1. What is $L(G)$?
$L(G) = \{ w \in \{a,b\}^* | n_a(w) = 2k, n_b(w) = 2l, k, l \in \mathbb{N} \}$, i.e. the language of strings containing an even number of a’s and an even number of b’s.

Consider the following invariant on the sentential forms of $L(G)$: all sentential forms $w$ of $L(G)$ contain one variable, $S$ or $A$, and the parity of $n_a(w)$ and $n_b(w)$ are even if $w$ contains $S$ and odd if $w$ contains $A$.

Initially, the sentential form $w = S$ contains the variable $S$ and $n_a(w)$ and $n_b(w)$ are both even. For an arbitrary sentential form $w$ with one variable $S$ where $n_a(w)$ and $n_b(w)$ are even, applying $S \rightarrow abA$ or $S \rightarrow baA$ makes the parity of $n_a(w)$ and $n_b(w)$ odd and the resulting sentential form will contain the variable $A$. For an arbitrary sentential form $w$ with one variable $A$ where $n_a(w)$ and $n_b(w)$ are odd, applying $A \rightarrow abS$ or $A \rightarrow baS$, makes the parity of $n_a(w)$ and $n_b(w)$ even and the resulting sentential form will contain the variable $S$. The other non-terminal production rules do not change the parity of $n_a(w)$ and $n_b(w)$ or the variable in the sentential form. Thus, the invariant holds.

The only way to produce a sentence (a string of terminals) is by applying $S \rightarrow \lambda$, where the invariant states $n_a(w)$ and $n_b(w)$ are both even in all strings containing $S$. Thus all productions of this grammar have an even number of $a$’s and $b$’s.

2. Construct CFGs for the following languages

   (a) $\{a^ib^ie^{i+j} | i, j \geq 0\} $

   Let $G = (\{S, S_1\}, \{a, b\}, S, P)$ with productions
   $S \rightarrow aSc \mid bS_1c \mid \lambda$
   $S_1 \rightarrow bS_1c \mid \lambda$

   (b) $\{a^ib^j | 0 \leq i \leq j\}$

   Let $G = (\{S\}, \{a, b\}, S, P)$ with productions
   $S \rightarrow aSb \mid Sb \mid \lambda$

3. Consider the language $L = \{a^ib^je^k | i, j, k > 0, i = j \lor j = k\}$
(a) Is \( L \) regular?
Assume \( L \) is regular. Let \( L' = (L \cap a^*b^*c) - a^*bc = \{a^n b^n c \mid n > 1\} \).
\( L' \) is regular by the closure properties of regular languages under intersection and set difference, but \( L' \) can be shown to be non regular by the pumping lemma. This is a contradiction, thus \( L \) is not regular.

(b) Is \( L \) context-free?
Yes. The CFG in Example 5.13 on page 149 of the textbook produces \( L \).

4. For each word, determine whether or not it is generated by each CFG. If it is, draw a derivation tree for the word.

(a) \( aabb \)
1) \( S \Rightarrow aSb \Rightarrow aabb \), 2) No, 3) No, 4) No

(b) \( abaa \)
1) No, 2) \( S \Rightarrow aS \Rightarrow abS \Rightarrow abaa \), 3) No, 4) No

(c) \( abba \)
1) No, 2) \( S \Rightarrow aS \Rightarrow abS \Rightarrow abba \), 3) No, 4) No

(d) \( aaaa \)
1) No, 2) \( S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaaS \Rightarrow aaaa \),
3) \( S \Rightarrow aS \Rightarrow aX \Rightarrow aaXa \Rightarrow aaaa, 4) No

5. Consider the CFG given.

(a) Show that \( G \) is ambiguous.
\( aaaa \) can be derived two ways.
1) \( S \Rightarrow aSA \Rightarrow aaSAA \Rightarrow aaaaA \Rightarrow aaaa \)
2) \( S \Rightarrow aSA \Rightarrow aaA \Rightarrow aaaa \Rightarrow aaaa \)

(b) Find an unambiguous CFG equivalent to \( G \).
The following grammar is generated from a DFA that recognizes \( L(G) \), thus it is unambiguous.
\( S \rightarrow aA \mid \lambda \)
\( A \rightarrow aB \)
\( B \rightarrow aB \mid \lambda \)

(c) Find an unambiguous CFG that generates \( L(G) \setminus \lambda \)
The following grammar is generated from a DFA that recognizes \( L(G) \setminus \lambda \), thus it is unambiguous.
\( S \rightarrow aA \)
\( A \rightarrow aB \)
\( B \rightarrow aB \mid \lambda \)

6. Do Problem 6, Section 6.1 of the text.
\( S \rightarrow aS \mid \lambda \). This language is \( a^* \).

7. Consider the CFG \( G \) given.
(a) Eliminate $\lambda$-productions from $G$.

\[
S \rightarrow AbB \mid B \mid bB \\
A \rightarrow C \mid a \\
B \rightarrow S \mid b \\
C \rightarrow BbS
\]

(b) Eliminate unit productions from (a)

\[
S \rightarrow AbB \mid b \mid bB \\
A \rightarrow BbS \mid a \\
B \rightarrow AbB \mid b \mid bB \\
C \rightarrow BbS
\]

(c) Eliminate useless productions from (b)

\[
S \rightarrow AbB \mid b \mid bB \\
A \rightarrow BbS \mid a \\
B \rightarrow AbB \mid b \mid bB
\]

8. Construct a reduced grammar for the grammar given.

\[
S \rightarrow bbc
\]