CSE 105 Midterm Solution

Problem 1. Write a regular expression for the language of strings in \( \{0, 1\}^* \) that do contain any contiguous string of 1’s of length greater than 2.

Solution: The idea here is to avoid the occurrence of substrings of 111. So, in general, we may have several 0’s or a 1 followed by a several 0’s or two 1’s followed by a several 0’s. Of course, we also have to deal with the boundary cases. Using this idea, we can write the following regular expression for the given language:

\[(0 \cup 10 \cup 110)^* (\epsilon \cup 1 \cup 11)\]

Problem 2. Design a finite automaton for the language of all strings in \( \{0, 1\}^* \) such that the difference between the number of 1’s and 0’s is not divisible by 5.

Solution: In order to construct a DFA for this language, we need to keep track of the difference between the number of 1’s and 0’s mod 5. Hence, we need a total of 5 states. Depending on which symbol we see at the input, we should move either clockwise or counter-clockwise in the diagram. Figure 1 gives the DFA for this language.

Figure 1: DFA for the language of all strings for which the difference between the number of 1’s and 0’s is not divisible by 5.
**Problem 3.** Show that the language \( L = \{1^l0^m1^n \mid n \geq l + m, \text{ and } l, m, n \geq 0\} \) is not regular. Provide a complete proof using pumping lemma.

**Solution:** We will proceed by contradiction. Assuming \( L \) is regular, we can apply the pumping lemma to any string whose length is at least \( p \), the pumping length. Let \( s = 0^p1^p \). Hence \( s \) can be broken into three parts, \( s = xyz \), satisfying the following conditions:

1. for each \( i \geq 0, xy^iz \in E; \)
2. \( |y| > 0; \) and
3. \( |xy| \leq p. \)

Because \( |xy| \leq p \) and \( |y| > 0, y = 0^j \) for some \( 0 < j \leq p \). If we pump up once, the resulting string \( xy^2z = 0^p + j1^p \) is not in \( L \), contradicting the pumping lemma. Therefore, \( L \) cannot be regular.

**Problem 4.** Convert the following NFA into an equivalent DFA. \( \Sigma = \{0, 1\} \).

![NFA Diagram]

**Solution:** An equivalent DFA (without optimization) for the NFA given above is the following:

![DFA Diagram]