CSE 322
Winter Quarter 2001
Assignment 1
Due Friday, January 12

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

1. (10 points) This problem is designed to strengthen your ability to prove facts by induction. The reversal w^R of a string w can be defined recursively in the following way.

$$\begin{aligned} \varepsilon^R &= \varepsilon \\ (xa)^R &= ax^R \end{aligned}$$

   where $a \in \Sigma$.

   Prove the following:

   (a) For $a \in \Sigma$, $a^R = a$.

   (b) For x and y strings over Σ, $(xy)^R = y^R x^R$. For this your proof should be by induction on the length of y. You may use recursive definition of reversal and any basic facts about concatenations such as associativity and the identity properties of ε.

2. (10 points) This problem is designed to build the background for analysis of finite automata. You may have heard of the pigeon hole principle before.

   Pigeon Hole Principle: If $f : S \to T$ and S has more elements than T, then there are members x and y of S such that $x \neq y$ and $f(x) = f(y)$. Put another way, if there are n pigeons that are living in m coops and $n > m$ then there are at least two pigeons that live in the same coop.

   Suppose we have a directed graph $G = (V, E)$ where V has cardinality n. A path in G is a sequence $(v_0, v_1, \ldots, v_k)$ of vertices such that for $0 \leq i < k$, (v_i, v_{i+1}) is in E. For each i we say that v_i is visited on the path $(v_0, v_1, \ldots, v_k)$. The path $(v_0, v_1, \ldots, v_k)$ is of length k, which is the number of edges traversed on the path.

   Prove that every path of length n or longer visits some vertex at least twice.

3. (15 points) This problem is designed to help you think more abstractly about algorithms that will be useful in the analysis of finite automata. Given a directed graph $G = (V, E)$ and a vertex $v \in V$ define the set of vertices reachable from v as follows:

$$R(v) = \{v_k \ : \ v_0 = v \text{ and } (v_0, v_1, \ldots, v_k) \text{ is a path in } G \text{ for some } v_0, \ldots, v_{k-1}\}.$$

   Notice that $v \in R(v)$ by letting $k = 0$ in the definition. Consider the following algorithm for computing $R(v)$.

```
X = {v};
repeat
  X' = X;
  X = X' union {y : (x,y) in E and x in X'};
until X = X'
R(v) = X
```

(a) Consider the graph $G = (V, E)$ with $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (1, 3), (2, 4), (3, 2), (4, 3)\}$. Run the algorithm for $R(1)$, $R(2)$, $R(3)$ and $R(4)$ showing the result after each iteration of the repeat loop.

(b) If G has n vertices then what is the maximum number of times the repeat loop can be executed?

(c) Modify the algorithm for $R(v)$ to compute $R^+(v)$ which is the set of vertices reachable from v with a path of length at least 1, that is,

$$R^+(v) = \{v_k \ : \ v_0 = v \text{ and } (v_0, v_1, \ldots, v_k) \text{ is a path in } G \text{ for some } v_0, \ldots, v_{k-1} \text{ and } k > 0\}.$$

Note that if $v \in R^+(v)$ then there is a cycle in G which includes v. Recall that a cycle is a path whose first vertex matches last vertex.

(d) Use R and R^+ to compute for a pair of vertices u and v whether or not there are infinitely many paths from u to v. For this problem you should give an algorithm, that calls R and R^+ as subroutines, for determining for a given u and v if there are infinitely many paths from u to v. You should also explain why your algorithm is correct.