Name ____________________________

CS 383

Exam 2

April 20, 2007

Note: Each problem is worth 20 points.

1. Here are some languages:

   a) Strings of 0's and 1's where every 0 is followed immediately by a 1.

   b) $\{0^n | n \geq 0\}$

   c) $\{0^n 1^n | n \geq 0\}$

   d) $\{0^n 1^n 2^n | n \geq 0\}$

   e) $\{0^n 1^n 2^n 3^n | n \geq 0\}$

   f) Strings whose lengths are multiples of 3.

   g) Strings of odd length whose center element is 0.

   Mark with an R the languages that are Regular, with a C the languages that are Context Free, and with a G (for General) the languages that can be accepted by a Turing Machine. (Mark with all that apply: if a language is both Regular and Context Free mark with both R and C, etc.) You do not need to justify your answers.

   a)
   b)
   c)
   d)
   e)
   f)
   g)

2. Let $\mathcal{L}$ be the language of strings of 0's and 1's such that at least a third of the digits in each string in $\mathcal{L}$ are 0 and at least a third of the digits are 1. For example, 001111 and 1010101 are both strings in $\mathcal{L}$. Show that $\mathcal{L}$ is not context free.

3. Design a Turing Machine that accepts strings of 0's and 1's where the length of the string is odd and the center element of the string is 0. For example 00000, 11011, and 10011 are all strings in this language. You can use any kind of TM that suits you: multiple tracks, multiple tapes, etc. (My solution uses a standard TM so you don't have to use a fancy machine, but you are welcome to do anything that works.)

4. Here is a grammar $\mathcal{G}_1$:

$$S \to 0\ S\ 0 \mid 1\ S\ 1 \mid A$$
$$A \to 0 \mid 1 \mid \varepsilon$$

Here is another grammar $\mathcal{G}_2$:

$$S \to Z\ S_1 \mid Y\ S_2 \mid ZZ \mid YY \mid 0 \mid 1$$
$$S_1 \to S\ Z$$
$$S_2 \to S\ Y$$
$$Z \to 0$$
$$Y \to 1$$

a) What does it mean when we say that $\mathcal{G}_2$ is the Chomsky Normal Form for $\mathcal{G}_1$?

b) Do $\mathcal{G}_1$ and $\mathcal{G}_2$ derive the same language?

c) Why do we care about Chomsky Normal Form?

5. You have probably noticed that we didn't have a pumping lemma for languages accepted by Turing Machines (i.e., Recursively Enumerable languages). Suppose $\mathcal{L} = \left\{0^{n^2} \middle| n \geq 0\right\}$, which is certainly RE. Let w be a very long string in $\mathcal{L}$. This string contains only 0's, so it doesn't matter how many portions it is decomposed into. Could there be a portion of w that could be pumped any number of times, always producing another string in $\mathcal{L}$? Why or why not?

Please write and sign the Honor Pledge.