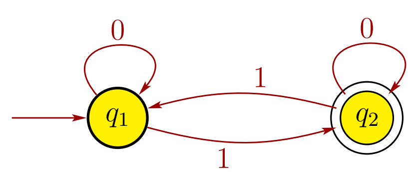
- The regular expression $R_1 = ((0 \cup 1)(0 \cup 1))^* \cup (0^*10^* \cup 0^*1)(0^*10^*1)^*$ cannot generate the string $w = 11100 \in A$, so (i) is incorrect.
- We now show that $R_2 = (00 \cup 01 \cup 10 \cup 11)^* \cup 0^*1(0 \cup 10^*1)^*$ in (ii) satisfies $A = L(R_2)$. We can obtain regular expressions r_1 and r_2 for A_1 and A_2, respectively, by converting DFAs for the languages into regular expressions. A DFA M_1 for A_1 is



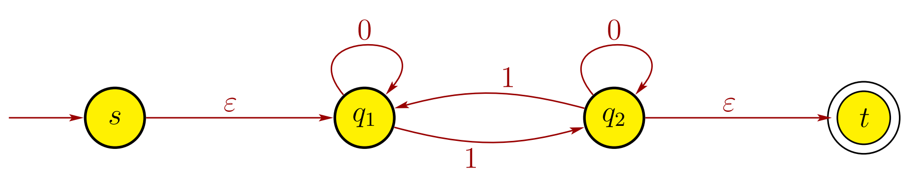
While we can use the algorithm in part of the proof of Kleene's theorem (Lemma 1.60) to convert the DFA M_1 into a regular expression r_1, the DFA is simple enough to be able to analyze it directly to obtain r_1. Specifically, note that every string accepted by M_1 has to be processed as follows:

  - start in q_1,
  - loop from q_1 back to q_1 zero or more times.

Looping from q_1 back to q_1 corresponds to $(0 \cup 1)(0 \cup 1) = (00 \cup 01 \cup 10 \cup 11)$, so looping zero or more times yields $((0 \cup 1)(0 \cup 1))^*$ or $(00 \cup 01 \cup 10 \cup 11)^*$. Thus, we get $r_1 = ((0 \cup 1)(0 \cup 1))^*$ and $r_1' = (00 \cup 01 \cup 10 \cup 11)^*$ as regular expressions for A_1.
A DFA M_2 recognizing A_2 is



To obtain a regular expression corresponding M_2, we apply the algorithm in Kleene's theorem (Lemma 1.60) to convert M_2 into a regular expression r_2 as follows. First, convert M_2 into an equivalent GNFA:



We will first eliminate state q_1, so we define $C = \{s, q_2\}$ as the set of states (except for q_1) with edges directly into q_1, and $D = \{q_2\}$ as the set of states (except for q_1) with edges directly from q_1. Eliminating q_1 by taking into account all paths going directly from a state in C to q_1, looping in q_1 zero or more times, and then directly going to a state in D results in

Next removing state q_2 results in the regular expression $r_2 = 0^*1(0 \cup 10^*1)^*$ for A_2. Thus, a regular expression for $A = A_1 \cup A_2$ is $R = r_1' \cup r_2 = (00 \cup 01 \cup 10 \cup 11)^* \cup 0^*1(0 \cup 10^*1)^*$, which is R_2.

- For $R_3 = 0^*10^*(0^*10^*10^*)^* \cup ((0 \cup 1)(0 \cup 1))^*$ in (iii), we can show that $A = L(R_3)$ as follows. First, write $A = A_2 \cup A_1$, with A_1 and A_2 as defined above. We can again use regular expression $r_1 = ((0 \cup 1)(0 \cup 1))^*$ for A_1. For A_2, note that $r_3 = 0^*10^*$ defines the language of strings in Σ^* with exactly a single 1. Also, $r_4 = (0^*10^*10^*)^*$ defines the language of strings in Σ^* with an even number of 1s. Concatenating these two gives a regular expression $r_2' = r_3 r_4 = 0^*10^*(0^*10^*10^*)^*$ for the language of strings in Σ^* with an odd number of 1s. Thus, a regular expression for A is $r_2' \cup r_1 = 0^*10^*(0^*10^*10^*)^* \cup ((0 \cup 1)(0 \cup 1))^*$, which is R_3.

1.11. Answer: (c).

- The regular expression generates $01 \notin A$, so (a) is incorrect. In fact, the language A is not regular, so A does not have a regular expression.
- The CFG can yield $S \Rightarrow 1$, but $1 \notin A$ because the string has odd length, so (b) is incorrect.
- The language A is even-length palindromes of 0's and 1's, and the CFG in part (c) is correct, as seen in HW 5, problem 1(b).

1.12. Answer: (e).

The language $A = \{\, b^i a^j \mid i \geq 0, j \geq 0, i = j \,\} = \{\, b^n a^n \mid n \geq 0 \,\}$.

- 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 option (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.

1.13. Answer: (d).

- 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 the class of CFLs.

The language $\{ a^n b^n \mid n \geq 0 \}$ is context-free (by a slight variation of slide 2-5) but not regular (by a slight variation of slide 1-105), so it cannot have a regular expression, making (a) incorrect. This CFL is also infinite, so (b) 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 (c) is incorrect. Using the example from the previous item then shows that a CFL can be finite and it can be infinite, so (d) is correct.

1.14. Answer: (b).

- Suppose that L is the language with regular expression a^*b, which includes a Kleene-star $*$ but $\varepsilon \notin L$, so (a) is incorrect.
- Because L has a regular expression, L must be regular by Kleene's Theorem. Corollary 2.32 ensures L is also context-free, so L must have a context-free grammar in Chomsky normal form by Theorem 2.9, showing (b) is correct.
- If L has regular expression ε^*, which includes a Kleene-star $*$, then $L = \{\varepsilon\}$, which is a finite language, making (c) incorrect.

1.15. Answer: (c).

- The string $[^n]^n \in A$ for each $n \geq 1$, so A is infinite, making (a) incorrect.
- We can prove that A is nonregular using the pumping lemma, as follows. Suppose that A is regular, and let p be the pumping length. Consider the string $s = [^p]^p \in A$, and note that $|s| = 2p \geq p$, so the conclusions of the pumping lemma will hold. Thus, we can split $s = xyz$ such that $xy^iz \in A$ for all $i \geq 0$, $|y| > 0$, and $|xy| \leq p$. The last property implies that x and y have only left brackets, so $x = [^j$ for some $j \geq 0$, $y = [^k$ for some $k \geq 1$ (using the second property), and $z = [^m]^p$ for some $m \geq 0$, where $j + k + m = p$ because $[^p]^p = s = xyz = [^j[^k[^m]^p = [^{j+k+m}]^p$. The pumping lemma implies that $xyyz \in A$, where $xyyz = [^j[^k[^k[^m]^p = [^{p+k}]^p$ because $j + k + m = p$. However $[^{p+k}]^p \notin A$ because $k \geq 1$, so the pumping lemma does not hold, proving that A is nonregular.
- HW 5, problem 1(i) gives the following CFG G for A: $G = (V, \Sigma, R, S)$ with set of variables $V = \{S\}$, where S is the start variable; set of terminals $\Sigma = \{ [,] \}$; and rules

$$S \rightarrow \varepsilon \mid SS \mid [S]$$

Thus, A is a CFG, so (c) is correct, and (d) is incorrect.

1.16. Answer: (e).

- We know that $A = \{ a^n b^n \mid n \geq 0 \}$ is nonregular (slide 1-105). Let $B = A$, so $A \subseteq B$ with B nonregular, so (a) is incorrect. If we instead let $B = \Sigma^*$ for $\Sigma = \{a, b\}$, then B is regular because it has a regular expression, making (b) incorrect.
- The language $A = \{ a^n b^n c^n \mid n \geq 0 \}$ is non-context-free (slide 2-96), so A is also nonregular by Corollary 2.32. Let $B = A$, so $A \subseteq B$ with B non-

context-free, so (c) is incorrect.
If we instead let $B = \Sigma^*$ for $\Sigma = \{a, b, c\}$, then $A \subseteq B$ and B is regular because it has a regular expression, so B is also context-free (Corollary 2.32), making (d) incorrect.

1.17. Answer: (a)

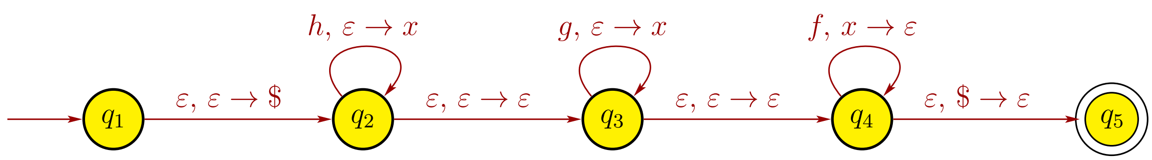
- Suppose that $w \in A$ with $w \neq \varepsilon$. Then $w^n \in A^*$ for each $n \geq 0$, where $w^i \neq w^j$ for each $i \neq j$ because $w \neq \varepsilon$. Thus, A^* is infinite, so (a) is correct.
- To show that (b), (c), and (d) are incorrect, consider $A = \{b\}$. Then $A \circ A = \{bb\} \neq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

1.18. 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.

1.19. Answer: (h).

- For $\Sigma = \{f, g, h\}$, the PDA



  recognizes the language $A = \{\, h^i g^j f^k \mid i, j, k \geq 0 \text{ and } i + j = k \,\}$, which is a slight variation of HW 6, problem 1(e). Note that A differs from L_1, L_2, and L_3. For example, the string $h^1 g^2 f^3 \in A$ but $h^1 g^2 f^3 \notin L_m$ for $m = 1, 2, 3$. Thus, (h) is correct.
- The languages L_1 and L_2 are context-free, so they have PDAs (HW 6, problem 1(g) is a slight variation of L_2) but just not the one that is given. On the other hand, L_3 is not context-free (slight variation of slide 2-96), so L_3 does not have a PDA (Theorem 2.20).

1.20. Answer: (a).

- If A is non-context-free, it must also be nonregular by Corollary 2.32, so option (a) is correct.
- The language $B = \Sigma^*$ is both context-free and regular, so option (b) is incorrect. Also, $\overline{B} = \emptyset$, which is finite, so option (c) is incorrect.
- For the alphabet $\Sigma = \{a, b, c\}$, consider the language $A = \{\, a^n b^n c^n \mid n \geq 0 \,\} \subseteq \Sigma^*$, and slide 2-96 proves A is non-context-free. Define $B = \Sigma^*$, which is regular so it is also context-free by Corollary 2.32. But $A \cup B = B$ is context-free, showing that option (d) is incorrect.

2. We say that a DFA M for a language A is *minimal* if there does not exist another DFA M' for A such that M' has strictly fewer states than M. Suppose that $M = (Q, \Sigma, \delta, q_0, F)$ is a minimal DFA for A. Using M, we construct a DFA $\overline{M}$ for the complement $\overline{A}$ as $\overline{M} = (Q, \Sigma, \delta, q_0, Q - F)$. Prove that $\overline{M}$ is a minimal DFA for $\overline{A}$.

   Answer:
   We prove this by contradiction. Suppose that $\overline{M}$ is not a minimal DFA for $\overline{A}$. Then there exists another DFA D for $\overline{A}$ such that D has strictly fewer states than $\overline{M}$. Now create another DFA D' by swapping the accepting and non-accepting states of D. Then D' recognizes the complement of $\overline{A}$. But the complement of $\overline{A}$ is just A, so D' recognizes A. Note that D' has the same number of states as D, and $\overline{M}$ has the same number of states as M. Thus, since we assumed that D has strictly fewer states than $\overline{M}$, then D' has strictly fewer states than M. But since D' recognizes A, this contradicts our assumption that M is a minimal DFA for A. Therefore, $\overline{M}$ is a minimal DFA for $\overline{A}$.

3. For alphabet $\Sigma = \{a, b, c\}$, the language

$$A = \{\, bc^n babc^n b \mid n \geq 0 \,\}$$

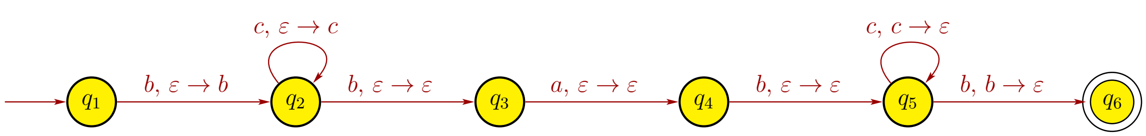
   is **context-free**. This is a slight variation of HW 5, problem 1(g) and HW 6, problem 1(h).

   (a) A CFG $G = (V, \Sigma, R, S)$ has a set of variables $V = \{S, T\}$, where S is the start variable; set of terminals $\Sigma = \{a, b, c\}$; and rules

$$\begin{aligned} S &\rightarrow bTb \\ T &\rightarrow cTc \mid bab \end{aligned}$$

   There are infinitely many other correct CFGs for L.

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



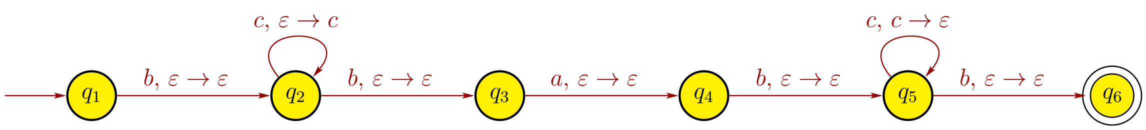
   where an edge label "$x, y \rightarrow z$" means read x, pop y, and push z.

   In the above PDA, we can make either or both of the following modifications, and still end up with a correct PDA for L.

   - Push b instead of ε in the transition from q_2 to q_3, and correspondingly pop b rather than ε in the transition from q_4 to q_5.
   - Add a new start state q_0 with an edge to q_1 with label "$\varepsilon, \varepsilon \rightarrow \$$", and also add an edge from q_6 (no longer an accept state) to a new accept state q_7 with label "$\varepsilon, \$ \rightarrow \varepsilon$". In this case, then the transition from q_1 to q_2 could instead push ε instead of b, and the transition from q_5 to q_6 could correspondingly

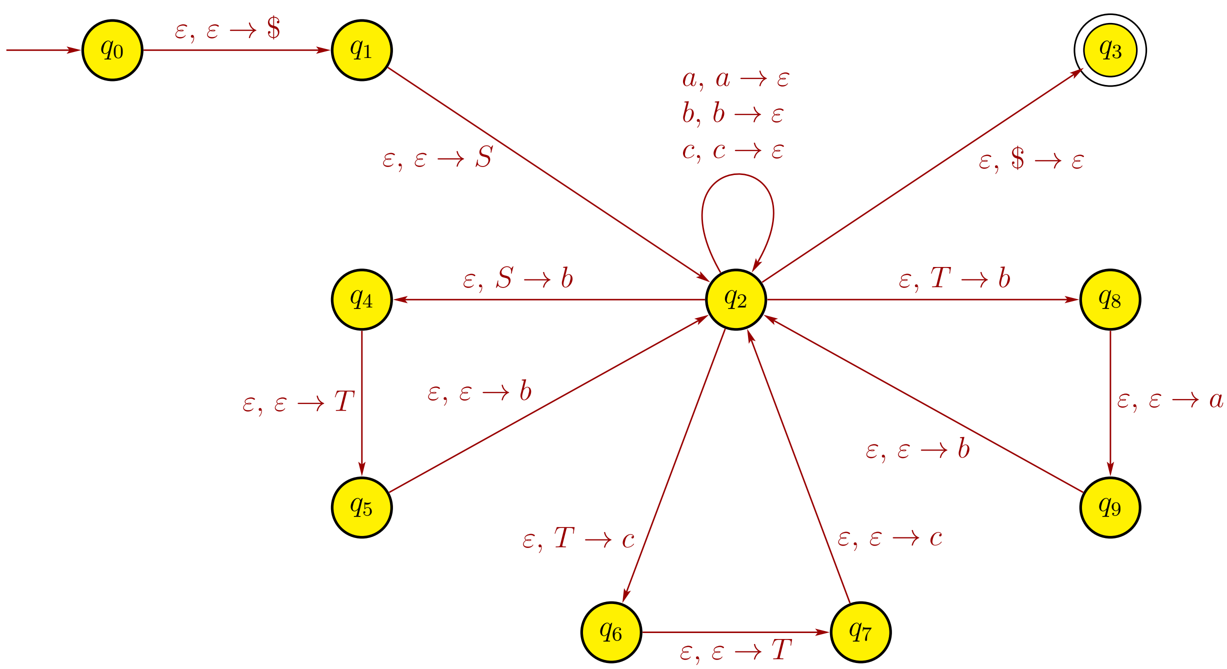
pop ε rather than b. This is because popping \$ in the transition from q_6 to q_7 makes sure the stack is empty so that the c's in the first group match the c's in the second group.

Suppose that we did not add the new states q_0 and q_7 described above but still modified the above PDA to push ε instead of b in the transition from q_1 to q_2, and correspondingly pop ε rather than b in the transition from q_5 to q_6. The resulting PDA



is **incorrect**. The problem with this modification is that by popping ε in the transition from q_5 to q_6, the PDA may move to q_6 (and accept) with the stack non-empty. Thus, there could be more c's in the first group than in the second group, leading to incorrectly some strings not in L. For example, the PDA with this modification would then accept $bccbabcb \notin L$.

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



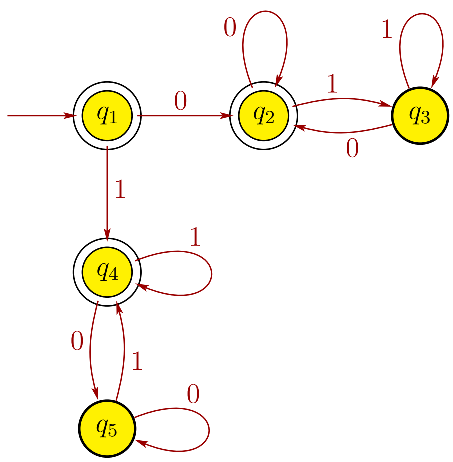
Note that

- The path $q_2 \to q_4 \to q_5 \to q_2$ corresponds to the rule $S \to bTb$, 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 $T \to cTc$, 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 $T \to bab$, where the symbols on the right side of the rule are pushed in reverse order.

4. Let $A = \{\, w \in \{0,1\}^* \mid n_{01}(w) = n_{10}(w) \,\}$, where $n_s(w)$ is the number of occurrences of the substring $s \in \{0,1\}^*$ in w. For example, the string $w_1 = 0001101100$ has $n_{01}(w_1) = 2$ and $n_{10}(w_1) = 2$, so $w_1 \in A$. Also, the string $w_2 = 00011011001$ has $n_{01}(w_2) = 3$ and $n_{10}(w_2) = 2$, so $w_2 \notin A$.

   The language A is regular. (HW 4, problem 3e considers essentially the same language.) A regular expression for the language is $0(0 \cup 11^*0)^* \cup 1(1 \cup 00^*1)^* \cup \varepsilon$. Another regular expression is $0(0 \cup 1)^*0 \cup 1(0 \cup 1)^*1 \cup 0 \cup 1 \cup \varepsilon$. A DFA for the language is



   There are infinitely many other correct regular expressions and DFAs for A.