CSE 322
Winter Quarter 2003
Assignment 5
Due Friday, February 14, 2003

All solutions should be neatly written or type set. All major steps in proofs and algorithms must be justified.

1. (10 points) In this problem you will apply the pumping lemma to show that a language is not regular. Consider the language $P = \{0^n : n \text{ is prime}\}$. So $P = \{0^2, 0^3, 0^5, 0^7, \ldots\}$. Use the pumping lemma to show that P is not regular.

2. (10 points) In this problem you will show that regular languages are closed under some new operations.

   (a) Define $\text{init}(L) = \{x : \text{ for some } y, xy \in L\}$. Show that if L is regular then so is $\text{init}(L)$. One approach is to start with a DFA for L and show how to modify it to accept $\text{init}(L)$.

   (b) Define $\text{mid}(L) = \{y : \text{ for some } x \text{ and } z, xyz \in L\}$. Show that if L is regular then so is $\text{mid}(L)$. One approach is to start with a DFA for L and show how to construct an ϵ-NFA to accept $\text{mid}(L)$.

3. (10 points) In this problem we'll explore the minimum state DFA. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts L. For $q, p \in Q$ define $q \sim p$ if for all $x \in \Sigma^*$, $\hat{\delta}(q, x) \in F$ if and only if $\hat{\delta}(p, x) \in F$. That is, $q \sim p$ if whether or not you start in state p or q in consuming a string, you either accept both or not accept both. It turns out that the relation $\sim$ is an equivalence relation, that is reflexive, symmetric, and transitive. For $q \in Q$, define $[q] = \{p \in Q : p \sim q\}$ to be the equivalence class of q. We can now define, $M' = (Q', \Sigma, \delta', q_0', F')$ where

$$\begin{aligned} Q' &= \{[q] : q \in Q\} \\ q_0' &= [q_0] \\ F' &= \{[q] : q \in F\} \\ \delta'([q], a) &= [\delta(q, a)] \end{aligned}$$

   Note that the states of M' are subsets of the states of M. It turns out that M' is the unique minimal state DFA that accepts L. There is a nice closure algorithm for contructing M'. The algorithm generates all unordered pairs $\{p, q\}$ such that $p \not\sim q$.

```
Algorithm to find all pairs of nonequivalent states:
X := {{p,q} : p in F and q not in F}
  //final and nonfinal states are not equivalent;
repeat
  X' := X;
  X := X union {{p,q} : for some a in Sigma, {delta(p,a),delta(q,a)} in X'};
    //if two states transition on the same symbol to nonequivalent
    //states then they are also nonequivalent
until X' = X
```

In the end $p \sim q$ if and only if $\{p, q\} \notin X$. The table filling algorithm on page 156 of the book is essentially this closure algorithm. Consider the following DFA. Use the closure algorithm above to find the minimum state finite automaton equivalent to it. Feel free to use a table to assist you.

		0	1
$>$	A	C	B
	B	D	A
	C	F	D
	D	G	E
	E	F	C
$*$	F	A	G
$*$	G	B	F

Note that $>$ indicates the start state and $*$ the final states. Show the final X of the closure algorithm and give a minimum state DFA using a transition table base on this algorithm.