

COL352 Office Hours 2

Question 1

Is there a class of regular language such that for any language in this class, the number of states in any DFA recognizing the language is at least exponential in the number of states in the smallest NFA recognizing the language?

Answer. Let's try to construct such a language. The easiest way of showing that the number of states in a DFA is at least something is using the pigeonhole principle (as done in the quiz). So we need to try to construct a language which has a parameter k , $O(k)$ states in the NFA, and an exponential number of states in any DFA. To get to the pigeonhole argument, we'll need to show that for any string of length $O(k)$, the run ends in a different state. Consider any two distinct length k strings x, y . Then suppose their runs end in the same state. We need some sort of a contradiction to fit in easily. Since these are distinct strings, there exists at least one index at which they differ. Suppose it is i . Then everything before index i is identical, and we know that the rest of the run ends on some state, but has a variable length. Let's try to make it some fixed number which is in $O(k)$. Try k . For that we'll just append some equal elements (0 or 1) to the end. Now we have two strings of length $\geq k$, whose runs end at the same state, but have the k^{th} character different. This motivates the question – what about the language L_k which has the k^{th} element from the end equal to some fixed character? If we can show this is a regular language with the smallest NFA having at most $O(k)$ states, then we'll be done. Note that it suffices to exhibit any NFA with number of states in $O(k)$. However, this is pretty simple, as done in class, and it indeed works too.

Formalizing the above answer, we can get to the following theorem:

Theorem 1

Let $\Sigma = \{0, 1\}$. Define $L_k = \{x \in \Sigma^* \mid |x| \geq k \wedge x_{|x|-k+1} = 1\}$. Then the class of languages $\mathcal{L} = \{L_k \mid k \geq 1\}$ is such a class.

Proof. Consider the following NFA:

$$\mathcal{N} = (Q, \Sigma, \Delta, Q_0, A)$$

Where

1. Q is a set of $k + 1$ states q_i .
2. $A = \{q_k\}$.
3. $Q_0 = \{q_0\}$.
4. $\Delta = \{(q_i, a, q_{i+1}) \mid 1 \leq i < k, a \in \Sigma\} \cup \{(q_0, 0, q_0), (q_0, 1, q_1), (q_0, 1, q_1)\}$.

This is clearly a working NFA for the language L_k . Consider any DFA D that accepts L_k .

Claim 1.1

For $x \neq y$ and $|x| = |y| = k$, we have $\hat{\delta}(q_0, x) \neq \hat{\delta}(q_0, y)$.

Proof. Suppose this is false, and that $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y) = q$ (say). Consider the first location where x and y differ, say i . Then consider the strings $x \cdot x_1 \cdots x_{i-1}$ and $y \cdot y_1 \cdots y_{i-1}$. Clearly, the k^{th} characters of both the strings are different, so exactly one string is in L_k . However, note that by the definition of i , we have $x_1 \cdots x_{i-1} = y_1 \cdots y_{i-1}$, so $\hat{\delta}(q_0, x \cdot x_1 \cdots x_{i-1}) = \hat{\delta}(\hat{\delta}(q_0, x), x_1 \cdots x_{i-1}) = \hat{\delta}(q, x_1 \cdots x_{i-1}) = \hat{\delta}(q, y_1 \cdots y_{i-1}) = \hat{\delta}(\hat{\delta}(q_0, y), y_1 \cdots y_{i-1}) = \hat{\delta}(q_0, y \cdot y_1 \cdots y_{i-1})$. This implies that either both strings are

accepted by the DFA or they aren't, which is a contradiction. \square

Now suppose there is a DFA which has $< 2^k$ states that accepts L_k . By the pigeonhole principle, there must exist two distinct strings $x, y \in \Sigma^k$ where $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$. By the previous claim, this is a contradiction. Hence, any DFA recognizing L_k must have at least 2^k states. This implies $|Q_D| = 2^{|Q_N|-1} \geq 2^{|Q_M|-1}$, where M is the minimal NFA that recognizes L_k . \square