Turing Machines – Part 2

EXTRA CREDIT (optional)

Due by class time Friday, April 7

1. Modify your machine which scans for **xx** (from Exercise #13 of the previous worksheet) so that it scans until it finds *three* **x**'s in a row. When it finds a group of three consecutive **x**'s, it should halt. For an added touch, have the machine move back to the first of the three **x**'s and halt there. Make sure your machine works for any block of **0**'s, **1**'s, and **x**'s, such as **0011x01xx1011x111xx00xxx00**.

2. Try out the machine Decimal Add1 in **other-TMs.txt**. This machine takes a string of digits as input, representing a number in decimal (base 10) notation, and adds 1 to the number, performing any carries as needed. For example, **19** gets transformed to **20**, and **999** gets transformed to **1000**. Try a few other inputs as well. We can easily transform this machine into a decimal "counter" by modifying the last rule so that, instead of halting after adding 1, it returns to the starting state *s1*. Modify the rule in this way and then observe the behavior of the machine on the input **0**.

3. The machine Binary Add1 is just like Decimal Add1, except that it adds 1 to a number written in *binary* (base 2) notation. For example, **11** (the binary representation of 3) gets transformed to **100** (the binary representation of 4), and **1111** (15) gets transformed to **10000** (16). Try out a few other inputs as well. We can turn this machine into a binary "counter" in the same way as before, by having the machine return to state *s1* instead of halting. Modify the machine in this way and then run it on the input **0**.

4. Construct a Turing machine to do the following. Assume that the machine is started on a tape that contains nothing but a string of **$**'s. The machine is started on the left end of this string. The purpose of the machine is to multiply the length of the string by 3. For example, if given a string of seven **$**'s, it should halt with twenty-one **$**'s on the tape. If it is started on a string that contains just one **$**, it should halt with three **$**'s on the tape. Here is one possible way (but not the only way!) that the machine might accomplish this task: Change one of the **$**'s to an **x**, then go to the end of the string and write two more **x**'s. Go back and process the next **$** in the same way. Continue until all the **$**'s have been processed. Then change all the **x**'s to **$**'s.