Name ________________________________

Exam One

CS139 - Theory of Computing
Drake University - Spring, 2004

Directions: Do all problems. Show all work. Please work first on problems with which you are most comfortable.

Problem 1. (20 points)

Write a formal description (set theory notation) for each of the following sets. Do so in the *simplest* possible way in order to express the correct set. Let $\mathcal{N}$ denote the set of natural numbers when useful. You may use the word "divisible" in your answers (as opposed to being forced to express this more primitively).

(a) The set of prime numbers less than thirteen.

(b) The set of natural numbers divisible by 5 but not divisible by 25.

(c) The set of natural numbers equal to twice themselves.

(d) The set of binary strings containing the substring `0101`.

(e) The set of binary strings equal to the composition of two copies of themselves.

Problem 2. (15 points)

Let $\mathcal{N}$ be the set of natural numbers (i.e. positive integers). Let $f : \mathcal{N} \to \mathcal{N} \cup \{0\}$ be defined as follows. For $n \in \mathcal{N}$, let $f(n)$ be the largest non-negative integer p for which 2^p divides n. So $f(1) = 0, f(2) = 1, f(3) = 0, f(4) = 2, f(5) = 0, f(6) = 1, f(7) = 0, f(8) = 3$, etcetera. Now define a binary relation g on $\mathcal{N}$ as follows:

$$g : \mathcal{N} \times \mathcal{N} \to \{\text{true}, \text{false}\},$$

$$g(m, n) \quad = \quad \begin{cases} \text{true} & \text{if } f(m) - f(n) \text{ is divisible by } 3 \\ \text{false} & \text{otherwise} \end{cases} .$$

Carefully show that g is an equivalence relation on $\mathcal{N}$. *Hint:* This looks more complicated than it really is.

Problem 3. (15 points)

We know that the union of regular language is regular. Using this fact, give a *careful proof by induction* that every finite language is regular. To help make sure you do this properly, I recommend that you use the notation $P(n)$ to mean that "any language of finite size n is regular". Don't forget about the empty language.

Problem 4. (16 points)

Determine the language of each of following **NFA**s. Don't be too brief or too verbose. Make sure your meaning is clear. Think about your wording (and perhaps notation) before you write.

(a)

(b)

(c)

(d)

Problem 5. (16 points)

Draw a **DFA** for each of the following languages. Remember that two transitions must leave each state.

(a) Binary strings containing 101 as a substring.

(b) $\{\epsilon, 0, 00, 1\}$

(c) ϕ

(d) Binary strings of length at least two whose next to last symbol is a zero.

Problem 6. (18 points)

Use the "subset construction" to produce a **DFA** that is equivalent to the **NFA** in part (a) of Problem 4. Do not short-cut this in any way. Do the whole construction. Remember that you should have sixteen states and two transitions out of each of these states.