V22.0453-001: Honors Theory of Computation

Problem Set 1 Solutions

Problem 4  Give a regular expression for each of the following languages.

1. \{w : The length of w is a multiple of 3\}.
   Solution: \((\Sigma \Sigma \Sigma)^*\)

2. \{w : w either starts with 01 or ends with 10\}.
   Solution: \((01\Sigma^*) \cup (\Sigma^*10)\)

3. \{w : w does not contain the substring 001\}.
   Solution: \((1 \cup 01)^*0^*\)

Problem 6

The procedure for converting an NFA to an equivalent DFA given in class yields an exponential blowup in the number of states. That is, if the original NFA has \(n\) states, then the resulting DFA has \(2^n\) states. In this problem, you will show that such an exponential blowup is necessary in the worst case.

Define \(L_n = \{w : \text{The } n\text{th symbol from the right is 1}\}\).

1. Give an NFA with \(n + 1\) states that recognizes \(L_n\).

2. Prove that any DFA with fewer than \(2^n\) states cannot recognize \(L_n\). (Hint: Let \(M\) be any DFA with fewer than \(2^n\) states. Start by showing that there exist two different strings of length \(n\) that drive \(M\) to the same state.)

   Proof: Let \(M\) be any DFA with fewer than \(2^n\) states. We will show that \(M\) cannot recognize \(L_n\). Since there are \(2^n\) strings of length \(n\), by the Pigeonhole Principle, there are two different strings \(x = x_1x_2 \cdots x_n\) and \(y = y_1y_2 \cdots y_n\) that drive \(M\) to the same state. Since \(x \neq y\), there is some \(i\) such that \(x_i \neq y_i\). Without loss of generality, say that \(x_i = 1\) and \(y_i = 0\). Let \(x' = x_0^{i-1}\) and \(y' = y_0^{i-1}\). It is easy to see that \(x' \in L_n\) and \(y' \notin L_n\). However, since \(x\) and \(y\) drive \(M\) to the same state, it is clear that \(x' = x_0^{i-1}\) and \(y' = y_0^{i-1}\) also drive \(M\) to the same state, yet \(x' \in L_n\) and \(y' \notin L_n\). Therefore \(M\) cannot recognize \(L_n\).