CSU390 Theory of Computation	Assignment #01
Fall 2006	September 12, 2006

Homework 01

Due: Tuesday, September 19, 2006

Instructions

1. Please review the homework grading policy outlined in the course information page.
2. On the *first page* of your solution write-up, you *must* make explicit which problems are to be graded for regular credit, which problems are to be graded for extra credit, and which problems you did not attempt. Use a table that looks like this:

Problem	1	2	3	4	5	6	7	8	9	...
Credit	RC	RC	RC	EC	RC	EC	NA	NA	EC	...

   where "RC" denotes "regular credit", "EC" denotes "extra credit", and "NA" denotes "not attempted". Failure to include such a table will result in an arbitrary set of problems being graded for regular credit, no problems being graded for extra credit, and a 5% penalty assessment.
3. You must also write down with whom you worked on the assignment. If this varies from problem to problem, write down this information separately with each problem.

Problems

Required: 4 of the following 5 problems
Points: 25 points per problem

1. a. Give the formal description of the FA shown in Exercise 1.21(b) (p. 86).

   b. Give the state transition diagram for the FA whose formal description is $(\{1, 2, 3, 4\}, \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}, \delta, 1, \{2, 3\})$, where δ is given by the following table:

	a	b	c
1	1	1	2
2	3	4	1
3	1	1	1
4	2	3	4

2. Do Exercise 1.6(a,b,c,d,e).
3. Do Exercise 1.6(f,g,h,i).
4. Do Exercise 1.6(j,k,l,m,n).
5. Read the informal definition of a finite state transducer (FST) given in Exercise 1.24 (p. 87). Construct a state transition diagram for an FST whose input and output alphabets are both $\{0, 1\}$ and which works as follows: The input string represents a binary number n in reverse (i.e., with least significant bit first) and the output string represents the binary number $n + 1$, also in reverse, except that any "overflow" bit is not produced (since the output string of an FST must have the same length as the input string). Technically, if the input represents the number n as a k-bit binary number, the output represents $(n + 1) \bmod 2^k$.

Here are some examples of input and corresponding output strings to illustrate the desired behavior:

In: 0010010110
Out: 1010010110

In: 11010010
Out: 00110010

In: 111
Out: 000

Note in the last example that the correct result obtained by adding 1 to the input would be 0001, but since the FST must produce an output string of the same length as the input string, the extra “overflow bit” is ignored.

In addition to giving a diagram for your FST, briefly describe what each state represents.