ASSIGNMENT 7: TURING MACHINES CONTINUED

MICHAEL BEESON

1. (Alphabet size doesn't matter.) Show that for any alphabet size between 2 (counting blank) and 128, the same functions are computed. Hint: write out the ascii code of the symbol using 8 squares, thus reducing the alphabet size to 2. Our definition of Turing machine fixed the maximum size of the alphabet, but this technique would work with any larger fixed maximum just as well. Hence, we can always assume that a Turing computable function is computed by a machine that does not use all the possible symbols.

2. (two-tape machines) An two-tape machine has two tapes, and at any moment is scanning one square on each tape. The next move depends on the state and on the two symbols being scanned, and can move both read heads (i.e. changed the scanned square on both tapes). Show that such machines cannot compute any more functions than one-tape machines. Hints: Simulate a two-tape machine by a one-tape machine, using the even-numbered squares for one tape and the odd-numbered squares for the other tape. Keep track of the two simulated head locations by coloring those two squares red. ("Coloring" can be officially kept track of by one bit of the state name or number.)

3. (variables) A Turing machine with variables has a fixed list of variables, and is allowed, in each step, to assign new 8-bit values to those variables and the instructions are allowed to make use of those variables. Such machines can be simulated by an ordinary Turing machine, using more states. You do not have to write out a proof of that, but show that you understand the idea by answering this question: if the machine to be simulated has n states and m variables, how many states will the simulating (ordinary) Turing machine need? What if some of the variables are known to need only 2 values?