pg. 183

4(b) \[ L = \{ wcw^R : w \in \{a, b\}^* \} \]

\[ \text{Diagram} \]

4(c) \[ L = \{ a^n b^m c^{n+m} : n \geq 0, m \geq 0 \} \]

\[ \text{Diagram} \]

4(e) \[ L = \{ a^n b^n c^n : n \geq 0 \} \]

\[ \text{Diagram} \]
4(f) \[ L = \{a^n b^m : n \leq m \leq 3n\} \]

\[
\begin{array}{c}
q_0 \xleftarrow{a, \lambda} q_1 \xrightarrow{\lambda, \$} q_2 \xrightarrow{\lambda, \$, \$} q_3 \\
q_1 \xrightarrow{b, \lambda} \lambda \\
\end{array}
\]

**Pg195**

5. \[ S \rightarrow aABB | aAA, \quad A \rightarrow aBB | a, \quad B \rightarrow bBB | A \]

\[ M = (Q, \Sigma, \delta, q_0, \$, \{q_2\}) \]
\[ Q = \{q_0, q_1, q_2\} \]
\[ \Sigma = \{a, b\} \]
\[ \Gamma = \{A, B, S, \$\} \]
\[ \delta = \{ \delta(q_0, \lambda, \$) = \{(q_1, SS)\}, \]
\[ \delta(q_1, a, S) = \{(q_1, ABB), (q_1, AA)\}, \]
\[ \delta(q_1, a, A) = \{(q_1, BB), (q_1, \lambda)\}, \]
\[ \delta(q_1, b, B) = \{(q_1, BB)\}, \]
\[ \delta(q_1, \lambda, B) = \{(q_1, A)\}, \]
\[ \delta(q_1, \lambda, \$) = \{(q_2, \$)\} \} \]

15. \[ M = (\{q_0, q_1\}, \{a, b\}, \{A, S\}, \delta, q_0, \$, \{q_1\}) \]
\[ \delta = \{ \delta(q_0, a, \$) = \{(q_0, AS)\}, \]
\[ \delta(q_0, b, A) = \{(q_0, AA)\}, \]
\[ \delta(q_0, a, A) = \{(q_1, \lambda)\} \} \]

\[ (q_0 A q_1) = a \]
\[ (q_0 S q_0) = a(q_0 A q_0)(q_0 S q_0) | a(q_0 A q_1)(q_1 S q_0) \]
\[ (q_0 S q_1) = a(q_0 A q_0)(q_0 S q_1) | a(q_0 A q_1)(q_1 S q_1) \]
\[ (q_1 S q_0) = (q_0 A q_0)(q_0 S q_0) | (q_0 A q_1)(q_1 S q_0) \]
\[ (q_1 S q_1) = (q_0 A q_0)(q_0 S q_1) | (q_0 A q_1)(q_1 S q_1) \]
\[ (q_0 A q_0) = b(q_0 A q_0)(q_0 A q_1) | a(q_0 A q_1)(q_1 A q_1) \]
\[ (q_0 A q_1) = b(q_0 A q_0)(q_0 A q_1) | a(q_0 A q_1)(q_1 A q_1) \]
\[ (q_1 A q_0) = (q_0 A q_0)(q_0 A q_0) | (q_0 A q_1)(q_1 A q_0) \]
\[ (q_1 A q_1) = (q_0 A q_0)(q_0 A q_1) | (q_0 A q_1)(q_1 A q_1) \]
The production $B \Rightarrow A$ is not in Greibach form. So first we convert this to Greibach form we substitute $A$ with its production, so that $B \Rightarrow aBB|a$. So the $G$ can be written in Greibach form as follows:

$$G:\begin{align*}
S & \Rightarrow aABB|aAA, \\
A & \Rightarrow aBB|a, \\
B & \Rightarrow bBB|A.
\end{align*}$$

We can now construct an npda $M$ corresponding to $G$, where

$M = (\{q_0,q_1,q_f\},T,V \cup \{z\},\delta,q_0,z,\{q_f\})$, where $z \notin V$.

The input alphabet of $M$ is identical with the set of terminals of $G$, i.e. $T = \{a,b\}$. The stack alphabet contains the variables of the grammar, i.e. $V = \{S,A,B,z\}$

First we have the following rules relating to the initial and final states.

$$\delta(q_0,\lambda,z) = \{(q_1,Sz)\},$$
$$\delta(q_1,\lambda,z) = \{(q_1,z)\}.$$ 

Now we write rules for each production. For example,

Rule for $S \Rightarrow aABB$ is $\delta(q_1,a,S) = \{(q_1,ABB)\}$ and
Rule for $A \Rightarrow a$ is $\delta(q_1,a,A) = \{(q_1,\lambda)\}$.

Following similar procedure, we can find rules for other productions. So we can write the npda for $G$ as follows:

$$\delta(q_0,\lambda,z) = \{(q_1,Sz)\},$$
$$\delta(q_1,a,S) = \{(q_1,ABB), (q_1,AA)\},$$
$$\delta(q_1,a,A) = \{(q_1,BB), (q_1,\lambda)\},$$
$$\delta(q_1,b,B) = \{(q_1,BB)\},$$
$$\delta(q_1,a,B) = \{(q_1,BB), (q_1,\lambda)\},$$
$$\delta(q_1,\lambda,z) = \{(q_1,z)\}.$$
Given the npda \( M = (\{q_0,q_1\}, \{a,b\}, \{A,z\}, \delta, q_0, z, \{q_1\}) \), with transitions

\[
\begin{align*}
\delta(q_0,a,z) &= \{(q_0,Az)\}, \\
\delta(q_0,b,A) &= \{(q_0,AA)\}, \\
\delta(q_0,a,A) &= \{(q_1,#)\}.
\end{align*}
\]

First we note that although the npda \( M \) has single accept state, it is not entered when the stack is empty. In order to satisfy the condition that the single accept state should be entered if and only if the stack is empty, we introduce a new state \( q_2 \) and an intermediate step in which we first remove the \( A \) from the stack to go the new state \( q_2 \) and then in next move we go from \( q_2 \) to the final state \( q_1 \) with the empty stack. So the new set of transition rules is

\[
\begin{align*}
\delta(q_0,a,z) &= \{(q_0,Az)\}, \\
\delta(q_0,b,A) &= \{(q_0,AA)\}, \\
\delta(q_0,a,A) &= \{(q_2,#)\}, \\
\delta(q_2,#,z) &= \{(q_1,\lambda)\}.
\end{align*}
\]

Also we note that the condition that each move either increases or decreases the stack content by a single symbol is satisfied for both the given and the new transition rules.

Let us denote the variable \((q_iAq_j)\) by \(A_{ij}\) and the variable \((q_izq_j)\) by \(B_{ij}\)

For the last two transition rules yield the following productions:

\[
\begin{align*}
A_{02} &\rightarrow a, \\
B_{21} &\rightarrow \lambda.
\end{align*}
\]

For the first transition rule, we can write the following productions:

\[
\begin{align*}
B_{00} &\rightarrow aA_{00}B_{00} | aA_{01}B_{10} | aA_{02}B_{20}, \\
B_{01} &\rightarrow aA_{00}B_{01} | aA_{01}B_{11} | aA_{02}B_{21}, \\
B_{02} &\rightarrow aA_{00}B_{02} | aA_{01}B_{12} | aA_{02}B_{22}.
\end{align*}
\]

Similarly for the 2\textsuperscript{nd} transition rule, we can write the following productions:

\[
\begin{align*}
A_{00} &\rightarrow bA_{00}A_{00} | bA_{01}A_{10} | bA_{02}A_{20}, \\
A_{01} &\rightarrow bA_{00}A_{01} | bA_{01}A_{11} | bA_{02}A_{21}, \\
A_{02} &\rightarrow bA_{00}A_{02} | bA_{01}A_{12} | bA_{02}A_{22}.
\end{align*}
\]

Now we note that the variables \(B_{10}, B_{11}, B_{12}, B_{20}, B_{22}, A_{10}, A_{11}, A_{12}, A_{20}, A_{21}\) and \(A_{22}\) do not occur on the left side and we can eliminate the productions that contain these variables. Therefore we are left with the context-free grammar with following productions:
8(b).

$L = \{ a^n b^j a^n b^l : n+j \leq k+1 \}$ is context free, as we can easily find an npda $M$ for this language as follows:

$M = (\{q_0,q_1,q_2,q_3,q_4,q_5\},\{a,b\},\{0,z\},\delta,q_0,z,\{q_1\})$, and the transitions are

\begin{align*}
\delta(q_0,\lambda,\lambda) &= (q_1,z), \\
\delta(q_1,a,\lambda) &= (q_1,0), \\
\delta(q_1,b,\lambda) &= (q_2,0), \\
\delta(q_2,b,\lambda) &= (q_2,0), \\
\delta(q_2,a,0) &= (q_3,\lambda), \\
\delta(q_3,a,0) &= (q_3,\lambda), \\
\delta(q_3,b,0) &= (q_4,\lambda), \\
\delta(q_4,b,0) &= (q_4,\lambda), \\
\delta(q_4,\lambda,0) &= (q_4,\lambda), \\
\delta(q_4,\lambda,z) &= (q_5,\lambda).
\end{align*}

8(e).

$L = \{ a^n b^j a^n b^l : n \leq k, j \leq l \}$ is NOT context free, as we can find contradiction to the pumping lemma as follows:

Consider pumping length $m$ and the string $w = a^m b^m a^m b^m$.

Assume $L$ is context free, then by pumping lemma, $w = uvxyz$, where $|vxy| \leq m$, $|vy| \geq 1$. Now $vxy$ cannot contain characters $a,b$ from more than two adjacent groups, as alternate groups are separated at least by $m$ characters. It can be shown that for any choice of $vxy$, we pump $vy$ to violate at least one of the conditions $n \leq k$ and $j \leq l$. For example, if we choose $vxy$ from 2nd group of $b$ and 3rd group of $a$, then we can pump up so that $j \geq l$. So the pumped string is not in $L$, which is a contradiction.

8(f).

$L = \{ a^n b^n c^j : n \leq j \}$ is NOT context free as can be shown by pumping lemma.

Consider pumping length $m$ and the string $w = a^m b^m c^m$.

Now, any choice of $vxy$ cannot involve all $a,b$ and $c$, since $a$ and $c$ are separated by $m$ characters. Therefore, by pumping appropriately we can always violate the condition $n \leq j$ or the condition $|a| = |b|$.
A_{02} \rightarrow a,
B_{21} \rightarrow \lambda,
B_{00} \rightarrow aA_{00}B_{00},
B_{01} \rightarrow aA_{00}B_{01} | aA_{02}B_{21},
B_{02} \rightarrow aA_{00}B_{02},
A_{00} \rightarrow bA_{00}A_{00},
A_{01} \rightarrow bA_{00}A_{01},
A_{02} \rightarrow bA_{00}A_{02}.
L = \{ w \in \{a,b,c\}^* : n_a(w) = n_b(w) = 2n_c(w) \} is NOT context-free as we can argue that we cannot construct an npda for L.

If we try to construct an npda such that we push twice when reading c, and pop once when reading a or b then we get $2n_c(w) = n_a(w) + n_b(w)$, but there is no way to tell whether $n_a(w) = n_b(w)$. On the other hand, if we construct the npda such that $n_a(w) = n_b(w)$, then there is no way to tell if $n_a(w) = 2n_c(w)$. 
$L = \{ w \in \{a,b,c\}^* : \text{n}_a(w) \neq \text{n}_b(w) \cup \text{n}_c(w) = \text{n}_c(w) \}$

+ $L$ is context-free, b/c:

1) $L = L_1 \cup L_2$, where $L_1 = \{ w \in \{a,b,c\}^* : \text{n}_a(w) \neq \text{n}_b(w) \}$
   $L_2 = \{ w \in \{a,b,c\}^* : \text{n}_a(w) \neq \text{n}_c(w) \}$

1.1) It is easy to show $L_1, L_2$ are context-free by construction, since
2) Context-free languages are closed under union.

+ $L$ is not context-free b/c:

   - [$\text{n}_a + \text{n}_b \cup \text{n}_c \equiv \text{n}_a = \text{n}_b \land \text{n}_a = \text{n}_c \rightarrow \text{n}_a = \text{n}_b = \text{n}_c$]

1) $L = \{ w \in \{a,b,c\}^* : \text{n}_a(w) = \text{n}_b(w) = \text{n}_c(w) \}$

2) $L \cap \{a^*b^*c^*\}$ regular
   - not context-free, easy to show by pumping lemma.

3) Context-free languages are closed under regular intersection.