Stroke Arithmetic TMs

If we think of a turing machine as a primitive device, why not use a primitive encoding for numbers by representing the non-negative integer, n, with n strokes, *n* = /.../. Build turing machines to add, subtract, and multiply written in stroke notation. Below, we use the notation *i* to represent i strokes, /.../.

We start with a stroke addition machine that accepts input in the form, for example, *i*+*j*=, and returns the original tape with the sum appended to the right, ///+//=/////. This stroke addition machine is built in class and the students are given an assignment of constructing a stroke subtraction machine that accepts input in the form, *i*-*j*=, and return:

- *i*-*j*=<*stroke answer*>, if i>=j.
- i-j=-(j-i), otherwise. For example //-///=-/.

The addition and subtraction examples focus on marking and changing the relative positions of the markers written on the tape and used by the states to keep track of the progress made in completing the task.

In the next class we discuss the student variations of the stoke subtraction machine before continuing on with a stroke multiply machine. The general operation of the stoke multiple machine is discussed in class focusing on what multiplication really is, namely, continued addition. Normally I give students about a week to construct their versions of a stroke multiply machine. After the multiply machines are submitted we spend a class reviewing and comparing the machines submitted by the students.

We discuss the timings for all these machines by reviewing the number of time the representations of the two numbers are scanned in the various processes.

MULTITAPE TM

Take a second look at the above constructing three tape turing machines to perform add, subtract and multiply machines.

FUTURE DEVELOPMENT

1. The jFLAP8 turing sub-machines may provide us an opportunity to construct a stroke machine that searches for the unique input symbol, (+-*) and “call” the appropriate stroke machine.
2. I have hesitated including a divide machine, i/j=. Perhaps I should consider this a team presentation.