From lecture 7:

Theorem 4.9.

If L_1 and L_2 are context-free languages over an alphabet Σ, then

$$L_1 \cup L_2, \quad L_1 L_2 \quad \text{and } L_1^*$$

are also CFLs.

Exercise 5.19.

Suppose M_1 and M_2 are PDAs accepting L_1 and L_2, respectively. For both the languages L_1L_2 and L_1^*, describe a procedure for constructing a PDA accepting the language.

In each case, nondeterminism will be necessary. Be sure to say precisely how the stack of the new machine works; no relationship is assumed between the stack alphabets of M_1 and M_2.

Answer begins with:
Let $M_1 = (Q_1, \Sigma, \Gamma_1, q_{01}, Z_{01}, A_1, \delta_1)$
and let $M_2 = (Q_2, \Sigma, \Gamma_2, q_{02}, Z_{02}, A_2, \delta_2)$.

Exercise 5.21.

Prove the converse of Theorem 5.28:
If there is a PDA $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ accepting L by empty stack (that is, $x \in L$ if and only if $(q_0, x, Z_0) \vdash_M^* (q, \Lambda, \Lambda)$ for some state q),
then there is a PDA M_1 accepting L by final state (i.e., the ordinary way).

Exercise 5.25.

A *counter automaton* is a PDA with just two stack symbols, A and Z_0, for which the string on the stack is always of the form $A^n Z_0$ for some $n \geq 0$.
(In other words, the only possible change in the stack contents is a change in the number of A's on the stack.)
For some context-free languages, such as *AnBn*, the obvious PDA to accept the language is in fact a counter automaton.
Construct a counter automaton to accept the given language in each case below.

a. $\{x \in \{a, b\}^* \mid n_a(x) = n_b(x)\}$

b. $\{x \in \{a, b\}^* \mid n_a(x) = 2n_b(x)\}$

Exercise 5.28.

In each case below, you are given a CFG G and a string x that it generates.
For the top-down PDA $NT(G)$, trace a sequence of movs by which x is accepted, showing at each step the state, the unread input, and the stack contents.
Show at the same time the corresponding leftmost derivation of x in the grammar. See Example 5.19 for a guide.

b. The grammar has productions $S \to S + S \mid S * S \mid [S] \mid a$, and $x = [a * a + a]$.

Exercise 5.34.

In each case below, you are given a CFG G and a string x that it generates.
For the top-down PDA *NB*(G), trace a sequence of movs by which x is accepted, showing at each step the stack contents and the unread input.
Show at the same time the corresponding rightmost derivation of x (in reverse order) in the grammar. See Example 5.24 for a guide.

a. The grammar has productions $S \to S[S] \ \mid \ \Lambda$ and $x = [][[]]$.