CS 341-H02, Spring 2026, Face-to-Face 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 1.40. 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.

   1.2. 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.

   1.3. 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 (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.

1.4. 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.

1.5. 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 next prove is also nonregular by the pumping lemma. 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 x 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 any alphabet Σ, let $A = \Sigma^*$, so $\overline{A} = \emptyset$, which is finite, so it is regular (slide 1-95), showing that (b) is incorrect.
- For any language A over any alphabet Σ, we have that $\overline{A} = \Sigma^* - A$, and $\overline{A} \cap A = \emptyset$, which is finite, so it is regular (slide 1-95), showing that (c) is correct.

1.6. Answer: (c).

- By slide 2-99, A is non-context-free, so Theorem 2.20 shows that A cannot have a PDA, making (b) incorrect.
- Because A is non-context-free, A is also not regular (Corollary 2.32), so (a) is incorrect. We can see that the regular expression $(0 \cup 1)^*(0 \cup 1)^*$ in (a) is wrong because it generates the string $01 \not\in A$.
- For any strings w_1 and w_2, we have that $(w_1 w_2)^R = w_2^{\mathcal{R}} w_1^R$. Thus, for any string $ww \in A$, its reverse is $(ww)^{\mathcal{R}} = w^{\mathcal{R}} w^{\mathcal{R}} \in A$, so A is closed under reversal, making (c) correct.
- The strings 0101 and 11 are in A, but their concatenation $010111 \not\in A$, so A is not closed under concatenation, making (d) incorrect.

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), so 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.

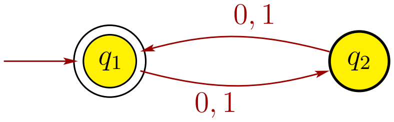
1.8. Answer: (b).
- The language A has regular expression 1^*0^*, so Kleene's Theorem (Theorem 1.54) implies that A is regular. Thus, (b) is correct, and (c) is incorrect. By Corollary 2.32, if a language is non-context-free, then it must also be non-regular, so (d) is incorrect.
- For each $i \geq 0$,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

1.9. Answer: (c).
- Consider the language $A^*(B \cup C)$ in part (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 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 (c) is correct.
- Consider the language $\overline{A}(B^* \cup C)$ in part (a). 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.
- For the language $\overline{A}(B^* \cup C)$ in part (b), 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 the language $A^*(B \cap C)$ in part (d), suppose that A, B, and C are all regular, so they are all context-free (Corollary 2.32). The class of regular languages is closed under Kleene star and intersection, implying that $A^*(B \cap C)$ is regular, so it is also context-free, showing that (d) is incorrect.

1.10. Answer: (e).

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 $R_2 = ((0 \cup 1)(0 \cup 1))^* \cup (0^*10^* \cup 0^*1)(0^*10^*1)^*$ cannot generate the string $w = 11100 \in A$, so R_2 is incorrect.
- We now show that $R_1 = (00 \cup 01 \cup 10 \cup 11)^* \cup 0^*1(0 \cup 10^*1)^*$ 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

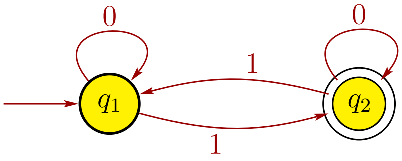
  

  While we can use the algorithm in part of the proof of Kleene's theorem (Lemma 1.60) to convert the DFA M_1 into a regular expression r_1 for A_1, the DFA is simple enough to be able to analyze it directly to obtain r_1. Specifically, note that every string accepted by M_1 has to be processed as follows:

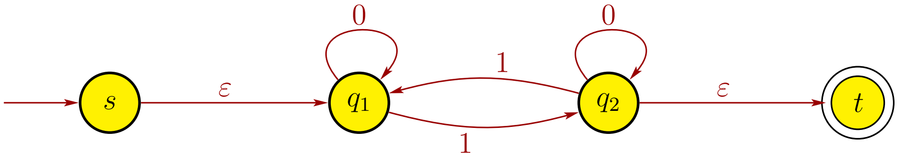
  - start in q_1,
  - loop from q_1 back to q_1 zero or more times.

  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_1.

- 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: (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

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

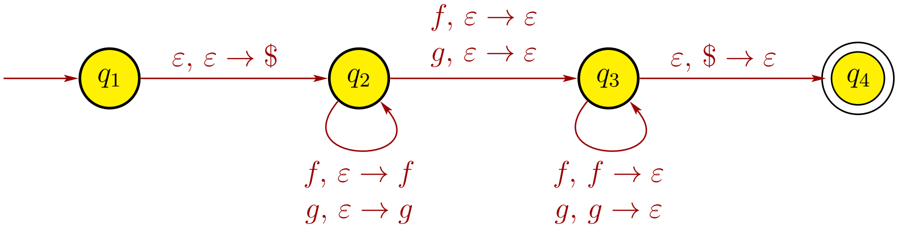
  Thus, A is a CFG, so (c) is correct, and (d) is incorrect.

1.14. Answer: (e).

- We know that $A = \{\, a^nb^n \mid n \geq 0 \,\}$ 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 $A = \{\, a^nb^nc^n \mid n \geq 0 \,\}$ 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.

1.15. Answer: (b).

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



  recognizes the language $L_2 = \{\, w \in \Sigma^* \mid w = w^{\mathcal{R}} \text{ and } |w| \text{ is odd} \,\}$, which is a slight variation of HW 6, problem 1(b).
- The string $\varepsilon \in L_1 = \{\, w \in \Sigma^* \mid w = w^{\mathcal{R}} \,\}$, but the given PDA rejects ε. Thus, $A \neq L_1$.
- The language L_3 is not context-free, which can be proven by slightly modifying the proof on slide 2-99, so L_3 cannot have a PDA (Theorem 2.20). Thus, $A \neq L_3$.

2. Short answers.

   (a) Let $\Sigma = \{e, f\}$ (**note the alphabet!**), and let A be the set of strings $w \in \Sigma^*$ such that $|w|$ is odd and w begins and ends in f, where $|w|$ denotes the length of w.

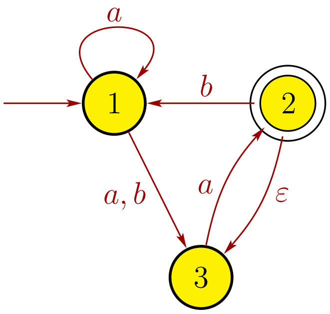
   Answer: There are infinitely many correct regular expressions for A, including
   $R_1 = f \ \cup \ f(e \cup f)((e \cup f)(e \cup f))^* f$,
   $R_2 = f \ \cup \ f((e \cup f)(e \cup f))^*(e \cup f)f$,
   $R_3 = f \ \cup \ f(e \cup f)f \ \cup \ f((e \cup f)(e \cup f))^*(e \cup f)f$.
   But $R_4 = f(e \cup f)((e \cup f)(e \cup f))^* f$ is incorrect because it cannot generate $f \in A$.

   (b) Suppose that language A_1 has CFG $G_1 = (V_1, \Sigma, R_1, S_1)$ and language A_2 has CFG $G_2 = (V_2, \Sigma, R_2, S_2)$. Give a CFG G_3 for $A_2 \circ A_1$ in terms of G_1 and G_2. You do not have to prove the correctness of your CFG G_3, but do not give just an example.

   Answer: $G_3 = (V_3, \Sigma, R_3, S_3)$, where

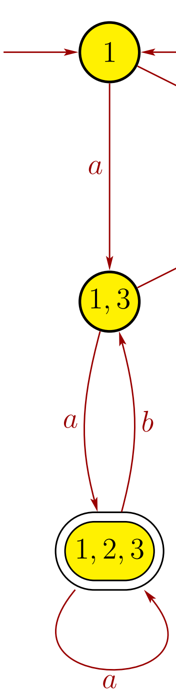
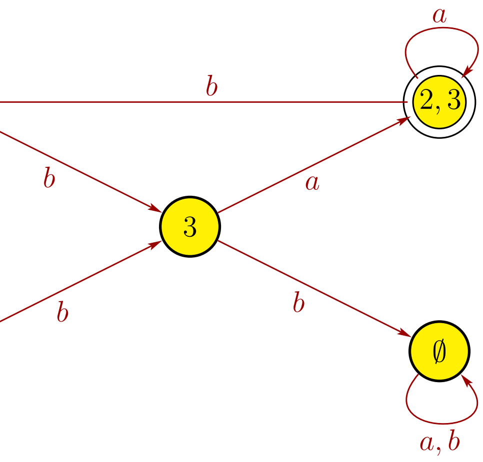
   - $V_3 = V_1 \cup V_2 \cup \{S_3\}$,
   - Σ is the same as in G_1 and G_2,
   - rules $R_3 = R_1 \cup R_2 \cup \{ \, S_3 \to S_2 S_1 \, \}$,
   - S_3 is the start variable, where $S_3 \notin V_1 \cup V_2$.

3. Let N be the following NFA with $\Sigma = \{a, b\}$, and let $C = L(N)$.



Give a DFA for C. You only need to draw the state diagram (graph); do not give the 5-tuple.

Answer: A DFA for C is below (there are other correct DFAs):

4. For $\Sigma = \{a, b, c\}$, let
$$L = \{\, c^{2n} b^k a^{3n} \mid n \geq 0, k \geq 2 \,\}.$$

Answer: This is similar to HW 6, problem 4(c).

(a) One CFG $G_1 = (V, \Sigma, R, S)$ for language L has a set of variables $V = \{S, T\}$, where S is the start variable; set of terminals $\Sigma = \{a, b, c\}$; and rules

$$\begin{array}{rcl} S & \rightarrow & ccSaaa \mid T \\ T & \rightarrow & bT \mid bb \end{array}$$

There are infinitely many other correct CFGs for L. Another correct CFG $G_2 = (V, \Sigma, R, S)$ for L has variables $V = \{S, T, U\}$, $\Sigma = \{a, b, c\}$, and rules

$$\begin{array}{rcl} S & \rightarrow & ccSaaa \mid T \\ T & \rightarrow & bbU \\ U & \rightarrow & bU \mid \varepsilon \end{array}$$

An **incorrect** CFG has rules

$$\begin{array}{rcl} S & \rightarrow & ccSaaa \mid T \\ T & \rightarrow & bbT \mid bT \mid \varepsilon \end{array}$$

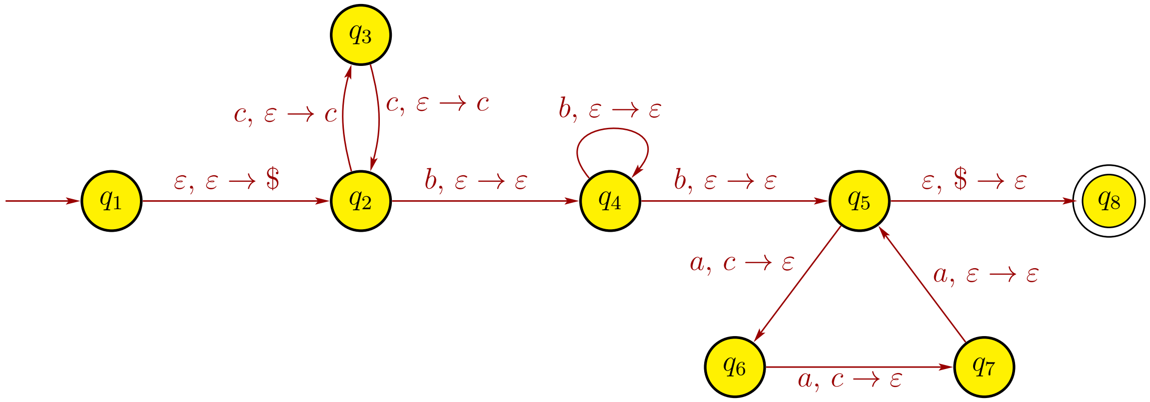
The derivation $S \Rightarrow ccSaaa \Rightarrow ccTaaa \Rightarrow ccaaa$ shows that the rules generate the string $ccaaa$, which is not in L because it does not have at least 2 b's.

Another **incorrect** CFG has rules

$$\begin{aligned} S &\rightarrow ccSaaa \mid T \\ T &\rightarrow bbT \mid \varepsilon \end{aligned}$$

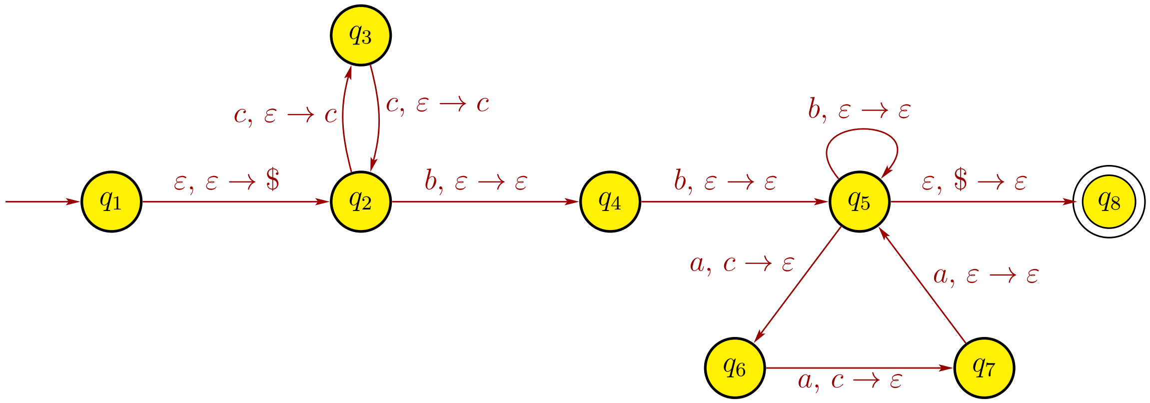
The derivation $S \Rightarrow ccSaaa \Rightarrow ccTaaa \Rightarrow ccaaa$ shows that the rules generate the string *ccaaa*, which is not in L because it does not have at least 2 b's. Also, the rules cannot generate $ccbbbaaa \in L$.

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



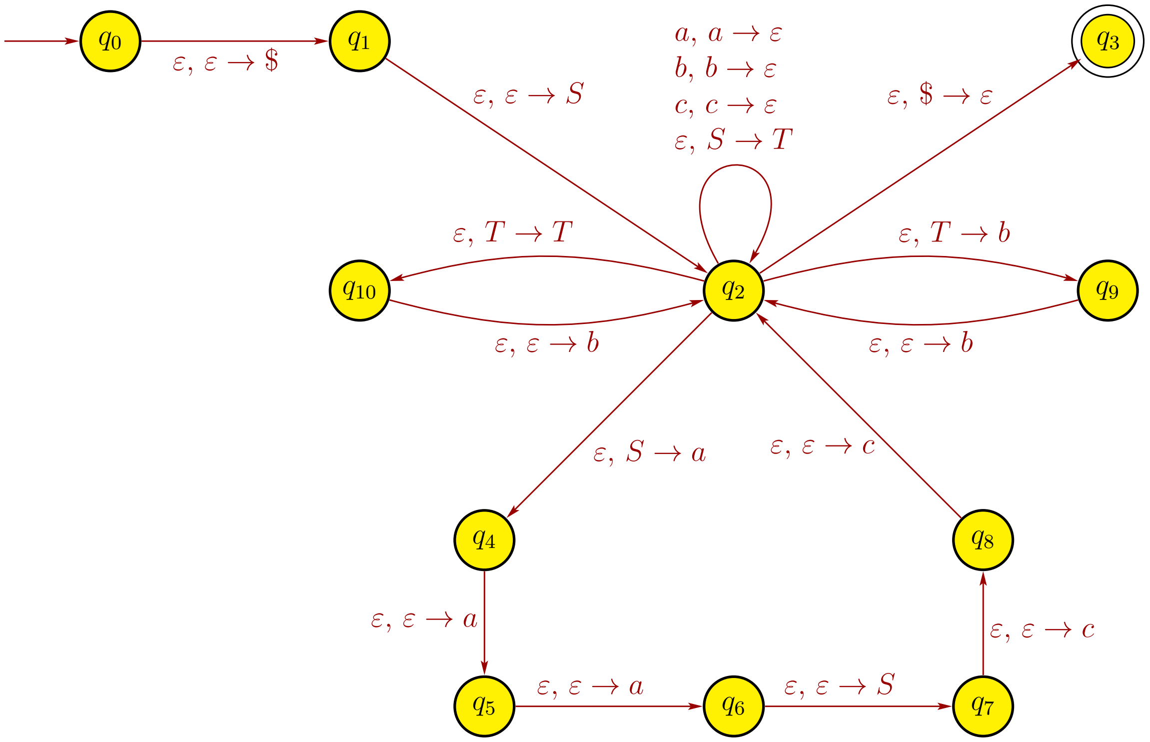
where an edge label "$x, y \rightarrow z$" means read x, pop y, and push z. Note that on the cycle $q_5 \rightarrow q_6 \rightarrow q_7 \rightarrow q_5$, the first two transitions pop c, but the third transition pops ε. This is because the cycle $q_2 \rightarrow q_3 \rightarrow q_2$ pushes only 2 c's, but an accepted string has to have 3 a's for every 2 c's.

An **incorrect** PDA is



which incorrectly accepts $ccbbaaab \notin L$.

We can also design a correct PDA for L by applying the algorithm from Lemma 2.21 to convert the second correct CFG G_1 in part (a) into a PDA.



Note that

- The path $q_2 \to q_4 \to q_5 \to q_6 \to q_7 \to q_8 \to q_2$ corresponds to the rule $S \to bbSaaa$, where the symbols on the right side of the rule are pushed in reverse order.
- The path $q_2 \to q_9 \to q_2$ corresponds to the rule $T \to bb$, where the symbols on the right side of the rule are pushed in reverse order.
- The path $q_2 \to q_{10} \to q_2$ corresponds to the rule $T \to bb$, where the symbols on the right side of the rule are pushed in reverse order.

5. For $\Sigma = \{e, f\}$ (**note the alphabet!**), let $A = \{\, w \in \Sigma^* \mid w \text{ has more } e\text{'s than } f\text{'s} \,\}$.

   Is A a regular or nonregular language? If A is regular, give a regular expression **and** DFA (only state diagram) for A. If A is not regular, prove that it is a nonregular language.

   Answer: The language A is **nonregular**. Note that A is from problem 3d from HW 4 but with a different Σ. For a generic symbol $c \in \Sigma$ and string $w \in \Sigma^*$, let $n_c(w)$ be the number of c's in string w.

   To prove that A is nonregular, suppose that A is a regular language. Let p be the "pumping length" of the pumping lemma (Theorem 1.70). Consider the string $s = f^p e^{p+1}$. Note that $s \in A$ since $n_e(s) = p + 1 > p = n_f(s)$, and $|s| = 2p + 1 \geq p$, so the pumping lemma will hold. Thus, we can split the string s into 3 parts $s = xyz$ satisfying the properties

(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 f's, the third property implies that x and y consist only of f's. So z will be the rest of the f's, followed by e^{p+1}. The second property states that $|y| > 0$, so y has at least one f. More precisely, we can then say that

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

Since $f^p e^{p+1} = s = xyz = f^j f^k f^m e^{p+1} = f^{j+k+m} e^{p+1}$, we must have that $j+k+m = p$. The first property implies that $xy^2z \in A$, where

$$\begin{aligned} xy^2z &= f^j f^k f^k f^m e^{p+1} \\ &= f^{p+k} e^{p+1} \end{aligned}$$

since $j+k+m = p$. But $xy^2z \notin A$ because $n_e(xy^2z) = p+1 \not> p+k = n_f(xy^2z)$ since $k \geq 1$, so we get a contradiction. Therefore, A is a nonregular language.