Homework 2
CS611: Theory of Computation

Problem 1. [Category: Design] Given language $L = \{w|w \text{ ends with } 000\}$, **[10 points]**

1. Design a NFA for language L with 4 states, you can just draw the diagram.
2. Convert the NFA to DFA using the approach we discussed in class, you need to use the states from your NFA to represent the states of this DFA, and draw the diagram of this DFA.

Problem 2. [Category: Design] Given the language $L = \{w \in \{0,1\}^* \mid \text{number of 0s in } w \text{ is at least 2 or the number of 1s exactly 2}\}$. **[10 points]**

1. Design a NFA for You can just draw the diagram.
2. Give the regular expression for this language.

Problem 3. [Category: Design] Give regular expressions that generate each of the following languages. In all cases, the alphabet is $\Sigma = \{0, 1\}$. Each regular language has infinitely many correct regular expressions, but you only need to give one. **[15 points]**

1. The language $\{w \in \Sigma^* \mid w \text{ has an odd number of 0s}\}$.
2. The language $\{w \in \Sigma^* \mid w \text{ ends in a double letter}\}$. (A string contians a *double letter* if it contains 00 or 11 as a substring.)
3. The language $\{w \in \Sigma^* \mid w \text{ contains exactly one double letter}\}$. For example, 10010 has exactly one double letter, but 100010 has two double letters.

Problem 4. [Category: Comprehension+Design]

1. Describe the language of the following regular expressions. A clear, crisp one-level interpretable English description is acceptable, like "This is the set of all binary strings with at least three 0s and at most hundred 1s", or like "$\{0^n(10)^m \mid n$ and m are integers$\}$". A vague, recursive or multi-level-interpretable description is not, like "This is a set of binary strings that starts and ends in 1, and the rest of the string starts and ends in 0, and the remainder of the string is a smaller string of the same form!" or "This is a set of strings like 010, 00100, 0001000, and so on!". You need not prove the correctness of your answer.
   (a) $(0^* \cup 0 \cup 1^*)^*$ **[1 points]**
   (b) $0(10)^*1$ **[2 points]**
   (c) $1^*(0 \cup 111^*)^*1^*$ **[2 points]**
2. Give regular expressions that accurately describe the following languages. You need not prove the correctness of your answer.

(a) All binary strings with no more than three 0s. **[1 points]**

(b) All binary strings such that in every prefix, the number of 0s and 1s differ by at most 1. **[2 points]**

(c) All binary strings with exactly one occurrence of the substring 000. **[2 points]**