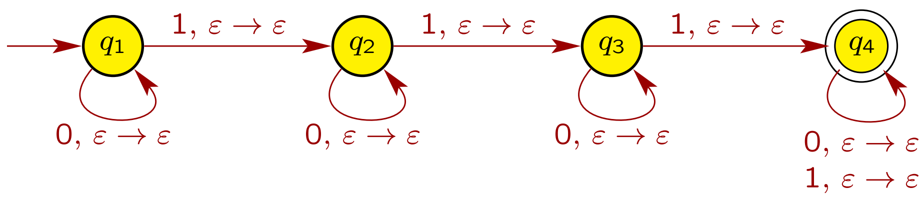
Homework 6 Solutions

1. Give pushdown automata that recognize the following languages. For each PDA, give both a state diagram and 6-tuple specification. Every context-free language has infinitely many (correct) PDAs, but you only need to give one.

   (a) $A = \{\, w \in \{0,1\}^* \mid w \text{ contains at least three 1s} \,\}$

   Answer:



We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1\}$
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

Input:	0			1			ε		
Stack:	0	1	ε	0	1	ε	0	1	ε
q_1			$\{(q_1,\varepsilon)\}$			$\{(q_2,\varepsilon)\}$			
q_2			$\{(q_2,\varepsilon)\}$			$\{(q_3,\varepsilon)\}$			
q_3			$\{(q_3,\varepsilon)\}$			$\{(q_4,\varepsilon)\}$			
q_4			$\{(q_4,\varepsilon)\}$			$\{(q_4,\varepsilon)\}$			

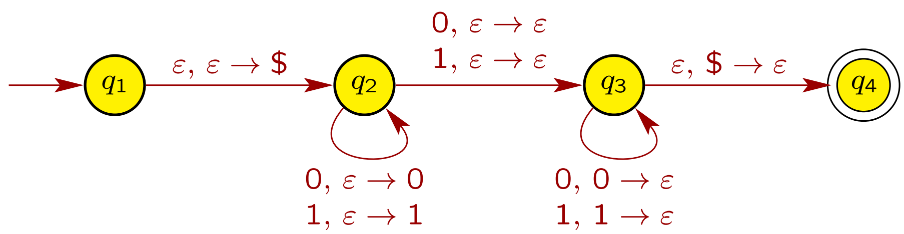
Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_4\}$

Note that A is a regular language, so the language has a DFA. We can easily convert the DFA into a PDA by using the same states and transitions and never push nor pop anything to/from the stack.

(b) $B = \{\, w \in \{0, 1\}^* \mid w = w^{\mathcal{R}} \text{ and the length of } w \text{ is odd} \,\}$

Answer:



Since the length of any string $w \in B$ is odd, w must have a symbol exactly in the middle position; i.e., $|w| = 2n + 1$ for some $n \geq 0$, and the $(n + 1)$th symbol in w is the middle one. If a string w of length $2n + 1$ satisfies $w = w^{\mathcal{R}}$, the first n symbols must match (in reverse order) the last n symbols, and the middle symbol doesn't have to match anything. Thus, in the above PDA, the transition from q_2 to itself reads the first n symbols and pushes these on the stack. The transition from q_2 to q_3 nondeterministically identifies the middle symbol of w, which doesn't need to match any symbol, so the stack is unaltered. The transition from q_3 to itself then reads the last n symbols of w, popping the stack at each step to make sure the symbols after the middle match (in reverse order) the symbols before the middle.

We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, \$\}$ (use $\$$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

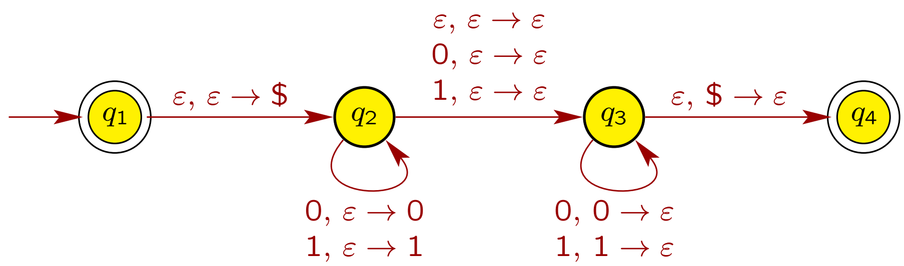
Input:	0				1				ε			
Stack:	0	1	$\$$	ε	0	1	$\$$	ε	0	1	$\$$	ε
q_1												$\{(q_2, \$)\}$
q_2				$\{(q_2, 0), (q_3, \varepsilon)\}$				$\{(q_2, 1), (q_3, \varepsilon)\}$				
q_3	$\{(q_3, \varepsilon)\}$					$\{(q_3, \varepsilon)\}$					$\{(q_4, \varepsilon)\}$	
q_4												

Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_4\}$

(c) $C = \{\, w \in \{0, 1\}^* \mid w = w^{\mathcal{R}} \,\}$

Answer:



The length of a string $w \in C$ can be either even or odd. If it's even, then there is no middle symbol in w, so the first half of w is pushed on the stack, we move from q_2 to q_3 without reading, pushing, or popping anything, and then match the second half of w to the first half in reverse order by popping the stack. If the length of w is odd, then there is a middle symbol in w, and the description of the PDA in part (b) applies.

We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3, q_4\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{0, 1, \$\}$ (use $\$$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

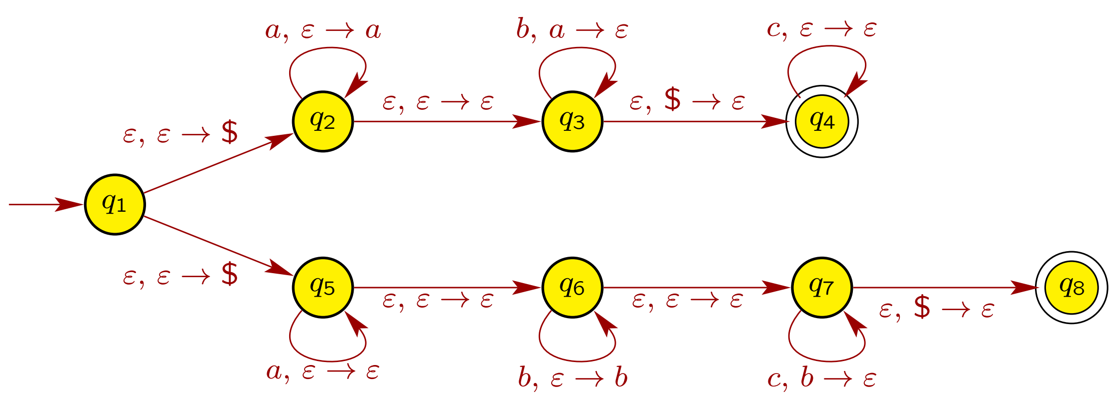
Input:	0				1				ε			
Stack:	0	1	$	ε	0	1	$	ε	0	1	$	ε
q_1												$\{(q_2, \$)\}$
q_2				$\{(q_2, 0), (q_3, \varepsilon)\}$				$\{(q_2, 1), (q_3, \varepsilon)\}$				$\{(q_3, \varepsilon)\}$
q_3	$\{(q_3, \varepsilon)\}$					$\{(q_3, \varepsilon)\}$					$\{(q_4, \varepsilon)\}$	
q_4												

Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_1, q_4\}$

(d) $D = \{\, a^i b^j c^k \mid i, j, k \geq 0, \text{ and } i = j \text{ or } j = k \,\}$

Answer:



The PDA has a nondeterministic branch at q_1. If the string is $a^i b^j c^k$ with $i = j$, then the PDA takes the branch from q_1 to q_2. If the string is $a^i b^j c^k$ with $j = k$, then the PDA takes the branch from q_1 to q_5.

We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, \ldots, q_8\}$
- $\Sigma = \{a, b, c\}$
- $\Gamma = \{a, b, \$\}$ (use \$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

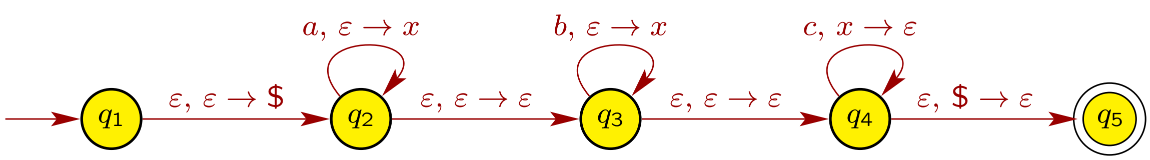
Input:	a					b					c					ε				
Stack:	a	b	c	\$	ε	a	b	c	\$	ε	a	b	c	\$	ε	a	b	c	\$	ε
q_1																				$\{(q_2, \$), (q_5, \$)\}$
q_2					$\{(q_2, a)\}$															$\{(q_3, \varepsilon)\}$
q_3						$\{(q_3, \varepsilon)\}$													$\{(q_4, \varepsilon)\}$	
q_4															$\{(q_4, \varepsilon)\}$					
q_5					$\{(q_5, \varepsilon)\}$															$\{(q_6, \varepsilon)\}$
q_6										$\{(q_6, b)\}$										$\{(q_7, \varepsilon)\}$
q_7												$\{(q_7, \varepsilon)\}$							$\{(q_8, \varepsilon)\}$	
q_8																				

Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_4, q_8\}$

(e) $E = \{\, a^i\, b^j\, c^k \mid i, j, k \geq 0 \text{ and } i + j = k \,\}$

Answer:



For every a and b read in the first part of the string, the PDA pushes an x onto the stack. Then it must read a c for each x popped off the stack.

We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, \ldots, q_5\}$
- $\Sigma = \{a, b, c\}$
- $\Gamma = \{x, \$\}$ (use \$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

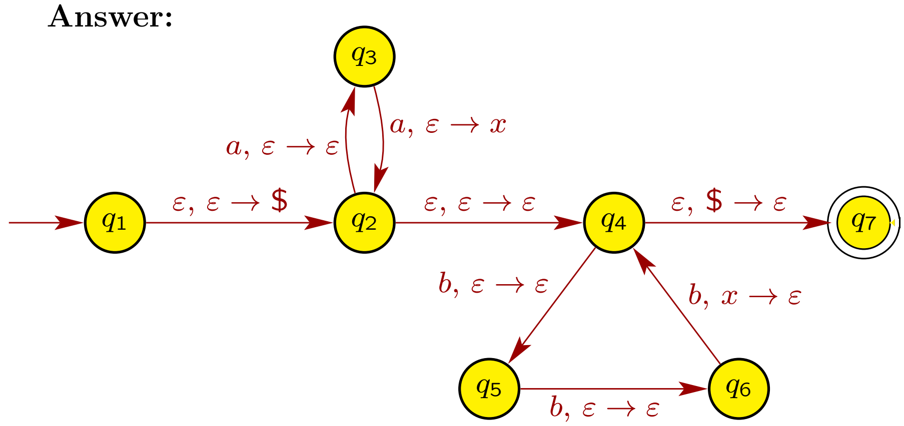
Input:	a			b			c			ε		
Stack:	x	\$	ε	x	\$	ε	x	\$	ε	x	\$	ε
q_1												$\{(q_2, \$)\}$
q_2			$\{(q_2, x)\}$									$\{(q_3, \varepsilon)\}$
q_3						$\{(q_3, x)\}$						$\{(q_4, \varepsilon)\}$
q_4							$\{(q_4, \varepsilon)\}$				$\{(q_5, \varepsilon)\}$	
q_5												

Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_5\}$

(f) $F = \{\, a^{2n} b^{3n} \mid n \geq 0 \,\}$

Answer:



The PDA pushes a single x onto the stack for every 2 a's read at the beginning of the string. Then it pops a single x for every 3 b's read at the end of the string.

We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, \ldots, q_7\}$
- $\Sigma = \{a, b\}$
- $\Gamma = \{x, \$\}$ (use $\$$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

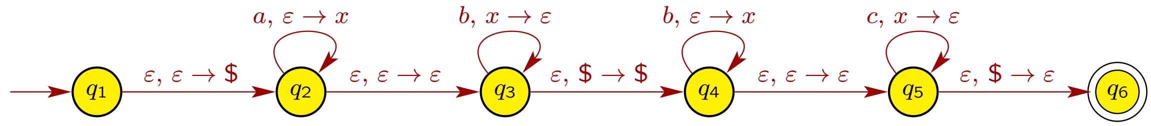
Input:	a			b			ε		
Stack:	x	$\$$	ε	x	$\$$	ε	x	$\$$	ε
q_1									$\{\,(q_2, \$)\,\}$
q_2			$\{\,(q_3, \varepsilon)\,\}$						$\{\,(q_4, \varepsilon)\,\}$
q_3			$\{\,(q_2, x)\,\}$						
q_4						$\{\,(q_5, \varepsilon)\,\}$		$\{\,(q_7, \varepsilon)\,\}$	
q_5						$\{\,(q_6, \varepsilon)\,\}$			
q_6				$\{\,(q_4, \varepsilon)\,\}$					
q_7									

Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_7\}$

(g) $H = \{ a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i + k = j \}$

Answer: A PDA M for H is as follows:



We formally express the PDA for H as a 6-tuple $M = (Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, \ldots, q_6\}$
- $\Sigma = \{a, b, c\}$
- $\Gamma = \{x, \$\}$ (use $\$$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

Input:	a			b			c			ε		
Stack:	x	$\$$	ε	x	$\$$	ε	x	$\$$	ε	x	$\$$	ε
q_1												$\{ (q_2, \$) \}$
q_2			$\{ (q_2, x) \}$									$\{ (q_3, \varepsilon) \}$
q_3				$\{ (q_3, \varepsilon) \}$							$\{ (q_4, \$) \}$	
q_4						$\{ (q_4, x) \}$						$\{ (q_5, \varepsilon) \}$
q_5							$\{ (q_5, \varepsilon) \}$				$\{ (q_6, \varepsilon) \}$	
q_6												

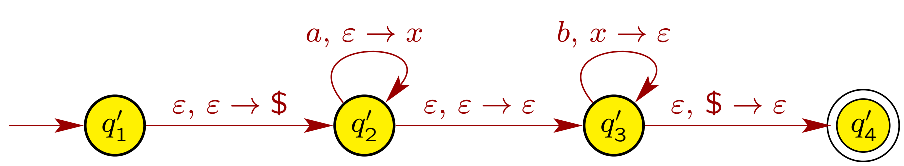
Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_6\}$

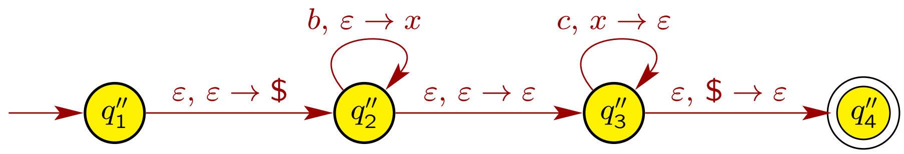
To explain the PDA M for $H = \{ a^i b^j c^k \mid i, j, k \geq 0 \text{ and } i + k = j \}$, note that $H = L_1 \circ L_2$ for

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

because concatenating any string $a^i b^i \in L_1$ with any string $b^k c^k \in L_2$ results in a string $a^i b^i b^k c^k = a^i b^{i+k} c^k \in H$. Thus, for a string $a^i b^j c^k \in H$, the number i of a's at the beginning has to be no more than the number j of b's in the middle (because $i + k = j$ implies $i \leq j$), and the remaining number $j - i$ of b's in the middle must match the number k of c's at the end. Hence, if we have PDAs M_1 and M_2 for L_1 and L_2, respectively, then we can then build a PDA for H by connecting M_1 and M_2 so that M_1 processes the first part of the string $a^i b^i$, and M_2 processes the second part of the string $b^k c^k$. A PDA M_1 for L_1 is



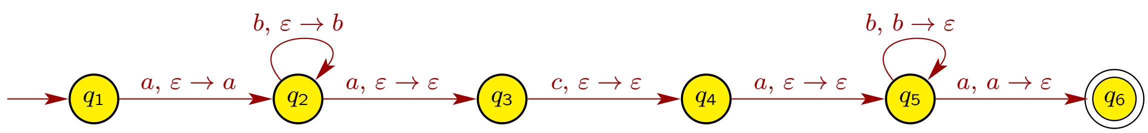
(Another PDA for L_1 is given slide 2-38 of the notes.) Similarly, a PDA M_2 for L_2 is



But in connecting the two PDAs M_1 and M_2 to get a PDA M for H, we need to make sure the stack is empty after M_1 finishes processing the first part of the string and before M_2 starts processing the second part of the string. This is accomplished in the PDA M for H by the transition from q_3 to q_4 with label "$\varepsilon, \$ \to \$$".

(h) $L = \{ ab^n acab^n a \mid n \geq 0 \}$.

Answer: A PDA M for L is as follows:



We formally express the PDA for L as a 6-tuple $M = (Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, \ldots, q_6\}$
- $\Sigma = \{a, b, c\}$
- $\Gamma = \{a, b\}$ (use a to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

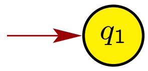
Input:	a			b			c			ε		
Stack:	a	b	ε	a	b	ε	a	b	ε	a	b	ε
q_1			$\{(q_2, a)\}$									
q_2			$\{(q_3, \varepsilon)\}$			$\{(q_2, b)\}$						
q_3									$\{(q_4, \varepsilon)\}$			
q_4			$\{(q_5, \varepsilon)\}$									
q_5	$\{(q_6, \varepsilon)\}$				$\{(q_5, \varepsilon)\}$							
q_6												

Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_6\}$

(i) $\emptyset$, with $\Sigma = \{0, 1\}$

Answer:



Because the PDA has no accept states, the PDA accepts no strings; i.e., the PDA recognizes the language $\emptyset$.

We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1\}$
- $\Sigma = \{0, 1\}$
- $\Gamma = \{x\}$
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

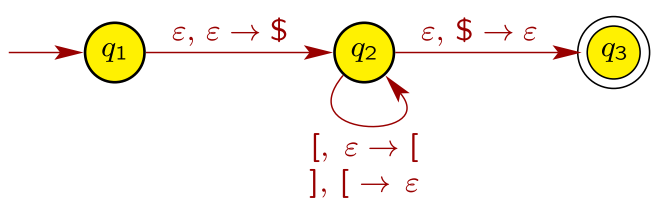
Input:	0		1		ε	
Stack:	x	ε	x	ε	x	ε
q_1						

  Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \emptyset$

(j) The language J of strings of properly balanced left and right brackets: every left bracket can be paired with a unique subsequent right bracket, and every right bracket can be paired with a unique preceding left bracket. Moreover, the string between any such pair has the same property. For example, $[\,]\,[\,[\,[\,]\,[\,]\,]\,[\,]\,] \in J$.

Answer:



We formally express the PDA as a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, q_3\}$
- $\Sigma = \{\,[,\,]\,\}$
- $\Gamma = \{\,[,\ \$\,\}$ (use $\$$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

Input:	[			]			ε		
Stack:	[	$\$$	ε	[	$\$$	ε	[	$\$$	ε
q_1									$\{\,(q_2, \$)\,\}$
q_2			$\{\,(q_2, [)\,\}$	$\{\,(q_2, \varepsilon)\,\}$				$\{\,(q_3, \varepsilon)\,\}$	
q_3									

Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_3\}$

2. (a) Example 2.36 of the Sipser's book (also see slides 2-96 and 2-97 of the notes) proves that the language $\{ a^n b^n c^n \mid n \geq 0 \}$ is not context-free. Use this and the languages

$$\begin{aligned} A &= \{ a^m b^n c^n \mid m, n \geq 0 \} \text{ and} \\ B &= \{ a^n b^n c^m \mid m, n \geq 0 \} \end{aligned}$$

to show that the class of context-free languages is not closed under intersection.

Answer: The language A is context free since it has CFG G_1 with rules

$$\begin{aligned} S &\to XY \\ X &\to aX \mid \varepsilon \\ Y &\to bYc \mid \varepsilon \end{aligned}$$

The language B is context free since it has CFG G_2 with rules

$$\begin{aligned} S &\to XY \\ X &\to aXb \mid \varepsilon \\ Y &\to cY \mid \varepsilon \end{aligned}$$

But $A \cap B = \{ a^n b^n c^n \mid n \geq 0\}$, which we know is not context free from Example 2.36 of the textbook. Thus, the class of context-free languages is not closed under intersection.

(b) Recall DeMorgan's law (Theorem 0.20 of the textbook): for any two sets A and B, $\overline{A \cup B} = \overline{A} \cap \overline{B}$. Use part (a) and DeMorgan's law to show that the class of context-free languages is not closed under complementation.

Answer: We will use a proof by contradiction, so we first assume the opposite of what we want to show; i.e., suppose the following is true:

R1. The class of context-free languages is closed under complementation.

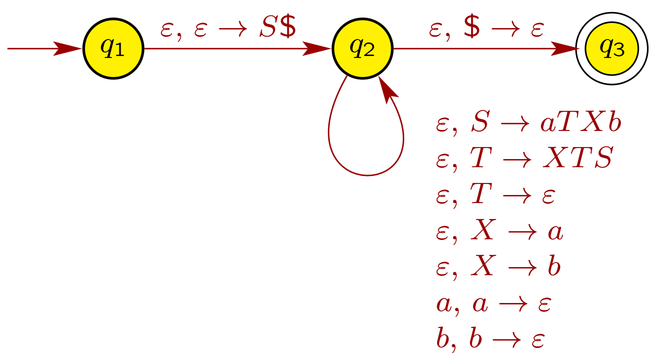
Define the context-free languages A and B as in the previous part. Then R1 implies $\overline{A}$ and $\overline{B}$ are context-free. We know the class of context-free languages is closed under union, as shown on slide 2-101, so we then must have that $\overline{A} \cup \overline{B}$ is context-free. Then again apply R1 to conclude that $\overline{\overline{A} \cup \overline{B}}$ is context-free. Now DeMorgan's law states that $A \cap B = \overline{\overline{A} \cup \overline{B}}$, but we showed in the previous part that $A \cap B$ is not context-free, which is a contradiction. Therefore, R1 must not be true.

3. (Optional) Consider the following CFG $G = (V, \Sigma, R, S)$, where $V = \{S, T, X\}$, $\Sigma = \{a, b\}$, the start variable is S, and the rules R are

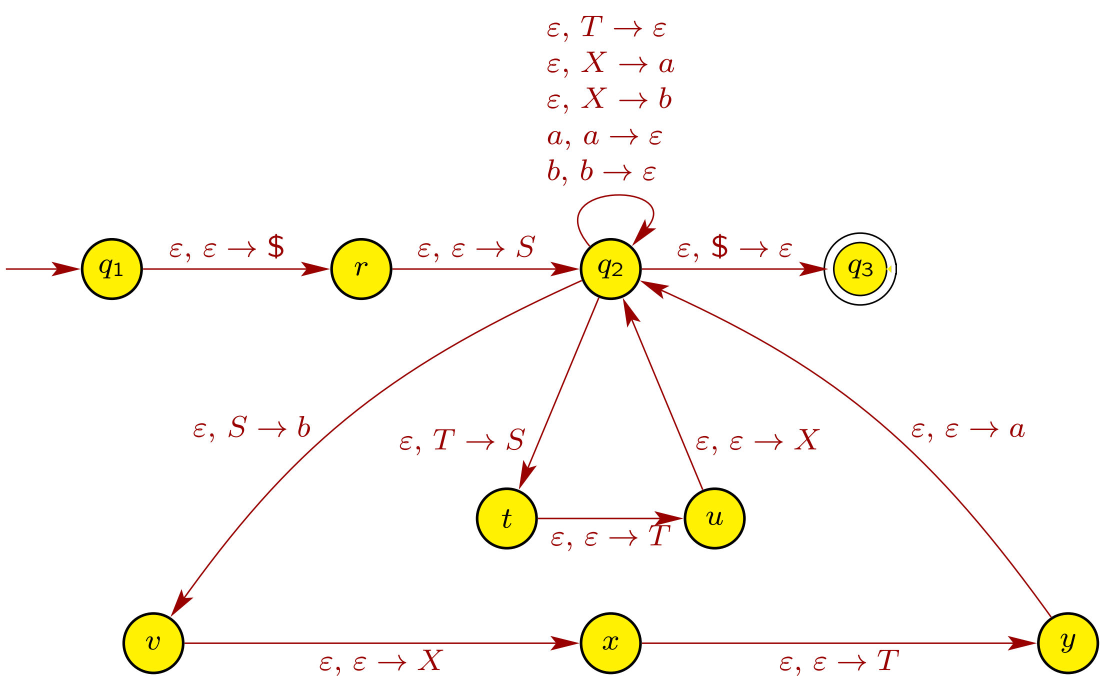
$$\begin{aligned} S &\rightarrow aTXb \\ T &\rightarrow XTS \mid \varepsilon \\ X &\rightarrow a \mid b \end{aligned}$$

Convert G to an equivalent PDA using the procedure given in Lemma 2.21.

Answer: First we create a PDA for G that allows for pushing strings onto the stack:



Then we need to fix the non-compliant transitions, i.e., the ones for which a string of length more than 1 is pushed onto the stack. The only non-compliant transitions are the first two from q_2 back to itself, and the transition from q_1 to q_2. Fixing these gives the following PDA:



4. Each of the languages in the parts below is of one of following three types:

Type REG.	It is regular.
Type CFL.	It is context-free, but not regular.
Type NCFL.	It is not context-free.

For each of the following languages, specify which type it is. Also follow these instructions:

- If a language L is of Type REG, give a regular expression **and** a DFA for L. For your DFA, give both a state diagram and 5-tuple description.
- If a language L is of Type CFL, give a context-free grammar **and** a PDA for L. Be sure to give a 4-tuple description for your CFG. For your PDA, give both a state diagram and 6-tuple description. **Also, prove that L is not regular.**
- If a language L is of Type NCFL, prove that L is not context-free.

(a) $A = \{ \mathtt{0}^{2i}\, \mathtt{1}^{3j}\, \mathtt{0}^{k} \mid i, j, k \geq 0, \text{and } i = j = k \}$.

Answer: Language A is of type NCFL. To prove this, assume for a contradiction that A is a CFL, and note that we can write A as $A = \{ \mathtt{0}^{2n}\, \mathtt{1}^{3n}\, \mathtt{0}^{n} \mid n \geq 0 \}$. Let p be the pumping length of the pumping lemma for CFLs (Theorem 2.34), and consider string $s = \mathtt{0}^{2p}\mathtt{1}^{3p}\mathtt{0}^{p} \in A$. Note that $|s| = 6p > p$, so the pumping lemma will hold. Thus, there exist strings u, v, x, y, z such that $s = uvxyz = \mathtt{0}^{2p}\mathtt{1}^{3p}\mathtt{0}^{p}$, $uv^ixy^iz \in A$ for all $i \geq 0$, and $|vy| \geq 1$. We now consider all of the possible choices for v and y:

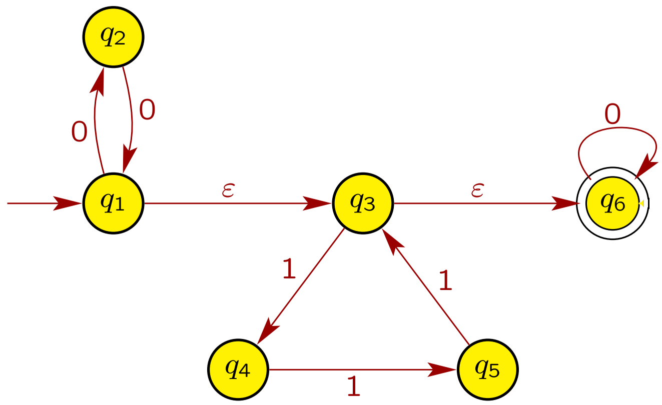
- Suppose strings v and y are uniform (e.g., $v = \mathtt{0}^j$ for some $j \geq 0$, and $y = \mathtt{1}^k$ for some $k \geq 0$). Then $|vy| \geq 1$ implies that $j \geq 1$ or $k \geq 1$ (or both), so uv^2xy^2z won't have the correct number of 0's at the beginning, 1's in the middle, and 0's at the end since only at most two groups of symbols are changed, not all three groups. Hence, $uv^2xy^2z \notin A$.
- Now suppose strings v and y are not both uniform. Then uv^2xy^2z will not have the form $\mathtt{0}\cdots\mathtt{0}\mathtt{1}\cdots\mathtt{1}\mathtt{0}\cdots\mathtt{0}$. Hence, $uv^2xy^2z \notin A$.

Thus, there are no options for v and y such that $uv^ixy^iz \in A$ for all $i \geq 0$. This is a contradiction, so A is not a CFL.

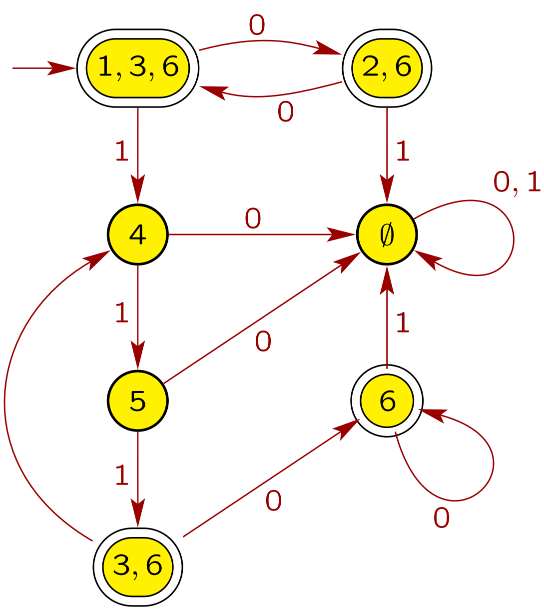
(b) $B = \{ \mathtt{0}^{2i}\, \mathtt{1}^{3j}\, \mathtt{0}^{k} \mid i, j, k \geq 0 \}$.

Answer: Language B is of type REG. This is because there is no relationship among i, j, and k in $\mathtt{0}^{2i}\mathtt{1}^{3j}\mathtt{0}^{k}$ in the specification of B. A regular expression for B is $(\mathtt{00})^*(\mathtt{111})^*(\mathtt{0})^*$; there are infinitely many other correct regular expressions for B.

Although coming up directly with a DFA for B is a bit tricky, designing an NFA N for B is easier. We will then convert N into a DFA D for B. An NFA N for B is as follows:



We now convert N into a DFA D using the algorithm in the proof of Theorem 1.39 to get the state diagram



The 5-tuple description of D is $D = (Q, \Sigma, \delta, q_{1,3,6}, F)$, with

- $Q = \{q_{1,3,6}, q_{2,6}, q_4, q_\emptyset, q_5, q_6, q_{3,6}\}$,
- $\Sigma = \{0, 1\}$,
- start state $q_{1,3,6}$,
- set of accepting states $F = \{q_{1,3,6}, q_{2,6}, q_6, q_{3,6}\}$,

- transition function δ:

	0	1
$q_{1,3,6}$	$q_{2,6}$	q_4
$q_{2,6}$	$q_{1,3,6}$	$q_\emptyset$
q_4	$q_\emptyset$	q_5
$q_\emptyset$	$q_\emptyset$	$q_\emptyset$
q_5	$q_\emptyset$	$q_{3,6}$
q_6	q_6	$q_\emptyset$
$q_{3,6}$	q_6	q_4

There are infinitely many other correct DFAs for B.

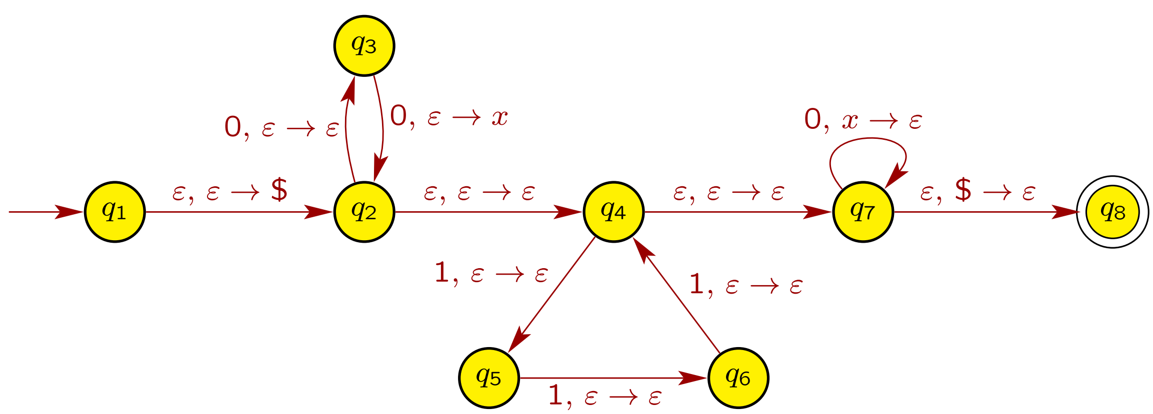
(c) $C = \{ 0^{2i}\, 1^{3j}\, 0^k \mid i, j, k \geq 0, \text{and } i = k \}$.

Answer: Language C is of type CFL. Strings in C have twice as many 0s at the beginning of the string as at the end of the string. The 1s in the middle do not have to match anything, but the number of them must be a multiple of 3.

A CFG for C is $G = (V, \Sigma, R, S)$ with $V = \{ S, X \}$ with S the start variable, $\Sigma = \{ 0, 1 \}$, and rules in R as

$$
\begin{aligned}
S &\rightarrow 00S0 \mid X \\
X &\rightarrow 111X \mid \varepsilon
\end{aligned}
$$

There are infinitely many other correct CFGs for C. A PDA for C is



Strings in C have twice as many 0s at the beginning of the string as at the end of the string. The PDA handles this by pushing only a single x from reading two 0s in q_2 and q_3, and pops a single x for every 0 read in q_7. Because the number of 1s in the middle doesn't have to match anything, the transitions among q_4, q_5, and q_6 do not pop/push anything from/to the stack. There are infinitely many other correct PDAs for C. The 6-tuple description of the PDA is $(Q, \Sigma, \Gamma, \delta, q_1, F)$, where

- $Q = \{q_1, q_2, \ldots, q_8\}$
- $\Sigma = \{0, 1\}$

- $\Gamma = \{x, \$\}$ (use $\$$ to mark bottom of stack)
- transition function $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$ is defined by

Input:	0			1			ε		
Stack:	x	$\$$	ε	x	$\$$	ε	x	$\$$	ε
q_1									$\{ (q_2, \$) \}$
q_2			$\{ (q_3, \varepsilon) \}$						$\{ (q_4, \varepsilon) \}$
q_3			$\{ (q_2, x) \}$						
q_4						$\{ (q_5, \varepsilon) \}$			$\{ (q_7, \varepsilon) \}$
q_5						$\{ (q_6, \varepsilon) \}$			
q_6						$\{ (q_4, \varepsilon) \}$			
q_7	$\{ (q_7, \varepsilon) \}$							$\{ (q_8, \varepsilon) \}$	
q_8									

Blank entries are $\emptyset$.

- q_1 is the start state
- $F = \{q_8\}$

There are infinitely many other correct PDAs for C.

We now prove that C is nonregular. For a contradiction, suppose that C is regular, and let p be the pumping length in the pumping lemma for regular languages (Theorem 1.70). Consider the string $s = 0^{2p}1110^p \in C$, where $|s| = 3p+3 \geq p$, so the conclusions of the pumping lemma for regular languages will hold: we can split $s = xyz$ with (i) $xy^iz \in C$ for all $i \geq 0$, (ii) $|y| > 0$, and (iii) $|xy| \leq p$. Property (iii) implies that x and y are solely contained in the first p of the 0s in s, and z is the rest of the first group of 0s followed by 1110^p. Specifically, we can write

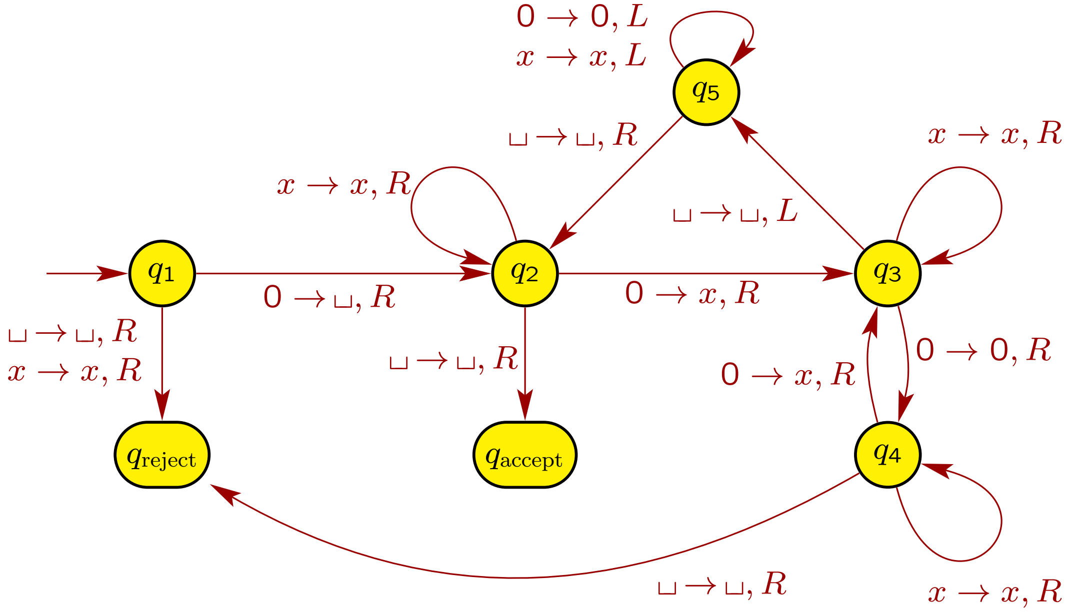
$$
\begin{aligned}
x &= 0^j \text{ for some } j \geq 0, \\
y &= 0^k \text{ for some } k \geq 0, \\
z &= 0^l 1110^p \text{ for some } l \geq 0, .
\end{aligned}
$$

Because $0^{2p}1110^p = s = xyz = 0^j0^k0^l1110^p = 0^{j+k+l}1110^p$, we get that $j+k+l = 2p$. Also, property (ii) implies that $k \geq 1$. By property (i), we must have that $xyyz \in C$, but

$$xyyz = 0^j0^k0^k0^l1110^p = 0^{2p+k}1110^p \notin C$$

because $j+k+l = 2p$ and $k \geq 1$. Since $xyyz \notin C$, this contradicts the pumping lemmma for regular languages, so C is not regular.

5. The Turing machine M below recognizes the language $A = \{ 0^{2^n} \mid n \geq 0 \}$.



In each of the parts below, give the sequence of configurations that M enters when started on the indicated input string.

(a) 00

Answer: $q_1 00$ $\quad \sqcup q_2 0$ $\quad \sqcup x q_3 \sqcup$ $\quad \sqcup q_5 x$ $\quad q_5 \sqcup x$ $\quad \sqcup q_2 x$ $\quad \sqcup x q_2 \sqcup$ $\quad \sqcup x \sqcup q_{\text{accept}}$

(b) 000000

Answer: $q_1 000000$ $\quad \sqcup q_2 00000$ $\quad \sqcup x q_3 0000$ $\quad \sqcup x 0 q_4 000$
$\sqcup x 0 x q_3 00$ $\quad \sqcup x 0 x 0 q_4 0$ $\quad \sqcup x 0 x 0 x q_3 \sqcup$ $\quad \sqcup x 0 x 0 q_5 x$ $\quad \sqcup x 0 x q_5 0 x$
$\sqcup x 0 q_5 x 0 x$ $\quad \sqcup x q_5 0 x 0 x$ $\quad \sqcup q_5 x 0 x 0 x$ $\quad q_5 \sqcup x 0 x 0 x$ $\quad \sqcup q_2 x 0 x 0 x$
$\sqcup x q_2 0 x 0 x$ $\quad \sqcup x x q_3 x 0 x$ $\quad \sqcup x x x q_3 0 x$ $\quad \sqcup x x x 0 q_4 x$ $\quad \sqcup x x x 0 x q_4 \sqcup$
$\sqcup x x x 0 x \sqcup q_{\text{reject}}$