1. (a) (5 points) Find a left-most derivation for $aaabbabbba$ in the following context-free grammar:

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

**Answer:**

$$
S \Rightarrow aB \Rightarrow aaBB \Rightarrow aaaBBB \Rightarrow aaabBB \Rightarrow aaabbB \Rightarrow aaabbaBB \\
\Rightarrow aaabbbB \Rightarrow aaabbbS \Rightarrow aaabbbba \Rightarrow aaabbbbaa
$$

(b) (5 points) Draw the corresponding parse-tree of your left-most derivation.

2. (5 Points) Show that each prefix\(^1\) of every word in the language of the following context-free grammar has at least as many 0's as 1's.

$$
S \rightarrow 0S \mid 0S1S \mid \varepsilon
$$

**Answer:** Let $L$ be the corresponding language. We prove this by induction on $m$, the size of the word. The base of induction trivially holds for $m = 0$. Now assume that the statement

\(^1\)A prefix is a substring that starts from the beginning of the word.
holds for every word of size less than some \( m > 0 \) (Induction hypothesis). We want to prove the statement for \( m \).

Let \( w \in L \) be a word of length \( m \), and consider a left-most derivation for \( w \). That is

\[
S \Rightarrow \ldots \Rightarrow w.
\]

We consider the three cases:

- The first rule that is applied is \( S \Rightarrow \varepsilon \): Then \( w = \varepsilon \), and we can verify it immediately.
- The first rule that is applied is \( S \Rightarrow 0S \): Then \( w = 0u \), where \( u \in L \). By induction hypothesis, every prefix of \( u \) has at least as many 0’s as 1’s. Adding a 0 to the beginning of \( u \) certainly will not violate this condition. So every prefix of \( w = 0u \) also has as many 0’s as 1’s.
- The first rule that is applied is \( S \Rightarrow 0S1 \): Then \( w = 0u1v \), where \( u, v \in L \). By induction hypothesis, prefixes of \( u \) and \( v \) have at least as many 0’s as 1’s. Consider a prefix of \( w \).

3. (25 points) For each one of the following languages give a proof that it is or is not regular.

(a) \( \{ 0^m1^n | m \geq 5 \text{ and } n \geq 0 \} \).

**Answer:** It is regular: We can express it as \( 0^50^*1^* \).

(b) \( \{ 0^m1^n | m \geq n^2 \} \).

**Answer:** It is not regular. Suppose to the contrary that it is regular. Consider the pumping constant \( p > 0 \) and set \( w = 0^p1^p \). By pumping lemma we can decompose it as \( w = xyz \), where

- \( |xy| \leq p \) which here implies that \( y \in 0^* \) and \( |y| \leq p \);
- \( |y| > 0 \);
- \( xy^iz \) is in the language for every \( i \geq 0 \).

Set \( i = 0 \). Note that \( xy^iz = xz = 0^{p^2-|y|}1^p \notin \{ 0^m1^n | m \geq n^2 \} \) which is a contradiction.

(c) The set of strings in \( \{ 0, 1 \}^* \) which are not of the form \( ww \) for some \( w \in \{ 0, 1 \}^* \).

**Answer:** It is not regular. Its complement is \( \{ ww | w \in \{ 0, 1 \}^* \} \) which (as we saw in the class) by applying pumping lemma to \( 0^p1^p \) can be shown that is not regular. Since regular languages are closed under complementing, the language in the question is not regular.

(d) \( \{ 0^{[\sqrt{n}]} | n = 0, 1, 2, \ldots \} \),

where for a real number \( x \), \( \lfloor x \rfloor \) denotes the largest integer that is less than or equal to \( x \).

**Answer:** Since \( \lfloor \sqrt{k^2} \rfloor = k \), we have that

\[
\{ 0^{[\sqrt{n}]} | n = 0, 1, 2, \ldots \} = \{ 0^k | k = 0, 1, 2, \ldots \},
\]

which is a regular language as it can be expressed as \( 0^* \).
(e) The first two Fibonacci numbers are 0 and 1, and each subsequent number is the sum of the previous two: 0, 1, 1, 2, 3, 5, 8, 13, ... Now the language in question is

\[ \{0^n|n \text{ is a Fibonacci number}\}. \]

**Answer:** It is not regular. Denote the \( k \)'th Fibonacci number by \( f_k \). Suppose to the contrary that it is regular. Consider the pumping constant \( p > 0 \). Pick a number \( k \) so that \( f_k > p \). Set \( w = 0^{f_k+1} \) By pumping lemma we can decompose it as \( w = xyz \), where

- \( |xy| \leq p; \)
- \( |y| > 0; \)
- \( xy^i z \) is in the language for every \( i \geq 0 \).

Set \( i = 2 \). Note that \( xy^2 z = xz = 0^{f_k+1+|y|} \). But \( f_{k+1} < f_{k+1} + |y| \leq f_{k+1} + p < f_{k+1} + f_k \leq f_{k+2} \) which shows that \( f_{k+1} + |y| \) is not a Fibonacci number and hence \( xy^2 z \) does not belong to the language, and this is a contradiction.

4. (20 points) For each one of the following languages construct a context-free grammar that generates that language:

(a) \( \{0, 1\}^* \).

**Answer:**

\[ A \rightarrow 0A \mid 1A \mid \varepsilon \]

(b) \( \{0^m1^n \mid m \geq n \text{ and } m - n \text{ is even}\} \).

**Answer:**

\[ A \rightarrow 00A \mid 0A1 \mid \varepsilon \]

(c) The complement of \( \{0^n1^n \mid n \geq 0\} \) over the alphabet \( \{0, 1\} \).

**Answer:**

\[
\begin{align*}
A & \rightarrow B10B \mid 0A1 \mid C \mid D \\
B & \rightarrow 0B \mid 1B \mid \varepsilon \\
C & \rightarrow 0C \mid 0 \\
D & \rightarrow 1D \mid 1
\end{align*}
\]

(d) The set of strings in \( \{0, 1\}^* \) which are not palindromes:

\( \{w \in \{0, 1\}^* \mid w \neq w^R\} \).

**Answer:**

\[
\begin{align*}
A & \rightarrow 0A0 \mid 1A1 \mid 0B1 \mid 1B0 \\
B & \rightarrow 0B \mid 1B \mid \varepsilon
\end{align*}
\]
5. (10 points) Show that the language of the grammar \( S \rightarrow 0S1 \mid 1S0 \mid SS \mid \varepsilon \) is \( \{ w \in \{0,1\}^* \mid w \text{ contains the same number of zeros and ones} \} \).

**Answer:** Define an auxiliary function \( f \) as
\[
f(w) = (\text{number of 0's in } w) - (\text{number of 1's in } w).
\]
Note that the language in question is \( \{ w \in \{0,1\}^* \mid f(w) = 0 \} \).

One direction of the question is easy. We only prove the difficult direction which says that if \( w \) contains the same number of zeros and ones, then it can be generated by \( S \rightarrow 0S1 \mid 1S0 \mid SS \mid \varepsilon \).

We prove this by induction on the size of \( w \). The base case where \( |w| = 0 \) is trivial. *Induction hypothesis:* If a string has the same number of zeros and ones, and its length is less than \( m \), then it can be generated by \( S \rightarrow 0S1 \mid 1S0 \mid SS \mid \varepsilon \).

Consider \( w \) with the same number of zeros and ones, and \( |w| = m \). Hence \( f(w) = 0 \). Consider all possible ways of splitting \( w = xy \).

- **There is a splitting with \( f(x) = f(y) = 0 \):** In this case by induction hypothesis it is possible to generate both \( x \) and \( y \) by \( S \rightarrow 0S1 \mid 1S0 \mid SS \mid \varepsilon \). In other words \( S \Rightarrow^* x \) and \( S \Rightarrow^* y \). Then we can use \( S \Rightarrow SS \) to generate \( w = xy \).
- **\( f(x) > 0 \) for every such splitting:** In this case \( w \) starts with a 0 (taking \( x = w_1 \) shows this) and ends with a 1 (taking \( y = w_m \) shows this). So we can use \( w \Rightarrow 0S1 \) to generate \( w \).
- **\( f(x) < 0 \) for every such splitting:** In this case \( w \) starts with a 1 (taking \( x = w_1 \) shows this) and ends with a 1 (taking \( y = w_m \) shows this). So we can use \( w \Rightarrow 1S0 \) to generate \( w \).
- **Note that there are not other cases, as if \( f \) wants to change signs then we find an splitting \( w = xy \) with \( f(x) = f(y) = 0 \). This is the first case that we considered.**

6. (10 Points) Use the equivalence of context-free grammars and push-down automata to show that if \( A \) and \( B \) are regular languages, then \( \{ xy \mid x \in A, y \in B, |x| = |y| \} \) is context-free.

**Answer:** Let \( M_1 \) and \( M_2 \) be NFA’s accepting \( A \) and \( B \) respectively. We can modify \( M_1 \) and \( M_2 \), if necessary, and assume that each one of them has at most one accept state. Recall that in the class we constructed an NFA \( N \) which was accepting \( \{ xy \mid x \in A, y \in B \} \). Here we construct a PDA Which is similar to \( N \) but with the difference that in the \( M_1 \) part it pushes a symbol to the stack every time that it reads a letter from the input, and in the \( M_2 \) part it pops a letter from the input every time that it reads a letter. Before accepting we make sure that the stack is empty.
7. (10 Points) Let $L$ be an infinite regular language over the single letter alphabet $\Sigma = \{0\}$. For every integer $m$, let

$$L_m = \{ w \in L \mid |w| \leq m \}$$

be the set of the strings of length at most $m$ in $L$. Show that there is a real number $c > 0$ and an integer $M > 0$ such that for every $m \geq M$, we have $\frac{|L_m|}{m} > c$.

**Answer:** Let $p$ be the pumping constant of $L$. Since $L$ is infinite, there is a word $w$ with $|w| > p$. Then by pumping lemma we can split $w = xyz$ such that

1. $|y| \leq |xy| \leq p$.
2. $|y| > 0$.
3. $xy^iz \in L$ for every $i \geq 0$.

Note that $xy^iz = 0^{|w|+(i-1)|y|}$. So the words $0^{|w|}, 0^{|w|+|y|}, 0^{|w|+2|y|}, 0^{|w|+3|y|}, \ldots$ all belong to $L$. Set $M = 2|w|$ and note that since $|y| \leq p$, for every $m \geq M$, we have

$$|L_m| \geq \frac{m - |w|}{p} \geq \frac{m - (m/2)}{p} \geq \frac{m}{2p}.$$ 

Hence we can take $c = 2p$.

8. For a positive integer $m$, a language $L$ over $\{0, 1\}$ is called $m$-bounded, if the length of every word in $L$ is at most $m$.

(a) (5 Points) How many $m$-bounded languages are there?

**Answer:** For every $k \geq 0$, there are exactly $2^k$ words of length exactly $k$. Hence there are $1 + 2 + 4 + \ldots + 2^m = 2^{m+1} - 1$ words of length at most $m$. Now each one of these words either belong or do not belong to an $m$-bounded language. So there are $2^{2^m+1} - 1$ such languages.

(b) (15 Points) For $m \geq 100$, show that for more than half of the $2^m$-bounded languages, there is no DFA with $2^m$ states that recognizes them. [Hint: Count the number of DFA’s with $2^m$ states.]

**Answer:** First we count the number of DFA’s $M = (Q, \Sigma, \delta, q_0, F)$ with $2^m$ states. There are $2^m$ choices for picking the start state $q_0$. There are $2^m$ choices for picking the set of accept states $F$. Now we count the number of choices for picking the transition $\delta : Q \times \Sigma \rightarrow Q$. For each one of the $2^{m+1}$ elements in $Q \times \Sigma$, we have to pick an element in $Q$. So there are $|Q|^{2^{m+1}} = 2^{2m^2+1}$ choices for $\delta$. Multiplying these numbers, we get that the number of such DFA’s is

$$2^m \times 2^m \times 2^{2m^2+1} \leq 2^{m+2m^2} \leq 2^{4m^2}.$$ 

By Part (a) there are $2^{2^{2m+1}-1} > 2^{2m}$ languages that are $2m$-bounded. So it suffices to prove that for $m \geq 100$,

$$2 \times 2^{4m^2} = 2^{1+4m^2} < 2^{2m}.$$ 

Equivalently

$$1 + m2^{m+2} < 2^m$$

which is straightforward to prove.