1. Read Kozen Chapters 19-20.

2. Give a CFG for the set of strings over the alphabet \{a,b\} that are **not** palindromes. Justify the correctness of your grammar with an explanation.

3. Consider the set

   \[ B = \{ x \in \{a,b\}^* \mid \#a(x) \leq 2 \cdot \#b(x) \} , \]

   the set of all strings over \{a,b\} with no more than twice as many a's as b's. Give a CFG for B.

4. Give a CFG for the set PAREN2 of balanced strings of parentheses of two types ( ) and [ ]. For example, ( [ ( ) ] ( [ ] ) ) is in PAREN2 but [ ( ) ] is not.

5. For each of the following languages, give a context-free grammar that generates the language.

   (a) \{ 0^i 1^j \mid i \geq 0 \}^* 
   (b) \{ 0^i 1^j 0^k \mid i = j \text{ or } j = k \} 
   (c) \{ 0^i 1^j \mid i \leq 2j \} 
   (d) \{ w \in \{0, 1\}^* \mid \#1(w) = \#0(w) \} 
   (e) **Optional Challenge:** \{ w \in \{0, 1\}^* \mid \neg (\exists i, j) (w = (0^i 1^j)^j) \}

6. **Optional Challenge:** Our definition of regular expressions earlier in the semester employed an implicit precedence of operators *, \cdot, +, and did not mention parentheses. Design a CFG to specify the set of regular expressions over an alphabet \(\Sigma\) that makes the order of precedence explicit. The grammar you come up with should have terminal symbols \(\Sigma\) and terminal symbols for the null string, the empty set, +, \cdot (the concatenation operator), left and right parentheses, and *. That is, the alphabet of the grammar should be \(\Sigma \cup \{e, \emptyset, +, \cdot, (, )^*, \}\). (Note that the \(e\) here is a symbol in a regular expression, not the empty string.)