1 \textbf{(10 points)} Let P be a PDA defined as follows.

\[ Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}, \Gamma = \{R\}, F = \{q_1, q_2\} \]

- \( \delta(q_0, 0, \epsilon) = \{(q_0, R)\} \)
- \( \delta(q_0, \epsilon, \epsilon) = \{(q_1, \epsilon)\} \)
- \( \delta(q_0, 1, R) = \{(q_2, \epsilon)\} \)
- \( \delta(q_1, \epsilon, R) = \{(q_1, \epsilon)\} \)
- \( \delta(q_2, 1, R) = \{(q_2, \epsilon)\} \)
- \( \delta(q_2, \epsilon, R) = \{(q_2, \epsilon)\} \)

1. Describe the language accepted by P.

2. Give the state diagram of P.

3. Write down the traces of computations of 001, 010 in P.

4. Show that 0011 \( \in L(P) \).

\textbf{Solution:}  
1. Describe the language accepted by P.
   The language is \( \{0^m1^n \mid 0 \leq n \leq m\} \).

2. Give the state diagram of P.
3. Write down the traces of computations of $001, 010$ in $P$.

Write each computation path of $001$ first.

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$001$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$001$</td>
<td>$\epsilon$</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>$001$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$01$</td>
<td>$R$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$01$</td>
<td>$R$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$1$</td>
<td>$\epsilon$</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>$001$</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$01$</td>
<td>$R$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$1$</td>
<td>$RR$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$1$</td>
<td>$RR$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$1$</td>
<td>$R$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$1$</td>
<td>$\epsilon$</td>
</tr>
</tbody>
</table>

Now we show all the computation paths for $010$. 
4. Show that $0011 \in L(P)$.

To show that $0011 \in L(P)$, it’s enough to give one computation path that halts at an accept state.

<table>
<thead>
<tr>
<th>State</th>
<th>String</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>010</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>010</td>
<td>$\epsilon$</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>010</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>10</td>
<td>$R$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>10</td>
<td>$R$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>10</td>
<td>$\epsilon$</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>010</td>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>10</td>
<td>$R$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>0</td>
<td>$\epsilon$</td>
</tr>
</tbody>
</table>

2 (18 points) Give context-free grammars for each of the following languages. To get full credit, it’s enough to solve four out of six items. More, will be considered bonus points.

1. $L = \{1^{2m}0^m | m \geq 0 \}$
2. $L = \{0,1\}^* \setminus \{1^{2m}0^m | m \geq 0 \}$
3. $L = \{0,1\}^* \setminus \{ww | w \in \{0,1\}^* \}$
4. $L = \{w | \text{Number of } 0's \text{ is exactly twice the number of } 1's \}$
5. $L = \{wz | w,z \in \{0,1\}^* , |w| = |z|, w \neq z \}$
6. $L = \{x^i y^j z^k | i,j,k \geq 0 \text{ and } i + j = k \}$

Solution: In all the solutions the upper case letters are variables and $S$ is the starting variable. The lower case letters are terminals.

1. $L = \{1^{2m}0^m | m \geq 0 \}$

$S \rightarrow \epsilon \mid 11S0$

2. $L = \{0,1\}^* \setminus \{1^{2m}0^m | m \geq 0 \}$ One can write the language $L$ as the union of two languages $L = \{1^*0^*\}^c \cup \{1^m0^m | m \neq 2n \}$. We write a CFG for each of these two languages separately and finally combine them to get a CFG for $L$.

$L_1 = \{1^*0^*\}^c$

Notice that a regular expression for $L_1$ is $(0 \cup 1)^*01(0 \cup 1)^*$. Let $S_1$ be the starting variable in the CFG.
Now we write a CFG for \( L_2 = \{1^m0^n \mid m \neq 2n\} \). Let \( S_2 \) be the starting variable.

\[
S_2 \rightarrow 11S_20 \mid R \mid P \mid T
\]

\[
R \rightarrow 1 \mid 1R
\]

\[
P \rightarrow 0 \mid 0P
\]

\[
T \rightarrow 1 \mid 1P
\]

Now we construct the union of these two CFGs. Let \( S \) be the starting variable.

\[
S \rightarrow S_1 \mid S_2
\]

\[
S_1 \rightarrow X01X
\]

\[
X \rightarrow \epsilon \mid 1X \mid 0X
\]

\[
S_2 \rightarrow 11S_20 \mid R \mid P \mid T
\]

\[
R \rightarrow 1 \mid 1R
\]

\[
P \rightarrow 0 \mid 0P
\]

\[
T \rightarrow 1 \mid 1P
\]

3. \( L = \{0,1\}^*\backslash\{ww \mid w \in \{0,1\}^*\} \)

Note that all strings of odd length are in \( L \) Any even length string in \( L \) can be divided into two odd length strings such that one has a 0 in the center and the other has a 1 in the center.

\[
S \rightarrow XY \mid YX \mid X \mid Y
\]

\[
X \rightarrow 0 \mid ZXZ
\]

\[
Y \rightarrow 1 \mid ZYZ
\]

\[
Z \rightarrow 0 \mid 1
\]

4. \( L = \{w \mid \text{Number of 0's is exactly twice the number of 1's}\} \)

\[
S \rightarrow \epsilon \mid SS \mid 00S1 \mid 0S1S0 \mid 1S00
\]

5. \( L = \{wz \mid w,z \in \{0,1\}^*, |w| = |z|, w \neq z\} \)

Note that this language is exactly the even length strings in language of item 5. Any string in this language can be written as concatenation of two odd length strings such that one has a 0 at the center and the other has a 1 at the center.

\[
S \rightarrow XY \mid YX
\]

\[
X \rightarrow 0 \mid ZXZ
\]

\[
Y \rightarrow 1 \mid ZYZ
\]

\[
Z \rightarrow 0 \mid 1
\]

6. \( L = \{x^iy^jz^k \mid i,j,k \geq 0 \text{ and } i + j = k\} \)

\[
S \rightarrow xSz \mid T
\]

\[
T \rightarrow ySz \mid \epsilon
\]
3 (12 points) Construct push-down automata recognizing the following languages.

1. \( L = \{1^{2m}0^{3m} | m \geq 0 \} \)

2. \( L = \{w \in \{0, 1\}^* \mid w \text{ is a valid string of parentheses.}\} \). For example "\(((\))" \(\in L\), ")()" \(\notin L\).

3. \( L = \{x^i y^j z^k \mid i, j, k \geq 0 \text{ and } i + j = k \} \)

4. \( L = \{x^i y^j z^k \mid i \neq j \text{ or } j \neq k \} \)

Solution: 1. \( L = \{1^{2m}0^{3m} | m \geq 0 \} \)

2. \( L = \{w \in \{0, 1\}^* \mid w \text{ is a valid string of parentheses.}\} \). For example "\(((\))" \(\in L\), ")()" \(\notin L\).

3. \( L = \{x^i y^j z^k \mid i, j, k \geq 0 \text{ and } i + j = k \} \)
4. \( L = \{ x^i y^j z^k \mid i \neq j \text{ or } j \neq k \} \)

**4 (10 points)** Prove that the class of context-free languages is closed under concatenation.

**Solution:** Suppose we have two context-free languages \( L_1 \) and \( L_2 \). We know that there are CFGs \( G_1 = \{ \Sigma_1, R_1, S_1, V_1 \} \) for \( L_1 \) and \( G_2 = \{ \Sigma_2, R_2, S_2, V_2 \} \) for \( L_2 \). Suppose that the variables \( V_1 \) of \( G_1 \) are distinct from the set of variables \( V_2 \) of \( G_2 \). Moreover assume that the starting variable of \( G_1 \) is \( S_1 \) and the starting variable of \( G_2 \) is \( S_2 \). Then we define a new variable \( S \) as the starting variable of our CFG \( G = \{ \Sigma, V, S, R \} \) for \( L = L_1 \circ L_2 \) as follows.

\[
R = R_1 \cup R_2 \cup \{ S \rightarrow S_1 S_2 \} \\
V = V_1 \cup V_2 \cup \{ S \} \\
\Sigma = \Sigma_1 \cup \Sigma_2
\]
Now we prove the construction is correct. Let $L$ be the language defined by the CFG above. We have to prove $L = L_1 \circ L_2$. First, suppose that there is $x \in L$. Therefore there is a computation path for constructing $x$ via the CFG $G$. However, the only rule that would produce $x$ is $S \rightarrow S_1 S_2$. This means $x$ could be written as $x_1 \circ x_2$ such that $S_1$ produces $x_1$ and $S_2$ produces $x_2$. However, since the variables in the rules involving $S_1$ are disjoint from the variables in the rules involving $S_2$; any output of $S_1$ is in $L_1$ and any output of $S_2$ is in $L_2$. This means that the $x_1 \in L_1$ and $x_2 \in L_2$, therefore, $x \in L_1 \circ L_2$.

Secondly, suppose $x \in L_1 \circ L_2$. We show that $x \in L$ as well. We can write $x = x_1 x_2$ such that $x_1 \in L_1$ and similarly, $S_2$ produces $x_2$. Now we can apply the rule $S \rightarrow S_1 S_2$ to produce the desired string $x$. ■

5(bonus) (10 points) Read theorem 2.34 from the book (pumping lemma for context-free languages). Then prove that the following language is not context-free.

$L = \{ w v^R w \mid w \in \{0,1\}^* \}$

**Solution:** Toward contradiction, assume that the language $L$ is context-free. Therefore pumping lemma should apply to $L$. Let $p$ be the pumping length, and choose the string $s = (0^p 1^p)(0^p 1^p)^R((0^p 1^p) = 0^p 1^p 0^p 1^p$. Suppose that we have an arbitrary partition $s = uvwxy$ such that

1. $|vwx| < p$
2. $|uv| > 0$

We show that there is an $i \leq 0$ such that $uv^iwx^iy \notin L$.

There are a few possible cases to consider.

Case 1. $vx$ has at least a 0. Since we know $|vwx| < p$, the substring $vwx$ can not contain 0’s from both sides of $1^p$. If $vx$ contains 0’s from the left side, then pick $i = 2$. This increases the number of 0’s on left side. We show this implies the new string $uv^2wx^2y \notin L$. Suppose $uv^2wx^2y = 0^m 1^{2p+k} 0^{2p} 1^p$ for $m > p$ and $k \geq 0$. We show this string can not be written in the form $uv^kw$. Note that any suitable $w$ must contain the entire substring $0^m$ and some 1’s. This would require have in total $3m$ many 0’s in the entire string. But we simply don’t have this many 0’s.

On the other case, assume that $vx$ contains some 0’s on the right side of $1^p$. Again by a similar argument we have $uv^2wx^2y \notin L$.

Case 2. $vx$ has at least a 1. In this case the argument is similar to the above. ■