Give context-free grammars for each of the following languages.

1. \( \{0^{2n}1^n \mid n \geq 0\} \)
   
   **Solution:** \( S \to \varepsilon \mid 00S1. \)

2. \( \{0^m1^n \mid m \neq 2n\} \)
   
   (Hint: If \( m \neq 2n \), then either \( m < 2n \) or \( m > 2n \).)

   **Solution:**
   
   To simplify notation, let \( \Delta(w) = \#(0, w) - 2\#(1, w) \). Our solution follows the following logic. Let \( w \) be an arbitrary string in this language.
   
   - Because \( \Delta(w) \neq 0 \), then either \( \Delta(w) > 0 \) or \( \Delta(w) < 0 \).
   - If \( \Delta(w) > 0 \), then \( w = 0^i z \) for some integer \( i > 0 \) and some suffix \( z \) with \( \Delta(z) = 0 \).
   - If \( \Delta(w) < 0 \), then \( w = x1^j \) for some integer \( j > 0 \) and some prefix \( x \) with either \( \Delta(x) = 0 \) or \( \Delta(x) = 1 \).
   - Substrings with \( \Delta = 0 \) is generated by the previous grammar; we need only a small tweak to generate substrings with \( \Delta = 1 \).

   Here is one way to encode this case analysis as a DFA. The nonterminals \( M \) and \( L \) generate all strings where the number of \( 0 \)s is More or Less than twice the number of \( 1 \)s, respectively. The last nonterminal generates strings with \( \Delta = 0 \) or \( \Delta = 1 \).

   \[
   S \to M \mid L \quad \{0^m1^n \mid m \neq 2n\} \\
   M \to 0M \mid 0E \\
   L \to L1 \mid E1 \\
   E \to \varepsilon \mid 0 \mid 00E1 \\
   \]

   Here is a different correct solution using the same logic. We either identify a non-empty prefix of \( 0 \)s or a non-empty prefix of \( 1 \)s, so that the rest of the string is as “balanced” as possible. We also generate strings with \( \Delta = 1 \) using a separate non-terminal.

   \[
   S \to AE \mid EB \mid FB \\
   A \to 0 \mid 0A \\
   B \to 1 \mid 1B \\
   E \to \varepsilon \mid 00E1 \\
   F \to 0E \\
   \]

   Alternatively, we can separately generate all strings of the form \( 0^{\text{odd}}1^* \), so that we dont have to worry about the case \( \Delta = 1 \) separately.

   \[
   S \to D \mid M \mid L \\
   D \to 0 \mid 00D \mid D1 \\
   M \to 0M \mid 0E \\
   L \to L1 \mid E1 \\
   E \to \varepsilon \mid 00E1 \\
   \]
Solution:

Intuitively, we can parse any string \( w \in L \) as follows. First, remove the first \( 2k \) 0s and the last \( k \) 1s, for the largest possible value of \( k \). The remaining string cannot be empty, and it must consist entirely of 0s, entirely of 1s, or a single 0 followed by 1s.

\[
S \to 00S1 \mid A \mid B \mid C \quad \{0^m1^n \mid m \neq 2n\}
\]
\[
A \to 0 \mid 0A \quad 0^+
\]
\[
B \to 1 \mid 1B \quad 1^+
\]
\[
C \to 0 \mid 0B \quad 01^+
\]

Let’s elaborate on the above, since \( k \) is maximal, \( w = 0^{2k}w'1^k \). If \( w' \) starts with 00, and ends with a 1, then we can increase \( k \) by one. As such, \( w' \) is either in \( 0^+ \) or \( 1^+ \). If \( w' \) contains both 0s and 1s, then it can contain only a single 0, followed potentially by \( 1^+ \). We conclude that \( w' \in 0^+ + 1^+ + 01^+ \).

3 \( \{0,1\}^* \setminus \{0^{2n}1^n \mid n \geq 0\} \)

Solution:

This language is the union of the previous language and the complement of \( 0^*1^* \), which is \((0+1)^*10(0+1)^*\).

\[
S \to T \mid X \quad \{0,1\}^* \setminus \{0^{2n}1^n \mid n \geq 0\}
\]
\[
T \to 00T1 \mid A \mid B \mid C \quad \{0^m1^n \mid m \neq 2n\}
\]
\[
A \to 0 \mid 0A \quad 0^+
\]
\[
B \to 1 \mid 1B \quad 1^+
\]
\[
C \to 0 \mid 0B \quad 01^+
\]
\[
X \to Z10Z \quad (0+1)^*10(0+1)^*
\]
\[
Z \to \varepsilon \mid 0Z \mid 1Z \quad (0+1)^*
\]

Work on these later:

4 \( \{w \in \{0,1\}^* \mid \#(0,w) = 2 \cdot \#(1,w)\} \) Binary strings where the number of 0s is exactly twice the number of 1s.

Solution:

\[
S \to \varepsilon \mid SS \mid 00S1 \mid 0S1S0 \mid 1S00.
\]

Here is a sketch of a correctness proof; a more detailed proof appears in the homework.

For any string \( w \), let \( \Delta(w) = \#(0,w) - 2 \cdot \#(1,w) \). Suppose \( w \) is a binary string such that \( \Delta(w) = 0 \). Suppose \( w \) is nonempty and has no non-empty proper prefix \( x \) such that \( \Delta(x) = 0 \). There are three possibilities to consider:

- Suppose \( \Delta(x) > 0 \) for every proper prefix \( x \) of \( w \). In this case, \( w \) must start with 00 and end with 1. Thus, \( w = 00x1 \) for some string \( x \in L \).
Suppose $\Delta(x) < 0$ for every proper prefix $x$ of $w$. In this case, $w$ must start with 1 and end with 00. Let $x$ be the shortest non-empty prefix with $\Delta(x) = 1$. Thus, $w = 1X00$ for some string $x \in L$.

Finally, suppose $\Delta(x) > 0$ for some prefix $x$ and $\Delta(x) < 0$ for some longer proper prefix $x$. Let $x$ be the shortest non-empty proper prefix of $w$ with $\Delta < 0$. Then $x = 0y1$ for some substring $y$ with $\Delta(y) = 0$, and thus $w = 0y1z0$ for some strings $y,z \in L$.

5 \{0,1\}^* \setminus \{ww \mid w \in \{0,1\}^*\}.

Solution:

All strings of odd length are in $L$.

Let $w$ be any even-length string in $L$, and let $m = |w| / 2$. For some index $i \leq m$, we have $w_i \neq w_{m+i}$. Thus, $w$ can be written as either $x1y0z$ or $x0y1z$ for some substrings $x, y, z$ such that $|x| = i - 1$, $|y| = m - 1$, and $|z| = m - i$. We can further decompose $y$ into a prefix of length $i - 1$ and a suffix of length $m - i$. So we can write any even-length string $w \in L$ as either $x1xz0$ or $x0xz1z$, for some strings $x, x, z, z$ with $|x| = |x| = i - 1$ and $|z| = |z| = m - i$. Said more simply, we can divide $w$ into two odd-length strings, one with a 0 at its center, and the other with a 1 at its center.

\[
\begin{align*}
S & \rightarrow AB \mid BA \mid A \mid B & \text{strings not of the form } ww \\
A & \rightarrow 0 \mid \Sigma A \Sigma & \text{odd-length strings with 0 at center} \\
B & \rightarrow 1 \mid \Sigma B \Sigma & \text{odd-length strings with 1 at center} \\
\Sigma & \rightarrow 0 \mid 1 & \text{single character}
\end{align*}
\]