From exercise class 10:

Exercise 5.18.

For each of the following languages, give a transition diagram for a deterministic PDA that accepts that language.

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

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

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

d. $\{a^n b^{n+m} a^m \mid n, m \geq 0\}$

From exercise class 10:

Exercise 5.16.

Show that if L is accepted by a PDA, then L is accepted by a PDA that never crashes (i.e., for which the stack never empties and no configuration is reached from which there is no move defined).

From lecture 10:

Stack in PDA contains symbols from certain alphabet.

Usual stack operations: pop, top, push

Extra possiblity: replace top element X by string α

$\alpha = \Lambda$	pop
$\alpha = X$	top
$\alpha = YX$	push
$\alpha = \beta X$	push*
$\alpha = \ldots$	

Top element X is required to do a move!

From exercise class 10:

Exercise 5.17.

Show that if L is accepted by a PDA, then L is accepted by a PDA in which every move
* either pops something from the stack (i.e., removes a stack symbol without putting anything else on the stack);
* or pushes a single symbol onto the stack on top of the symbol that was previously on top;
* or leaves the stack unchanged.

Hence, each action on the stack due to a move in the PDA has one of the following forms:
* either X/Λ (with $X \in \Gamma$),
* or X/YX (with $X, Y \in \Gamma$),
* or X/X (with $X \in \Gamma$).

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 5.25.

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.
Draw the nondeterministic top-down PDA $NT(G)$.

Trace a sequence of moves in $NT(G)$ 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 $x = [a * a + a]$.