Introduction to Theory of Computing	**Problem Set 3**
CS 481 Fall 2004	**Due Wednesday, September 22, 2004**

Reading. To review the material covered by this problem set, read sections 11 and 12 of Kozen (we will cover Pattern matching and regular expression during the coming week).

For each problem set please write both your name and your cornell.edu email address on top.

(1) Show that the following sets are not regular. In solving each problem, you may assume that all previous parts are solved, even if you do not have a solution for some part.

(a) $\{a^n b^m |$ such that $n \leq 2m\}$

(b) $\{a^n baba^n$ such that $n \geq 0\}$

(c) $\{w \in \{0,1\}^* |$ such that $w^R = w\}$, where w^R denotes string obtained by reversing w.

(d) $\{a^{n^3} |$ such that $n \geq 0$ integer$\}$

(e) The set PAREN of balanced strings of parentheses ( ). For example, the string ((( ) ( )) ( )) is in PAREN but the string) (( ) is not.

(2) Miscellaneous Exercise 31, p. 323. (See the hint at page 352.)

(3) Last problem set you gave a NFA for the following language. Let

$$Incomplete(\Sigma) = \{w \in \Sigma^* | \exists a \in \Sigma \text{ such that w does not contain a}\}.$$

We showed that if Σ has k characters, then there is a NFA accepting $Incomplete(\Sigma)$ with k states. Show that all DFA's accepting $Incomplete(\Sigma)$ have at least 2^k states.

As a hint, you may want to review the proof from class printed on the back.

Let $L = \Sigma^*1\Sigma^{k-1}$ for $\Sigma = \{0,1\}$. We showed in class that any DFA accepting L must have at least 2^k states. Here we review the proof.

The proof is by contradiction. Let $M = (Q, \Sigma, \delta, s, F)$ be a DFA accepting L with less than 2^k states. Consider all strings in Σ^k. There are 2^k such strings, and less than 2^k states, so there must be two strings $x, y \in \Sigma^k$ such that $x \neq y$ and $\hat{\delta}(s,x) = \hat{\delta}(s,y)$. By definition of a DFA, we get the following fact.

Fact *If $\hat{\delta}(s,x) = \hat{\delta}(s,y)$, then for all $w \in \Sigma^*$ we have $\hat{\delta}(s,xw) = \hat{\delta}(s,yw)$.*

Proof. By problem 1 on the 1st problem set, and the assumption, we get the following chain of equations

$$\hat{\delta}(s,xw) = \hat{\delta}(\hat{\delta}(s,x),w) = \hat{\delta}(\hat{\delta}(s,y),w) = \hat{\delta}(s,yw).$$

Now assume x and y differ at the jth letter (and maybe also at some other letters), and assume that x has a 0, while y has a 1 in this position. Let $w \in \Sigma^{j-1}$ be any string. Now we have that $xw \notin L$ while $yw \in L$, and by the above fact $\hat{\delta}(s,xw) = \hat{\delta}(s,yw)$. Let $q = \hat{\delta}(s,xw) = \hat{\delta}(s,yw)$. Now we get the desired contradiction as follows: If q is an accepting state, then M accepts xw which is wrong; while if q is not an accepting state, then M does not accept yw which is wrong. Hence M does not accept L in either case.