Problem 1

Consider the following regular expression over the alphabet $\Sigma = \{0, 1\}$:

$$(00 \cup 10)^* (11)^*$$

Construct a DFA that recognizes the language described by this regular expression. (Hint: first construct an NFA, then convert it to a DFA.)

Problem 2

Consider the language $A$ over the alphabet $\Sigma = \{0, 1\}$ given by $A = \{w \mid$ every odd position of $w$ is a 1}. (In a string, the first symbol is position 1.)

a. Design a DFA that recognizes language $A$.

b. Use the procedure described in Lemma 1.60 in the textbook (which was also covered in class in lecture 4) to convert the DFA you designed in part (a) to a regular expression for $A$. Show your steps.

Problem 3

Feline internet sensation Grumpy Cat is well known for her hatred of regular languages. All of these regular expressions and DFAs have made her very, very grumpy. Please help improve Grumpy Cat’s mood by using the pumping lemma to show her that the following languages are not regular.
a. 
\[ L_1 = \{0^n1^n2^n \mid n \geq 0\} \]

b. 
\[ L_2 = \{0^i1^j \mid i > j\} \]

The following questions are lab problems.

Lab Problem 1

Recall that a **pumping length** for a language \( A \) is a positive integer \( p \) such that all strings \( s \in A \) with \( |s| \geq p \) can be written in the form \( s = xyz \), where

(i) \( |xy| \leq p \),

(ii) \( |y| \geq 1 \), and

(iii) \( xy^iz \in A \) for all \( i \geq 0 \).

Note that if \( A \) is finite with its longest string of length \( \ell \), then \( p = \ell + 1 \) is a valid pumping length for \( A \) because there are no strings \( s \in A \) with \( |s| \geq \ell + 1 \). Then it is vacuously true that all such strings satisfy the three conditions above.

The **pumping lemma** states that every regular language has a pumping length. If a language \( A \) has a pumping length, the **minimum pumping length** of \( A \), \( p_{\text{min}} \), is the smallest pumping length for \( A \). Note that this implies that every integer \( p \geq p_{\text{min}} \) is also a valid pumping length for \( A \).

For example, if \( A = ab^* \), the minimum pumping length is 2. To show that 1 is not a pumping length, note that the string \( s = a \) is in \( A \) but cannot be pumped. Writing \( s = xyz \), in order to satisfy condition (ii) in the definition of pumping length, we must have \( x = \varepsilon \), \( y = a \), and \( z = \varepsilon \). But then \( xz \) is not in \( A \), violating condition (iii). This shows that 1 is not a pumping length. However, 2 is a pumping length because for any string \( |s| \geq 2 \), we can take \( x = a \), \( y = b \), and \( z \) to be everything else, so that we have \( |xy| \leq 2 \), \( |y| \geq 1 \), and \( xy^iz \in A \) for all \( i \geq 0 \).

For each of the following languages, give the minimum pumping length \( p_{\text{min}} \) and prove your answer.
a. $bbba^*$
b. $(ab)^*$
c. $a^*ab^*ab^*$
d. $L_d = \{w \in \{a,b\}^* \text{ such that } w \text{ ends in } ab\}$
e. $L_e = \{ab, bb, abababab, aabb\}$

**Lab Problem 2**

Consider the language $F$ defined over the alphabet $\Sigma = \{a, b, c\}$ as follows:

$F = \{a^iv \mid i \geq 0, \ v \text{ is a sequence of } b\text{'s and } c\text{'s, and if } i = 1 \text{ then } v \text{ is palindromic: it is the same read backwards as forwards}\}.$

a. Show that $F$ is not regular.

b. Give a pumping length $p$ and demonstrate that, for all strings $w \in F$ such that $|w| \geq p$, we can write $w = xyz$ with $|xy| \leq p$ and $|y| \geq 1$ such that $xy^iz \in F$ for all $i \geq 0$. In other words, demonstrate that the pumping lemma is not helpful for proving $F$ is not regular.

c. Explain why parts (a) and (b) do not contradict the pumping lemma.

**Lab Problem 3**

Regular languages are among Nyan Cat’s favorite things, and all this talk about the pumping lemma and non-regular languages has made it unhappy. In order to try to cheer it up, you try to come up with ways to obtain new regular languages from old ones, in addition to the standard regular operations.

Given two languages $A$ and $B$ over the same alphabet $\Sigma$, define the following operations:

- The prefix closure of language $A$ is given by

$$PC(A) = \{x \mid xy \in A \text{ for some } y \in \Sigma^*\}.$$
• The *avoids* operation for languages $A$ and $B$ is defined

$$A \text{ avoids } B = \{ w \mid w \in A \text{ and } w \text{ doesn’t contain any string in } B \text{ as a contiguous substring} \}.$$ 

• The *halving* of language $A$ is defined

$$A_{\frac{1}{2}} = \{ x \mid \text{for some } y \in \Sigma^*, |x| = |y| \text{ and } xy \in A \}.$$ 

Show that the class of regular languages is closed under each of these three operations, i.e., that if $A$ and $B$ are regular languages, then so are $PC(A)$, $A$ avoids $B$, and $A_{\frac{1}{2}}$. 