University of Washington
CSE 322: Introduction to Formal Models in Computer Science
Homework #2
Due: Wednesday, April 17, 2002, 10:30am

Spring 2002 April 8, 2002

Written homework is due at the *beginning* of class on the day specified. Any homework turned in after the deadline will be considered late. **Late homework policy:** You may turn in your homework after the deadline and before 5pm on the day it was due, but at a cost of a **20% penalty**. No homework will be accepted after 5pm on the due date.

Please staple all of your pages together (and order them according to the order of the problems below) and have your name on each page, just in case the pages get separated. Write legibly (or type) and organize your answers in a way that is easy to read. Neatness counts!

For each problem, make sure you have acknowledged all persons with whom you worked. Even though you are encouraged to work together on problems, the work you turn in is expected to be your own. When in doubt, invoke the *Gilligan's Island* rule (see the course organization handout) or ask the instructor.

* * *

Regular problems (to be turned in) **:**

1. Give the state diagram for a DFA M that accepts the language

   $L = \{ w \in \{0, 1\}^* \mid w$ is the binary representation of a multiple of 5$\}$.

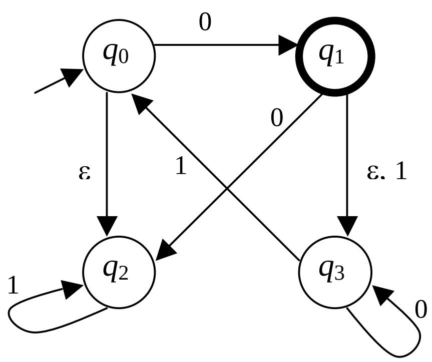
   For the purposes of this problem, you may assume that ε represents the integer 0, and that leading 0s are okay. For example, ε, 11001, and 00101 are all in L, but 110 and 00001 are not.

   Hint: Your DFA should probably have 5 states in it (let's call them q_0, q_1, q_2, q_3, and q_4). What do each of the 5 states represent? Now think about what happens if we are, say, in state q_3 after having consumed some string w and the next input symbol is a 1. What is the next state M should be in?

2. Problem 1.27. (Note that according to the errata, all of the Cs in the last two lines of the problem should be Ds.)
3. Exercise 1.5, parts b, e, and f.

(problems 4 through 6 on the next page)

4. Consider the following NFA:



Use the construction in Theorem 1.19 to convert this NFA into an equivalent DFA. ~~Do not eliminate unreachable states. (Sipser calls them unnecessary states.)~~ **New:** Use the method discussed in lecture. You may omit unreachable (unnecessary) states. Give the formal description of your machine. As a sanity check, how many states should your DFA have if you exactly follow the construction in Theorem 1.19? How many states does your simplified machine have?

5. Let $L = \{ w \in \{0, 1\}^* \mid$ the 8th symbol from the right end of the string is a 1$\}$.
   a) Give a (simple) NFA that accepts L.
   b) Give a DFA that accepts L. (Don't draw the states and transitions as a state diagram. Instead, define the transition function δ formally (and concisely).)

6. Let L be the language in problem #5 above. Prove that there is no DFA M with fewer than 256 states such that $L(M) = L$.
   Hint: Suppose such a machine M exists. If you can find two strings of the same length that differ in the 8th symbol from the right and lead to the same state in M, then you should be able to argue that M will either accept both strings or reject both strings, even though clearly one of the strings is in L and the other is not in L. You might find the pigeonhole principle useful in finding such strings.

* * *

Bonus Problem (optional):

1. Problem 1.42

* * *

Suggested problems (highly recommended, but not to be turned in) **:**

1. The rest of the exercises in 1.5.
2. Exercise 1.10.
3. Problem 1.25.
4. Problem 1.26.
5. In problem #1 (of the regular problems):
   a) Prove that your machine accepts (recognizes) L.
   b) Now suppose that instead of multiples of 5, we want to recognize strings that represent multiples of k. For $\sigma \in \{0, 1\}$, what would $\delta(q_i, \sigma)$ be? What is the initial state and the set F of accept states?
6. Practice doing the subset construction. Invent some (small – 3 or 4 states) DFA and use the subset construction to construct an NFA that accepts the same language.