## Problem 1:

Apply the state minimization algorithm from the lectures to the FSM below. Write out the groups of states that you begin with as a sequence of sets of states. At each step, say which symbol and which group of states you are considering and how this splits the groups of states. Show how all the states are grouped after each step. When you have finished, draw the diagram for the resulting minimized FSM.

## Problem 2:



Apply the construction given in lecture to convert the NFA below to a DFA that recognizes exactly the same language.

## Problem 3:



Draw NFAs that recognize the languages described by each of the following regular expressions. Use the construction given in class or in the book or produce something simpler if you can.
a) $\left(1(10)^{*}\right)^{*} \cup 01^{*}$
b) $\left(1^{*} 0 \cup 1\right)^{*} \cup\left(0 \cup 0^{*} 1\right)^{*}$

## Problem 4:

Use the method given in class to design a linear-time algorithm to determine all occurrences of the string 01011010111 in strings over the alphabet $\{0,1\}$.

## Problem 5:

Prove that the set of all binary strings with fewer 0's than 1's is not recognized by any DFA.

## Problem 6:

An integer quadratic polynomial is a function of the form $f(x)=a x^{2}+b x+c$ where $a \neq 0, b$, and $c$ are integers. Show that the set of all integer quadratic polynomials is countable.

## Problem 7:

Let $B$ be the set of all infinite binary sequences. Show that $B$ is uncountable using a proof by diagonalization. (Note that infinite binary sequences are not strings since any string has finite length.)

## Problem 8:

[This problem will not be graded.] Show that the following problem is undecidable using the fact that the halting problem is undecidable: Given the code $\langle P\rangle$ of a Java program $P$ and an input $x$ to $P$, determine whether or not $P$ ever prints out a 1 on input $x$.

## Extra Credit 9:

In this question you are to design a finite state controller for the traffic lights at an intersection of a busy two-way East-West street and a small two-way street that heads South of the E-W street but does not cross it: There is a left-turn lane for the westbound traffic on the E-W street to turn Southound and there is a sensor L that detects traffic waiting in this left-turn lane. There is also a sensor N to detect waiting Northbound traffic on the small street. We use input notation 0 to denote that neither sensor detects a car, B to denote that both sensors detect cars, and L and N to denote that only one of the sensors is activated.

There are 4 traffic lights: Eastbound, Westbound. Northbound, and Left-turn that each cycles from Green to Yellow to Red back to Green in response to input signals. (We think of these outputs as EG,EY,ER, etc.) The length of time that a Yellow occurs is governed by a timer signal T that ends it. (We won't worry about how T is activated.)

Under normal circumstances, the Eastbound and Westbound lights are Green and the Northbound and Left-turn lights are Red. If there is waiting Northbound traffic, then all other lights must turn to Yellow and then to Red before the Northbound light can turn Green. Northbound traffic is rare enough that it takes precedence over all other traffic. If there is waiting Left-turn traffic, then the Eastbound and Northbound lights must turn to Yellow and then to Red before that traffic can turn left but Westbound traffic is unaffected. Left-turn traffic takes precedence over everything but Northbound traffic. So that the lights don't switch back instantly, after the Northbound light turns Green it must stay Green so long as there is still waiting traffic and for at least two timer signals before it turns Yellow. (There is no such requirement for the length of Green at any other traffic light.)

