Problem 1 (160 pts).

Consider a language $L_1 \subseteq \{0, 1\}^*$ where each unique string represents level i of a full binary tree, as viewed from left to right, for $i \geq 0$, and in the following manner:

- Each left child is represented by a 0.
- Each right child is represented by a 1.

Note that we consider level 0 to be the root node and exclude it. Use the Pumping Lemma to show that L_1 is not regular.

Problem 2 (160 pts).

Describe $L(G_1)$ for the following CFG G_1 on $\Sigma = \{0, 1\}$.

$$S \to S0S0S1S1S \mid S0S1S0S1S \mid S0S1S1S0S \mid S1S0S0S1S \mid S1S0S1S0S \mid S1S1S0S0S \mid \varepsilon$$

Problem 3 (160 pts).

Describe $L(G_2)$ for the following CFG G_2 on $\Sigma = \{0, 1\}, 2\}$:

$$\begin{aligned} S &\to 0T \mid 1S \mid 2S \\ T &\to 0T \mid 1U \mid 2S \\ U &\to 0U \mid 1U \mid 2U \mid \varepsilon \end{aligned}$$

Problem 4 (160 pts).

Let $\Sigma = \{0, 1, .., k\}$ for some $k > 0$.

- Describe how you would design a CFG to accept $L_2 = \{ww^r : w \in \Sigma^*\}$
- Describe how you would design a CFG to accept $L_3 = \{w = w^r : w \in \Sigma^*\}$

Problem 5 (160 pts)**.**

Recall the Tower of Hanoi puzzle, which involves a set of n disks of increasing diameter, which fit on three pegs, and has the following rules:

- No larger disk may be place on top of a smaller disk.
- Only one disk may be moved at a time.
- All disks are initially placed on one peg (in increasing order of size from top to bottom of course).

Also recall that the objective of the game is to move the entire stack of disks to another peg.

Now suppose we start with the tower on the leftmost peg and wish to move it to the rightmost peg. Using the criteria of the n-disk puzzle as a guideline, define a set of terminals so that the movement of each disk can be uniquely determined via a single terminal, and then provide a grammar on those terminals which generates the solution to the n-disk puzzle.