2.2 a. The following grammar generates $A$:

$$
S \rightarrow RT \\
R \rightarrow aR \mid \varepsilon \\
T \rightarrow bTe \mid \varepsilon
$$

The following grammar generates $B$:

$$
S \rightarrow TR \\
T \rightarrow aTb \mid \varepsilon \\
R \rightarrow cR \mid \varepsilon
$$

Both $A$ and $B$ are context-free languages and $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$. We know from Example 2.36 that this language is not context free. We have found two CFLs whose intersection is not context free. Therefore the class of context-free languages is not closed under intersection.

b. First, the context-free languages are closed under the union operation. Let $G_1 = (V_1, \Sigma, R_1, S_1)$ and $G_2 = (V_2, \Sigma, R_2, S_2)$ be two arbitrary context free grammars. We construct a grammar $G$ that recognizes their union. Formally, $G = (V, \Sigma, R, S)$ where:

i) $V = V_1 \cup V_2$

ii) $R = R_1 \cup R_2 \cup \{S \rightarrow S_1, \ S \rightarrow S_2\}$

(Here we assume that $R_1$ and $R_2$ are disjoint, otherwise we change the variable names to ensure disjointness)

Next, we show that the CFLs are not closed under complementation. Assume, for a contradiction, that the CFLs are closed under complementation. Then, if $G_1$ and $G_2$ are context free grammars, it would follow that $\overline{L(G_1)}$ and $\overline{L(G_2)}$ are context free. We previously showed that context-free languages are closed under union and so $\overline{L(G_1) \cup L(G_1)}$ is context free. That, by our assumption, implies that $\overline{L(G_1) \cup L(G_1)}$ is context free. But by DeMorgan's laws, $\overline{L(G_1) \cup L(G_1)} = L(G_1) \cap L(G_2)$. However, if $G_1$ and $G_2$ are chosen as in part (a), $\overline{L(G_1) \cup L(G_1)}$ isn't context free. This contradiction shows that the context-free languages are not closed under complementation.

2.6 b. 

$$
S \rightarrow XbXaX \mid T \mid U \\
T \rightarrow aTb \mid Tb \mid b \\
U \rightarrow aUb \mid aU \mid a \\
X \rightarrow aX \mid bX \mid \varepsilon
$$

d. 

$$
S \rightarrow M#P#M \mid P#M \mid M#P \mid P \\
P \rightarrow aPa \mid bPb \mid a \mid b \mid \varepsilon \mid \# \mid \#M^\# \\
M \rightarrow aM \mid bM \mid \#M \mid \varepsilon
$$

Note that we need to allow for the case when $i = j$, that is, some $x_i$ is a palindrome. Also, $\varepsilon$ is in the language since it's a palindrome.

2.13 a. $L(G)$ is the language of strings of 0s and #s that either contain exactly two #s and any number of 0s, or contain exactly one # and the number of 0s on the right-hand side of the # is twice the number of 0s on the left-hand side of the #.

b. Assume $L(G)$ is regular and obtain a contradiction. Let $A = L(G) \cap 0^*10^*$. If $L(G)$ is regular, so is $A$. But we can show $A = \{0^k10^k \mid k \geq 0\}$ is not regular by using a standard pumping lemma argument.

2.19 The grammar generates all strings not of the form $a^k b^k$ for $k \geq 0$. Thus the complement of the language generated is $L(G) = \{a^k b^k \mid k \geq 0\}$. The following grammar generates $\overline{L(G)}$: $\{\{S\}, \{a, b\}, \{S \rightarrow aSb \mid \varepsilon\}, S\}$. 


Problem 3.

(a) Suppose that $K$ were regular. Then the language $\overline{K}$ would be regular (by closure properties of regular languages).

Note that $\overline{K} = \{a, b, \#\}^* - K$

$$= \{w \in \{a, b, \#\}^* \mid \text{w is not of the form } x\#y \text{ or } x \# y \text{ where } x \text{ is a permutation of } y\}.$$ (but $w$ is not necessarily of the form $x\#y$ where $x$ is a permutation of $y$)

But, we consider $\overline{K} \cap a^*\#a^*$, which would be regular. \(\square\)

What is $\overline{K} \cap a^*\#a^*$?

$$\overline{K} \cap a^*\#a^* = \{x\#y \mid x, y \in a^*, \text{ and } x \text{ is a permutation of } y\}$$

$$= \{a^n\#a^n \mid n \geq 0\}.$$

Applying the pumping lemma for regular languages to $\overline{K} \cap a^*\#a^*$ will show the non-regularity of $\overline{K} \cap a^*\#a^*$ — a contradiction to (1) above.
(b) Notice that for all \( x, y \in \{a, b\}^* \),

\[ x \text{ is a permutation of } y \text{ if and only if } \begin{cases} \#_a(x) = \#_a(y) \\
\#_b(x) = \#_b(y). \end{cases} \]

Thus, \( x \$ y \in K \) if and only if \( \begin{cases} \#_a(x) = \#_a(y) \\
or \\
\#_b(x) = \#_b(y). \end{cases} \)

Now, we have:

\[ K = \{ x \$ y | x, y \in \{a, b\}^* \text{ and } \#_a(x) \neq \#_a(y) \} \]

\[ \cup \{ x \$ y | x, y \in \{a, b\}^* \text{ and } \#_b(x) \neq \#_b(y) \}. \]

Each disjunct is generated by a context-free grammar by considering "\( \neq \)" as "\( <\)" and "\( >\)".
2.26 Consider a derivation of \(w\). Each application of a rule of the form \(A \rightarrow BC\) increases the length of the string by 1. So we have \(n - 1\) steps here. Besides that, we need exactly \(n\) applications of terminal rules \(A \rightarrow a\) to convert the variables into terminals. Therefore, exactly \(2n - 1\) steps are required.

**Problem 4.**

(b) \(L = \{ x \in \{a, b\}^* \mid \text{\# of } a \text{ in } x \text{ is divisible by } 3 \} \)

**Idea:** Use variables \(S_k\) (\(k=0, 1, 2\)) to generate all strings \(x \in \{a, b\}^*\) with

\(x = x^r \quad (\text{palindrome})\)

and \(\#_b(x) \equiv k \pmod{3}\).

Start variable

\[ S_0 \rightarrow aS_0a \mid bS_2b \mid a \mid \varepsilon \]

\[ S_2 \rightarrow aS_2a \mid bS_1b \mid b \]

\[ S_1 \rightarrow aS_1a \mid bS_0b \]

Odd-length palindrome centered at \(a\)

Even-length palindrome centered at \(b\)