

CS 30 Lab 7 – Turing Machines

This lab lets you experiment with Turing machines. Remember that it's fine to talk things over with your labmates, and you're encouraged to call me or Dan over for help or answers to questions.

1. Go to the CS 30 web page, choose *Links*, and click on [Turing Machine simulator from *The Analytical Engine*](#) (under Other Links).
2. Read through the instructions and play around with the simulator until you're comfortable with it. To get you started, three Turing machines called Inverter, Eraser, and Looper are shown below, along with some sample input strings for each. Load the rules for each of these machines into the simulator and try them out on the sample inputs. Enter an input string in the window called Input Tape, press the Set Tape button, press the RESET button, then try STEP (to single-step the machine) or PLAY (to run the machine to completion). IMPORTANT: Remember to press RESET at the beginning of each new run.

Inverter

1 0 1 1 R
1 1 1 0 R
1 b 2 b R

Eraser

1 0 1 b R
1 1 1 b R
1 b 2 b R

Looper

1 0 1 0 R
1 1 1 1 L
1 b 2 b R

Sample inputs

1100101bbbbbb...
00000bbbbbbbb...

Sample inputs

01010101bbbbbb...
bbbbbbbbb...

Sample inputs

00000bbbbbbbb...
0001111bbbbbb...
111bbbbbbbbb...

3. Now see if you can figure out what each of the "mystery" Turing machines TM1 through TM6 does. A data file containing these machines is available in the instructions. Cut-and-paste the rules for TM1 into the simulator, then try the following inputs and record the output produced in each case. Do you see a pattern? How would you describe in words what this Turing machine does?

| TM1 Input | Output | TM1 Description: |
|----------------------|--------|------------------|
| 1. 011001qbbbbbb... | 1. | |
| 2. 111qbbbbbbbbb... | 2. | |
| 3. 00qbbbbbbbbb... | 3. | |
| 4. 0101qbbbbbbbbb... | 4. | |
| 5. 1101qbbbbbbbbb... | 5. | |

Next, try TM2 and record the output produced. Create more tests of your own to fill out the table, using inputs of the same form as shown in the table. Describe TM2's function to the right of its table.

| TM2 Input | Output | TM2 Description: |
|----------------------|--------|------------------|
| 1. 111bbbbbbbbb... | 1. | |
| 2. 1bbbbbbbbb... | 2. | |
| 3. 11111bbbbbbbbb... | 3. | |
| 4. | 4. | |
| 5. | 5. | |

(continued on back)

Now do the same thing for TM3 through TM6. Try to discover each Turing machine's intended purpose by experimenting with test inputs. Record at least five of your tests per machine in the tables below. Describe each Turing machine's function, if you can. **IMPORTANT:** The input format for TM3 and TM4 is two blocks of 1s containing no 0s, separated by q , followed by at least one blank.

| TM3 Input | Output | TM3 Description: |
|--------------------------|---------------|-------------------------|
| 1. 11 q 111bbbbbb ... | 1. | |
| 2. 1111 q 1bbbbbb ... | 2. | |
| 3. 1 q 1bbbbbbbbbb ... | 3. | |
| 4. | 4. | |
| 5. | 5. | |

| TM4 Input | Output | TM4 Description: |
|--------------------------|---------------|-------------------------|
| 1. 11111 q 11bbbb ... | 1. | |
| 2. 111 q 11bbbbbb ... | 2. | |
| 3. 111 q 111bbbbbb ... | 3. | |
| 4. | 4. | |
| 5. | 5. | |

IMPORTANT: Input for TM5 should be a block of 1s containing no 0s, followed by at least one blank.

| TM5 Input | Output | TM5 Description: |
|-----------------------|---------------|-------------------------|
| 1. 111111bbbbbbbb ... | 1. | |
| 2. 1111bbbbbbbbbb ... | 2. | |
| 3. 111bbbbbbbbbb ... | 3. | |
| 4. | 4. | |
| 5. | 5. | |

IMPORTANT: Input for TM6 should be a string of 0s and 1s, followed by at least one blank.

| TM6 Input | Output | TM6 Description: |
|-----------------------|---------------|-------------------------|
| 1. 10011001bbbb ... | 1. | |
| 2. 11011001bbbb ... | 2. | |
| 3. 1111bbbbbbbbbb ... | 3. | |
| 4. | 4. | |
| 5. | 5. | |

4. For another nice Turing machine simulator, go to the CS 30 Links page and select [Turing Machine simulator from Buena Vista University](#). This simulator uses a slightly different notation for writing Turing machine rules, but these differences are not too important. Follow the instructions to load and run the four sample programs available: **welcome**, **addition**, **multiply**, and **aNbN**.

Welcome just prints out a welcome message starting from a single x on the input tape. To try it, select "welcome.t" from the pull-down menu in the lower right corner of the simulator, then click Load. You will see the TM rules appear in the window. Next, click Auto to run the machine, or Step to single-step. The FastAuto button runs the machine at high speed. Click Load again to reset the machine to its starting state.

Addition adds two numbers represented as blocks of 1's separated by a single 0, leaving a single block of 1's behind on the tape representing the sum. Likewise, **multiply** computes the product of two numbers. Try running **multiply** using the FastAuto button. The **aNbN** program starts with a block of A's followed by a block of B's on the input tape and checks to see if the number of A's is the same as the number of B's. If so, it leaves a single \top behind on the tape when it halts.