1. (7.1, 10 points) Let $M$ be the PDA defined by

$$Q = \{q_0, q_1, q_2\}$$
$$\Sigma = \{a, b\}$$
$$\Gamma = \{A\}$$
$$F = \{q_1, q_2\}$$

$$\delta(q_0, a, \lambda) = \{[q_0, A]\}$$
$$\delta(q_0, \lambda, \lambda) = \{[q_1, \lambda]\}$$
$$\delta(q_0, b, A) = \{[q_2, \lambda]\}$$
$$\delta(q_1, \lambda, A) = \{[q_1, \lambda]\}$$
$$\delta(q_2, b, A) = \{[q_2, \lambda]\}$$

a) Describe the language accepted by $M$.

b) Give the state diagram of $M$.

c) Trace all computations of the strings $aab$, $abb$, $aba$ in $M$.

d) Show that $aabb, aaab \in L(M)$.

Solution:

a) The PDA $M$ accepts the language $\{a^i b^j \mid 0 \leq j \leq i\}$. Processing an $a$ pushes $A$ onto the stack. Strings of the form $a^i$ are accepted in state $q_1$. The transitions in $q_1$ empty the stack after the input has been read. A computation with input $a^i b^j$ enters state $q_2$ upon processing the first $b$. To read the entire input string, the stack must contain at least $j$ $A$’s. The transition $\delta(q_2, \lambda, A) = [q_2, \lambda]$ will pop any $A$’s remaining on the stack.

b) The state diagram of $M$ is

![State Diagram](image)

c) The computations of $aab$ in $M$ are as follows:

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$aab$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$aab$</td>
<td>$\lambda$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$aab$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$ab$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$ab$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$b$</td>
<td>$\lambda$</td>
</tr>
</tbody>
</table>
The computations of $abb$ in $M$ are as follows:

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$abb$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$abb$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$bb$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$bb$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$bb$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$b$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$b$</td>
<td>$\lambda$</td>
</tr>
</tbody>
</table>

The computations of $aba$ in $M$ are as follows:

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$aba$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$aba$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$ba$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$ba$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$ba$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$a$</td>
<td>$\lambda$</td>
</tr>
</tbody>
</table>

d) To show that the string $aabb$ and $aaab$ are in $L(M)$, we trace a computation of $M$ that accepts these strings.

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$aabb$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$abb$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$bb$</td>
<td>$AA$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$b$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$\lambda$</td>
<td>$\lambda$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$aaab$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$aab$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$ab$</td>
<td>$AA$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$\lambda$</td>
<td>$A$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$\lambda$</td>
<td>$\lambda$</td>
</tr>
</tbody>
</table>

2. (7.2, 10 points) Let $M$ be the PDA in Example 7.1.3.

| $b\lambda/B, a\lambda/A$ | $bB/\lambda, aA/\lambda$ |

$M : \begin{array}{c}
\text{State} \\
\text{String} \\
\text{Stack}
\end{array} \begin{array}{c}
q_0 \\
aaab \\
A \\
A
\end{array} \begin{array}{c}
\lambda \\
A \\
A
\end{array} \begin{array}{c}
q_0 \\
aab \\
B \\
B
\end{array} \begin{array}{c}
\lambda \\
A \\
A
\end{array} \begin{array}{c}
q_0 \\
ab \\
A \\
A
\end{array} \begin{array}{c}
\lambda \\
A \\
A
\end{array} \begin{array}{c}
q_0 \\
bb \\
AA \\
AA
\end{array} \begin{array}{c}
\lambda \\
A \\
A
\end{array} \begin{array}{c}
q_0 \\
\lambda \\
A
\end{array} \begin{array}{c}
\lambda \\
A
\end{array} \begin{array}{c}
q_0 \\
b \\
AAA \\

a) Give the transition table of $M$.
b) Trace all computations of the strings $ab, abb, abbb$ in $M$.
c) Show that $aaaa, baab \in L(M)$.
d) Show that $aaa, ab \notin L(M)$.

Solution:
\( Q = \{ q_0, q_1 \} \)
\( \Sigma = \{ a, b \} \)
\( \Gamma = \{ A, B \} \)
\( F = \{ q_1 \} \)
\( \delta(q_0, b, \lambda) = \{ [q_0, B] \} \)
\( \delta(q_0, a, \lambda) = \{ [q_0, A] \} \)
\( \delta(q_0, \lambda, \lambda) = \{ [q_1, \lambda] \} \)
\( \delta(q_1, b, B) = \{ [q_1, \lambda] \} \)
\( \delta(q_1, a, A) = \{ [q_1, \lambda] \} \)

b) The computations of \( ab \) in \( M \) are as follows:

\[
\begin{array}{ccc}
\text{State} & \text{String} & \text{Stack} \\
q_0 & ab & \lambda \\
q_1 & ab & \lambda \\
\end{array}
\]

The computations of \( abb \) in \( M \) are as follows:

\[
\begin{array}{ccc}
\text{State} & \text{String} & \text{Stack} \\
q_0 & abb & \lambda \\
q_1 & abb & \lambda \\
q_0 & bb & A \\
q_1 & b & B A \\
q_1 & b & BA \\
q_1 & \lambda & A \\
\end{array}
\]

The computations of \( abbb \) in \( M \) are as follows:

\[
\begin{array}{ccc}
\text{State} & \text{String} & \text{Stack} \\
q_0 & abbb & \lambda \\
q_1 & abbb & \lambda \\
q_0 & bbb & A \\
q_1 & bbb & A \\
q_0 & b & B A \\
q_1 & b & B A \\
q_1 & \lambda & B B A \\
q_1 & \lambda & B B A \\
\end{array}
\]

\[ \delta(q_0, b, \lambda) = \{ [q_0, B] \} \]
\[ \delta(q_0, a, \lambda) = \{ [q_0, A] \} \]
\[ \delta(q_0, \lambda, \lambda) = \{ [q_1, \lambda] \} \]
\[ \delta(q_1, b, B) = \{ [q_1, \lambda] \} \]
\[ \delta(q_1, a, A) = \{ [q_1, \lambda] \} \]

b) The computations of \( ab \) in \( M \) are as follows:

\[
\begin{array}{ccc}
\text{State} & \text{String} & \text{Stack} \\
q_0 & ab & \lambda \\
q_0 & b & A \\
q_1 & b & B A \\
q_1 & \lambda & BA \\
\end{array}
\]

The computations of \( abb \) in \( M \) are as follows:

\[
\begin{array}{ccc}
\text{State} & \text{String} & \text{Stack} \\
q_0 & abb & \lambda \\
q_0 & bb & A \\
q_1 & bb & A \\
q_0 & b & B A \\
q_1 & \lambda & B B A \\
q_1 & \lambda & B B A \\
\end{array}
\]

The computations of \( abbb \) in \( M \) are as follows:

\[
\begin{array}{ccc}
\text{State} & \text{String} & \text{Stack} \\
q_0 & abbb & \lambda \\
q_0 & bbb & A \\
q_1 & bbb & A \\
q_0 & b & B A \\
q_1 & b & B A \\
q_1 & \lambda & B B A \\
q_1 & \lambda & B B A \\
\end{array}
\]

\[ \delta(q_0, b, \lambda) = \{ [q_0, B] \} \]
\[ \delta(q_0, a, \lambda) = \{ [q_0, A] \} \]
\[ \delta(q_0, \lambda, \lambda) = \{ [q_1, \lambda] \} \]
\[ \delta(q_1, b, B) = \{ [q_1, \lambda] \} \]
\[ \delta(q_1, a, A) = \{ [q_1, \lambda] \} \]

\[ \delta(q_0, b, \lambda) = \{ [q_0, B] \} \]
\[ \delta(q_0, a, \lambda) = \{ [q_0, A] \} \]
\[ \delta(q_0, \lambda, \lambda) = \{ [q_1, \lambda] \} \]
\[ \delta(q_1, b, B) = \{ [q_1, \lambda] \} \]
\[ \delta(q_1, a, A) = \{ [q_1, \lambda] \} \]

c) To show that the string \( aaaa \) and \( baab \) are in \( L(M) \), we trace a computation of \( M \) that accepts these strings.
d) To show that the string \textit{aaa} and \textit{ab} are not in \(L(M)\), we trace all computations of these strings in \(M\), and check whether none of them accepts these strings. We have listed all the computations of \textit{ab} in (b), and none of them accepts it. Now we trace all computations of \textit{aaa} in \(M\).

Since none of the computations above is accepted, we have \textit{aaa} is not in \(M\).

3. \(7.3, 10\) points  Construct PDAs that accept each of the following languages.

a) \(\{a^i b^j \mid 0 \leq i \leq j\}\)

b) \(\{a^i c^j b^i \mid i, j \geq 0\}\)

c) \(\{a^i b^j c^k \mid i + k = j\}\)

d) \(\{w \mid w \in \{a, b\}^* \text{ and } w \text{ has twice as many } a\text{'s as } b\text{'s}\}\)

e) \(\{a^i b^j \mid i \geq 0\} \cup a^* \cup b^*\)

f) \(\{a^i b^j c^k \mid i = j \text{ or } j = k\}\)

g) \(\{a^i b^j \mid i \neq j\}\)

h) \(\{a^i b^j \mid 0 \leq i \leq j \leq 2i\}\)

i) \(\{a^{i+j} b^i c^j \mid i, j > 0\}\)

j) The set of palindromes over \(\{a, b\}\)

**Solution:**
accepts strings that have twice as many $a$’s as $b$’s. A computation begins by pushing a $C$ onto the stack, which serves as a bottom-maker throughout the computation. The stack is used to record the relationship between the number of $a$’s and $b$’s scanned during the computation. The stacktop will be a $C$ when the number of $a$’s processed is exactly twice the number of $b$’s processed. The stack will contain $n$ A’s if the automaton has read $n$ more $a$’s than $b$’s. If $n$ more $b$’s than $a$’s have been read, the stack will hold $2n$ B’s. When an $a$ is read with an $A$ or $C$ on the top of the stack, an $A$ is pushed onto the stack. This is accomplished by the transition to $q_2$. If a $B$ is on the top of the stack, the stack is popped removing one $b$. If a $b$ is read with a $C$ or $B$ on the stack, two $B$’s are pushed onto the stack. Processing a $b$ with an $A$ on the stack pops the $A$.

The lone accepting state of the automation is $q_5$. If the input string has twice as many $a$’s as $b$’s, the transition to $q_5$ pops the $C$, terminates the computation, and accepts the string.
e) The language $L = \{a^i b^j \mid 0 \leq i \leq j \leq 2 \cdot i\}$ is generated by the context-free grammar

$$S \rightarrow aSB \mid \lambda$$
$$B \rightarrow bb \mid b$$

The $B$ rule generates one or two $b$'s for each $a$. A pushdown automaton $M$ that accepts $L$ uses the $a$'s to record an acceptable number of matching $b$'s on the stack. Upon processing an $a$, the computation nondeterministically pushes one or two $A$'s onto the stack. The transitions
of \( M \) are

\[
\begin{align*}
\delta(q_0, a, \lambda) &= \{ [q_1, A] \} \\
\delta(q_0, \lambda, \lambda) &= \{ [q_3, \lambda] \} \\
\delta(q_0, a, \lambda) &= \{ [q_0, A] \} \\
\delta(q_0, b, A) &= \{ [q_2, \lambda] \} \\
\delta(q_1, \lambda, \lambda) &= \{ [q_0, A] \} \\
\delta(q_2, b, A) &= \{ [q_2, \lambda] \}
\end{align*}
\]

The states \( q_2 \) and \( q_3 \) are the accepting states of \( M \). The null string is accepted in \( q_3 \). For a nonnull string \( a^i b^j \in L \), one of the computations will push exactly \( j \) \( A \)'s onto the stack. The stack is emptied by processing the \( b \)'s in \( q_2 \).

The state diagram of the PDA is

\[
\begin{align*}
\text{i) } M : & \quad \overset{a\lambda/A}{q_0} \xrightarrow{\lambda\lambda/A} q_1 \xrightarrow{b\lambda/\lambda} q_2 \\
\text{j) } M : & \quad \overset{a\lambda/A, b\lambda/B}{q_0} \xrightarrow{a\lambda/\lambda, b\lambda/\lambda, \lambda\lambda/\lambda} q_3
\end{align*}
\]

4. (7.7, 10 points) Let \( L \) be the language \( \{ w \in \{ a, b \}^* \mid w \text{ has a prefix containing more } b \text{'s than } a \text{'s.} \} \). For example, \( baa, abba, abbaaa \in L \), but \( aab, aabbab \notin L \).

a) Construct a PDA that accepts \( L \) by final state.

b) Construct a PDA that accepts \( L \) by empty stack.

**Solution:**

\[
\begin{align*}
\text{a) } M : & \quad \overset{a\lambda/A, b\lambda/\lambda}{q_0} \xrightarrow{\lambda\lambda/B, \lambda\lambda/\lambda} q_1, q_2 \\
\text{b) } M : & \quad \overset{a\lambda/A, b\lambda/\lambda}{q_0} \xrightarrow{\lambda\lambda/B, \lambda\lambda/\lambda} q_1, q_2
\end{align*}
\]
5. (7.12, 20 points) Use the technique of Theorem 7.3.1 to construct a PDA that accepts the languages of the Greibach normal form grammar.

\[ S \rightarrow aABA \mid aBB \]
\[ A \rightarrow bA \mid b \]
\[ B \rightarrow cB \mid c \]

Solution: The state diagram for the extended PDA obtained from the grammar is

\[ Q = \{ q_0, q_1 \} \]
\[ \Sigma = \{ a, b, c \} \]
\[ \Gamma = \{ A, B \} \]
\[ F = \{ q_1 \} \]
\[ \delta(q_0, a, \lambda) = \{ [q_1, ABA], [q_1, BB] \} \]
\[ \delta(q_1, b, A) = \{ [q_1, A], [q_1, \lambda] \} \]
\[ \delta(q_1, c, B) = \{ [q_1, B], [q_1, \lambda] \} \]

6. (7.15, 20 points) Let \( M \) be the PDA in Example 7.1.1.

\[ Q = \{ q_0, q_1 \} \]
\[ \Sigma = \{ a, b, c \} \]
\[ \Gamma = \{ A, B \} \]
\[ F = \{ q_1 \} \]
\[ \delta(q_0, a, \lambda) = \{ [q_0, A] \} \]
\[ \delta(q_0, b, \lambda) = \{ [q_0, B] \} \]
\[ \delta(q_0, c, \lambda) = \{ [q_1, \lambda] \} \]
\[ \delta(q_1, a, A) = \{ [q_1, \lambda] \} \]
\[ \delta(q_1, b, B) = \{ [q_1, \lambda] \} \]

a) Trace the computation in \( M \) that accepts \( bbcbb \).

b) Use the technique from Theorem 7.3.2 to construct a grammar \( G \) that accepts \( L(M) \).

c) Give the derivation of \( bbcbb \) in \( G \).

Solution:

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_0 )</td>
<td>( bbcbb )</td>
<td>( \lambda )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( bcb )</td>
<td>( B )</td>
</tr>
<tr>
<td>( q_0 )</td>
<td>( cbb )</td>
<td>( BB )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( bb )</td>
<td>( BB )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( b )</td>
<td>( B )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( \lambda )</td>
<td>( \lambda )</td>
</tr>
</tbody>
</table>

b) First we add transitions to \( M \) as follows.
\[ \delta(q_0, a, \lambda) = \{[q_0, A]\} \]
\[ \delta(q_0, a, A) = \{[q_0, AA]\} \]
\[ \delta(q_0, a, B) = \{[q_0, AB]\} \]
\[ \delta(q_0, b, \lambda) = \{[q_0, B]\} \]
\[ \delta(q_0, b, A) = \{[q_0, BA]\} \]
\[ \delta(q_0, b, B) = \{[q_0, BB]\} \]
\[ \delta(q_1, a, A) = \{[q_1, \lambda]\} \]
\[ \delta(q_1, a, B) = \{[q_1, \lambda]\} \]
\[ \delta(q_1, b, B) = \{[q_1, \lambda]\} \]

Second the rules of the equivalent grammar \( G \) and the transition from which they were constructed are presented in Table 1.

c) \[
\begin{align*}
[q_0, bbcbb, \lambda] & \quad S \Rightarrow [q_0, \lambda, q_1] \\
[\vdash q_0, bbcbb, B] & \Rightarrow b([q_0, B, q_1]) \\
[\vdash q_0, cbb, BB] & \Rightarrow bb([q_1, B, q_1])([q_1, B, q_1]) \\
[\vdash q_1, bb, BB] & \Rightarrow bbc([q_1, B, q_1])([q_1, B, q_1]) \\
[\vdash [q_1, b, B] & \Rightarrow bbcb([q_1, \lambda, q_1])([q_1, B, q_1]) \\
[\vdash [q_1, \lambda, \lambda] & \Rightarrow bbcb([q_1, \lambda, q_1]) \\
& \Rightarrow bbcbb \\
\end{align*}
\]

7. (7.17, 20 points) Use the pumping lemma to prove that each of the following languages is not context-free.

a) \( \{a^k \mid k \text{ is a perfect square}\} \)

b) \( \{a^i b^j c^i d^j \mid i, j \geq 0\} \)

c) \( \{a^i b^{2i} a^i \mid i \geq 0\} \)

d) \( \{a^i b^j c^k \mid 0 < i < j < k < 2i\} \)

e) \( \{wu^r w \mid w \in \{a, b\}^*\} \)

f) The set of finite-length prefixes of the infinite string

\[ abaababaabaab \ldots ba^na^nb^{n+1}a \ldots \]

Solution:

a) Assume that language \( L \) consisting of strings over \( \{a\} \) whose lengths are a perfect square is context-free. By the pumping lemma, there is a number \( k \) such that every string in \( L \) with length \( k \) or more can be written \( uvwxy \) where
<table>
<thead>
<tr>
<th>Transition</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta(q_0, a, \lambda) = {[q_0, A]}$</td>
<td>$S \rightarrow (q_0, \lambda, q_1)$ $\langle q_0, \lambda, q_0 \rangle \rightarrow a\langle q_0, A, q_0 \rangle$ $\langle q_0, \lambda, q_1 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_0, a, A) = {[q_0, AA]}$</td>
<td>$\langle q_0, A, q_0 \rangle \rightarrow a\langle q_0, A, q_0 \rangle$ $\langle q_0, A, q_0 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow a\langle q_0, A, q_0 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_0, a, B) = {[q_0, AB]}$</td>
<td>$\langle q_0, B, q_0 \rangle \rightarrow a\langle q_0, A, q_0 \rangle$ $\langle q_0, B, q_0 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow a\langle q_0, A, q_0 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow a\langle q_0, A, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_0, b, \lambda) = {[q_0, B]}$</td>
<td>$\langle q_0, \lambda, q_0 \rangle \rightarrow b\langle q_0, B, q_0 \rangle$ $\langle q_0, \lambda, q_1 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_0, b, A) = {[q_0, BA]}$</td>
<td>$\langle q_0, A, q_0 \rangle \rightarrow b\langle q_0, B, q_0 \rangle$ $\langle q_0, A, q_0 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow b\langle q_0, B, q_0 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_0, b, B) = {[q_0, BB]}$</td>
<td>$\langle q_0, B, q_0 \rangle \rightarrow b\langle q_0, B, q_0 \rangle$ $\langle q_0, B, q_0 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow b\langle q_0, B, q_0 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow b\langle q_0, B, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_0, c, \lambda) = {[q_1, \lambda]}$</td>
<td>$\langle q_0, \lambda, q_0 \rangle \rightarrow c\langle q_1, \lambda, q_0 \rangle$ $\langle q_0, \lambda, q_1 \rangle \rightarrow c\langle q_1, \lambda, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_0, c, A) = {[q_1, A]}$</td>
<td>$\langle q_0, A, q_0 \rangle \rightarrow c\langle q_1, A, q_0 \rangle$ $\langle q_0, A, q_1 \rangle \rightarrow c\langle q_1, A, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_0, c, B) = {[q_1, B]}$</td>
<td>$\langle q_0, B, q_0 \rangle \rightarrow c\langle q_1, B, q_0 \rangle$ $\langle q_0, B, q_1 \rangle \rightarrow c\langle q_1, B, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_1, a, \lambda) = {[q_1, \lambda]}$</td>
<td>$\langle q_1, A, q_0 \rangle \rightarrow a\langle q_1, \lambda, q_0 \rangle$ $\langle q_1, A, q_1 \rangle \rightarrow a\langle q_1, \lambda, q_1 \rangle$</td>
</tr>
<tr>
<td>$\delta(q_1, b, \lambda) = {[q_1, \lambda]}$</td>
<td>$\langle q_1, B, q_0 \rangle \rightarrow b\langle q_1, \lambda, q_0 \rangle$ $\langle q_1, B, q_1 \rangle \rightarrow b\langle q_1, \lambda, q_1 \rangle$ $\langle q_0, \lambda, q_0 \rangle \rightarrow \lambda$ $\langle q_1, \lambda, q_1 \rangle \rightarrow \lambda$</td>
</tr>
</tbody>
</table>

Table 1: The rules of the equivalent grammar $G$ and the transition from which they were constructed (Problem 7.15 (b))
(i) \( \text{length}(vwx) \leq k \)
(ii) \( v \) and \( x \) are not both null
(iii) \( uv^iwx^iy \in L \) for all \( i \geq 0 \).

The string \( z = a^{k^2} \) must have a decomposition \( uvwx \) that satisfies the preceding conditions. Consider the length of the string \( uv^2wx^2y \) obtained by pumping \( uvwx \).

\[
\text{length}(z) = \text{length}(uv^2wx^2y) = \text{length}(uvwx) + \text{length}(v) + \text{length}(x) = k^2 + \text{length}(v) + \text{length}(x) \leq k^2 + k < (k + 1)^2
\]

Since the length of \( z \) is greater than \( k^2 \) but less than \( (k + 1)^2 \), we conclude that \( z \notin L \) and that \( L \) is not context-free.

b) Assume that language \( L = \{a^ib^jc^d | i, j \geq 0 \} \) is context-free. By the pumping lemma, there is a number \( k \) such that every string in \( L \) with length \( k \) or more can be written \( uvwx \) where

(i) \( \text{length}(vwx) \leq k \)
(ii) \( v \) and \( x \) are not both null
(iii) \( uv^iwx^iy \in L \) for all \( i \geq 0 \).

The string \( z = a^{k^2}b^{k^2}c^ka^k \) must have a decomposition \( uvwx \) that satisfies the preceding conditions. Consider the string \( uv^2wx^2y \) obtained by pumping \( uvwx \). Since \( v \) and \( x \) are not both null by condition (ii), we have that \( vwx \) contains at least one terminal. Without loss of generality, assume \( vwx \) contains a terminal which is either \( a \) or \( c \) (similar argument for the case that the terminal is either \( b \) or \( d \)). Condition (i) requires the length of \( vwx \) to be at most \( k \). This implies that \( vwx \) is a string that cannot contain both \( a \) and \( c \) types of terminal. Thus \( uv^2wx^2y \) increases the number of either \( a \)’s or \( c \)’s, but not both, compared with \( uvwx \). Hence \( uv^2wx^2y \notin L \), a contradiction. We conclude that \( L \) is not context-free.

c) Assume that language \( L = \{a^ib^{2i}a^i | i \geq 0 \} \) is context-free. By the pumping lemma, there is a number \( k \) such that every string in \( L \) with length \( k \) or more can be written \( uvwx \) where

(i) \( \text{length}(vwx) \leq k \)
(ii) \( v \) and \( x \) are not both null
(iii) \( uv^iwx^iy \in L \) for all \( i \geq 0 \).

The string \( z = a^{k^2}b^{2k}a^k \) must have a decomposition \( uvwx \) that satisfies the preceding conditions. Consider the string \( uv^2wx^2y \) obtained by pumping \( uvwx \). Since by assumption \( uv^2wx^2y \in L \), we must have that the union of \( v \) and \( x \) contains both \( a \) type and \( b \) type of terminals. Otherwise it only increases one type of terminal while keeping the other the same, thereby no longer in \( L \). Further more, condition (i) requires the length of \( vwx \) to be at most \( k \). This implies that the substring \( vwx \) of \( z \) cannot contain \( a \)’s from both sides of the \( b \)’s substring. Therefore \( uv^2wx^2y \) only increases the number of \( a \)’s either preceding or after \( b \)’s, but not both. Hence \( uv^2wx^2y \notin L \), and consequently, \( L \) is not context-free.
d) Assume that language \( L = \{a^ib^jc^k \mid 0 < i < j < k < 2i\} \) is context-free. By the pumping lemma, there is a number \( k \) such that every string in \( L \) with length \( k \) or more can be written \( uvwxy \) where

(i) \( \text{length}(vwx) \leq k \)
(ii) \( v \) and \( x \) are not both null
(iii) \( uv^iwv^jx \in L \) for all \( i \geq 0 \).

Without loss of generality, we assume \( k > 2 \), since we can always increase \( k \) while maintaining the three conditions above. Then the string \( z = a^kb^{k+2}c^k \) is in \( L \) and must have a decomposition \( uvwxy \) that satisfies the preceding conditions. Consider the string \( uv^kw^kxy \) obtained by pumping \( uvwxy \). Condition (i) requires the length of \( vwx \) to be at most \( k \). This implies that \( vwx \) is a string containing only one type of terminal or the concatenation of either \( a \) and \( b \) types, or \( b \) and \( c \) types. If \( c \) is not contained in \( vwx \), pumping \( v \) and \( x \) only increases the number of \( a \)'s or \( b \)'s. Thus the new string cannot keep the number of \( b \)'s less than the number of \( b \)'s which is less than the number of \( c \)'s, i.e. \( k + 2 \). If \( c \) is contained in \( vwx \), then \( a \) is not contained in \( vwx \). Thus \( uv^kw^kxy \) would have at least \( (k + 2) + (k - 1) = 2k + 1 \) number of \( c \)'s while keeping the number of \( a \)'s the same, i.e. \( k \). Hence \( uv^kw^kxy \notin L \), and consequently, \( L \) is not context-free.

e) Assume that language \( L = \{w^R \mid w \in \{a,b\}^*\} \) is context-free. By the pumping lemma, there is a number \( k \) such that every string in \( L \) with length \( k \) or more can be written \( uvwxy \) where

(i) \( \text{length}(vwx) \leq k \)
(ii) \( v \) and \( x \) are not both null
(iii) \( uv^iwv^jx \in L \) for all \( i \geq 0 \).

The string \( z = (a^kb^k)(a^kb^k)^R(a^kb^k) = a^kb^{2k}a^{2k}b^k \) must have a decomposition \( uvwxy \) that satisfies the preceding conditions. By condition (ii), we have \( v \) and \( x \) have at least one terminal. Without loss of generality, assume that at least one \( a \) is in \( v \) or \( x \) (similar argument for the case of at least one \( b \) in \( v \) or \( x \) ). Condition (i) requires the length of \( vwx \) to be at most \( k \). This implies that the substring \( vwx \) of \( z \) cannot contain \( a \)'s from both sides of \( b^{2k} \). If the \( a \)'s in the substring \( vwx \) of \( z \) are before \( b^{2k} \), then \( uv^2wx^2y \) increases the number of \( a \)'s before \( b^{2k} \) while keeping the number of \( a \)'s after \( b^{2k} \) the same as \( 2k \). Hence \( uv^2wx^2y \notin L \) is no longer in \( L = \{w^R \mid w \in \{a,b\}^*\} \). If the \( a \)'s in the substring \( vwx \) of \( z \) are after \( b^{2k} \), we have \( uv^2wx^2y \notin L \) by similar argument. Therefore \( L \) is not context-free.

f) Assume that the language \( L \) consisting of prefixes of string

\[
\text{abaabaaaaaaba} \cdots ba^nba^{n+1}b
\]

is context-free and let \( k \) be the number specified by the pumping lemma. Consider the string \( z = abab \cdots ba^kb \), which is in the language and has length greater than \( k \). Thus \( z \) can be written \( uvwxy \) where

(i) \( \text{length}(vwx) \neq k \)
(ii) $v$ and $x$ are not both null

(iii) $uv^iwx^iy \in L$ for all $i \geq 0$.

To show that the assumption that $L$ is context-free produces a contradiction, we examine all possible decomposition of $z$ that satisfy the conditions of the pumping lemma. By (ii), one or both of $v$ and $x$ must be nonnull. In the following argument we assume that $v \neq \lambda$.

Case 1: $v$ has no $b$'s. In this case, $v$ consists solely of $a$'s and lies between two consecutive $b$'s. That is, $v$ occurs in $z$ in a position of the form

\[ \cdots ba^nva^n+2b \cdots \]

where $i + \text{length}(v) + j = n + 1$. Pumping $v$ produces an incorrect number of $a$'s following $ba^n$ and, consequently, the resulting string is not in the language.

Case 2: $v$ has two or more $b$'s. In this case, $v$ contains a substring $ba^n$. Pumping $v$ produces a string with two substrings of the form $ba^n$. No string with this property is in $L$.

Case 3: $v$ has one $b$. Then $v$ can be written $a^iba^j$ and occurs in $z$ as

\[ \cdots ba^{n-1}ba^n-ivaba^{n+1-j}b \cdots \]

Pumping $v$ produces the substring

\[ \cdots ba^{n-1}ba^n-ia^iba^jababa^{n+1-j}b \cdots = \cdots ba^{n-1}ba^nba^jababa^{n+1}b \cdots , \]

which cannot occur in a string in $L$.

Regardless of its makeup, pumping any nonnull substring $v$ of $z$ produces a string that is not in the language $L$. A similar argument shows that pumping $x$ produces a string not in $L$ whenever $x$ is nonnull. Since one of $v$ or $x$ is nonnull, there is no decomposition of $z$ that satisfies the requirements of the pumping lemma and we conclude that the language is not context-free.

\[ \blacksquare \]