Exercise 1. Here’s a question in three parts:

a. Prove that \( \{0^m1^00^{m+n} \mid m, n \geq 0\} \) is not regular.

b. Prove that \( \{0^{f(n)} \mid n \geq 0, f(n) = n^2\} \) is not regular. What is it about \( f \) that causes this language to not be regular?

c. Prove that \( \{ww \mid w \in \{0,1\}^*\} \) is not regular. Hint: consider the intersection with \( 0^*10^*1 \), and use the Pumping Lemma.

Exercise 2. Prove that for every integer \( k > 1 \) a language \( A_k \subseteq \{0,1\}^* \) exists that is recognized by a DFA with \( k \) states but not by one with only \( k - 1 \) states.

Exercise 3. Suppose \( L \) is a regular language over \( \Sigma = \{0,1\} \); define

\[
\text{cycle}(L) = \{yx \mid xy \in L \text{ and } x, y \in \Sigma^*\}.
\]

For example, were \( \Sigma = \{0, \ldots, 9\} \), we would have \( \text{cycle}(\{123, 45\}) = \{123, 231, 312, 45, 54\} \). Show that \( \text{cycle}(L) \) is regular. (Hint: show how to define a finite set of pairs of regular expressions \( (\lambda_i, \rho_i) \) where \( L \) is the language defined by \( \bigcup \lambda_i \cdot \rho_i \) and for any \( x, y \in \Sigma^* \), \( xy \in L \) exactly when for some \( i, x \in \lambda_i \) and \( y \in \rho_i \).)

Exercise 4. Let \( L \) be any regular language over \( \Sigma^* \), and define

\[
E(L) = \{x_2x_4 \cdots x_{2n} \mid x_1x_2x_3 \cdots x_{2n} \in L, x_i \in \Sigma\}.
\]

Show that \( E(L) \) is also regular.

Exercise 5. Let \( \Sigma = \{0,1\} \), and define \( L \) to be the words in \( \Sigma^* \) which can be written as

\[
w = x_0y_0z_0x_1y_1z_1 \cdots x_ny_nz_n,
\]

where

\[
x_nx_{n-1} \cdots x_0 + y_ny_{n-1} \cdots y_0 = z_nz_{n-1} \cdots z_0
\]

is a valid calculation in binary. (This language can be thought of as an “addition checking” language.) Design a deterministic finite automaton to accept this language. (Hint: think about the algorithm you use for addition in base 2.)
Exercise 6. (A little difficult... ) Let $L_1$ and $L_2$ be regular languages over $\Sigma = \{0, 1\}$. We define the binary addition of two such languages as

$$L_1 \oplus L_2 = \{ z \in \Sigma^* \mid \exists x \in L_1 \exists y \in L_2 \text{ where } x + y = z \text{ (base 2)} \}$$

Show that $L_1 \oplus L_2$ is regular. Hint: Use the ideas of the previous two exercises: one lets you "addition check", and the other lets you "skip" over bits in the addition checking language.

Exercise 7. Let $G$ be the grammar

$$S \rightarrow aB | bA
A \rightarrow a | aS | bAA
B \rightarrow b | bS | aBB$$

Draw a parse tree for the string $w = aaabbabba$, using the grammar $G$. Then write a leftmost derivation for $w$, where in each step, we expand the leftmost variable using a production rule.

Exercise 8. Give context-free grammars that generate the following languages. In all parts the alphabet $\Sigma$ is $\{0, 1\}$.

a. $\{w \mid$ the length of $w$ is odd and its middle symbol is a $0\}$

b. $\{w \mid w$ is identical to the reversal of $w\}$

c. $\{w \mid w \neq 0^n1^n$ for any $n \geq 0\}$

d. $\{w \mid w$ contains an equal number of 1s and 0s\}$

Exercise 9. Let $G$ be the CFG generating well-formed formulas of propositional calculus with variables $p$ and $q$:

$$S \rightarrow \sim S \mid [S \supset S] \mid p \mid q.$$ The terminals of the grammar are $T = \{p, q, \supset, \sim, [\ ].\}$. Find a Chomsky normal form grammar generating $L(G)$. Use the dynamic programming parsing algorithm, described in class, to compute a parse tree of $\sim [p \supset \sim p]$.

Exercise 10. Give a context-free grammar that generates the language

$$A = \{a^i b^j c^k \mid i, j, k \geq 0 \text{ and either } i = j \text{ or } j = k\}.$$

Is your grammar ambiguous? (If you don’t know what the technical meaning of that is, look it up.) Why or why not?

Exercise 11. Show that if all the productions of a CFG are of the form $A \rightarrow wB$ or $A \rightarrow w$ where $w \in \Sigma^*$, then the grammar generates a regular language.

Exercise 12. (A little difficult... ) Give context-free grammars that generate the following languages.

a. $\{xy \mid x, y \in \{0, 1\}^* \text{ and } |x| = |y| \text{ but } x \neq y \}$

b. $\{x\#y \mid x, y \in \{0, 1\}^* \text{ and } x \neq y\}$

Hint: you can use (a) to solve (b). To solve (a), realize that $x$ and $y$ have to have some bit that is different in some position.