1. Let $L$ be the language defined in Example 6.11 on page 179 of the sixth edition, or on page 176 of the fifth edition. Work either Exercise 1 on page 180 of the sixth edition, or Exercise 1 on page 177 of the fifth edition. (Warning: these two exercises are not identical.)

The grammar is:

$$
S \rightarrow AB \\
A \rightarrow BB | a \\
B \rightarrow AB | b
$$

Use the CYK algorithm to determine if the given strings are in $L$. The fifth edition gives the strings $aabb$, $aabba$, and $abbbb$. The sixth edition gives the strings $abb$, $bbb$, $aabba$, and $abbbb$.

Thus, $bbb \in L$, but none of the others are in $L$.

The notation I used in my discussion of push-down automata in class on October 14, 2019, differs from the notation introduced in our textbook, Linz, in some important ways. As you work the homework problems, please use the notation given in the textbook. My notation was derived from the notation used for LALR parsers in the “Dragon” book, a standard text for compiler theory.

In the notation given in Linz, an $\text{i.d.}$ is an ordered triple $(\text{state}, \text{stack}, \text{stream})$, where the stack is written top-to-bottom and the input stream is written with the next input symbol on the left.

The left side of each transition rule is of the form $\delta(\text{state}, \text{input symbol or } \lambda, \text{stack symbol})$, and the right side is a list of ordered pairs of the form $(\text{state}, \text{string of stack symbols})$.

For example, suppose that $\delta(q_1, a, x) = \{(q_2, ax), (q_3, \lambda)\}$, and the current $\text{i.d.}$ is $(q_1, aab, xxz)$, then the next $\text{i.d.}$ is either $(q_2, ab, axxz)$ or $(q_3, ab, xz)$. The following figure shows part of the transition diagram.

This is the state diagram for a DPDA which accepts the Dyck language. For a DPDA to work, we must assume every input string ends with an $\text{eof}$ symbol, which we denote as $\$$.
And this is the transition table.

<table>
<thead>
<tr>
<th>state</th>
<th>read</th>
<th>pop</th>
<th>state</th>
<th>push</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$\lambda$</td>
<td>$z$</td>
<td>$q_1$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$a$</td>
<td>$z$</td>
<td>$q_0$</td>
<td>$az$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$a$</td>
<td>$a$</td>
<td>$q_0$</td>
<td>$aa$</td>
</tr>
<tr>
<td>$q_0$</td>
<td>$b$</td>
<td>$a$</td>
<td>$q_0$</td>
<td>$\lambda$</td>
</tr>
</tbody>
</table>

2. This just a yes/no question! Consider the ambiguous context-free grammar $G$ for algebraic expressions, where the start symbol is $E$

$$
E \rightarrow E + E \\
E \rightarrow E - E \\
E \rightarrow E \ast E \\
E \rightarrow -E \\
E \rightarrow (E) \\
E \rightarrow x \\
E \rightarrow y \\
E \rightarrow z
$$

Does there exist a DPDA which accepts $L(G)$? Hint: You can use an LALR parser for this grammar. Yes. The LALR parser is deterministic.

3. Work problem 6(b) on page 189 of the sixth edition, which problem 4(b) on page 185 of the fifth edition. Give a state diagram as well as the transition function. Can this language be accepted by a DPDA?

Yes, there is a DPDA that accepts $\{a^nb^n\}$. 
4. We write \( L_1 \subseteq_p L_2 \) to mean that there is a \( P \)-time reduction of \( L_1 \) to \( L_2 \). For example:
   \[
   \text{SAT} \subseteq_p \text{3-SAT} \subseteq_p \text{Independent Set} \subseteq_p \text{Subset Sum} \subseteq_p \text{Partition}
   \]

Prove that \( \text{Subset Sum} \subseteq_p \text{Partition} \).

An instance of the subset sum problem consists of a number \( K \) and a set \( X \) of items \( I_1, \ldots, I_n \) where the weight of \( I_i = x_i \). The question is whether there is a subset \( S \subseteq X \) of weight \( K \).

We reduce that instance of the subset sum problem to an instance of the partition problem. Let \( W = \sum_{i=1}^{n} x_i \). Let \( X^* \) consist of \( X \) together with two additional items, \( I_{n+1} \) and \( I_{n+2} \) of weights \( x_{n+1} = W - K + 1 \) and \( x_{n+2} = K + 1 \), respectively. The weight of \( X^* \) is \( \sum_{i=1}^{n+2} x_i = 2W + 2 \), and the question is whether \( X^* \) can be partitioned into two subsets each of weight \( W + 1 \).

If there is a solution \( S \) to the subset sum instance, let \( S^* = S \cup \{I_{n+1}\} \), a subset of \( X^* \) of weight \( K + (W - K + 1) = W + 1 \), and we are done.

Conversely, suppose there is a solution to the partition instance. The items \( I_{n+1} \) and \( I_{n+2} \) cannot be in the same subset, since the total weight of that subset would be at least \( K + 1 + W - K + 1 = W + 2 > W + 1 \). Let \( S^* \) be the subset of weight \( W + 1 \) which contains \( I_{n+1} \) and let \( S \) be the subset of \( X \) obtained by removing \( I_{n+1} \) from \( S^* \). The weight of \( S \) is \( W + 1 - (W - K + 1) = K \). Thus, \( S \) is a solution to the subset sum instance.