15-354: Computational Discrete Math — K. Sutner

Assignment 4 — Due: October 1, 2009

Problem 1: Divisibility In Reverse Binary (50 pts.)

Background

We have seen that one can check divisibility by a fixed modulus m on a DFA regardless of the base B. More precisely, we showed how to construct a DFA for words x written in base B notation with the MSD in the first position. Of course, there is no real reason why the MSD should be up front, it might as well be the last digit (reverse binary notation).

Task

1. Construct a DFA that tests divisibility by 3 for numbers written in reverse binary. To do this, find a way to express $\nu(xa)$ in terms of $\nu(x)$ and $|x|$ and choose your state set accordingly.
2. Show how to generalize this construction for arbitrary moduli m and bases B.
3. How does the size of your machine for $m = 5$ and $B = 2$ in reverse binary compare to the machine constructed in class for normal binary notation? What is going on?

Comment

Naturally one would like to know how large the minimal automata are in general, for arbitrary base B and arbitrary modulus m, and for ordinary base B notation as well as reverse base B. Lots of extra credit if you can give a general characterization.

Problem 2: Balance and Majority (50 pts.)

Background

Write $|x|_0$ for the number of 0's in a binary word x, and likewise $|x|_1$ for the number of 1's. The balance language L_{bal} is the set of all binary words that have the same number of 1's and 0's:

$$L_{\text{bal}} = \{\, x \in \mathbf{2}^\star \mid |x|_1 = |x|_0 \,\}.$$

The majority language L_{maj} is the set of all binary words that have at least as many 1's as 0's:

$$L_{\text{maj}} = \{\, x \in \mathbf{2}^\star \mid |x|_1 \geq |x|_0 \,\}.$$

Task

A. Prove that there is no finite state machine that accepts L_{bal}.

B. Prove that there is no finite state machine that accepts L_{maj}.

Comment

The essential issue is that in order to recognize these languages a DFA would require some kind of unbounding counting ability – and DFAs can only count in a bounded way. Try to come up with a clean, concise argument. Also note that your argument has to cover all possible finite state machines, you cannot make any assumptions about, say, the size of the machine.