MATH230: Assignment

Arithmetic with Turing Machines

This assignment is worth 20% of your final grade for MATH230. Each of the questions below are stated with their weight for this assignment. You may submit any combination of questions that you like; you may do as many or as few as you wish. Your grade will be determined by the sum of the points of the questions that you answer correctly, upto a maximum of 100 points. Partial credit will be given for Turing machines that don't pass all tests. Partial credit will be given for clear explanations of steps towards the specification of a Turing machine.

For each of the following exercises you must (i) achieve the required ouput for the specified inputs, (ii) explain in English how the machine works, and (iii) have the Turing machine halt with the read head on the starting (@) cell. Furthermore, assume all inputs are followed by an infinite tape of blank cells.

You may discuss your solutions with the teachers and anyone in the class. However, the final Turing machine instructions must be written by yourself. If any of the instructions for the assignment are unclear, then do not hesitate to ask the teachers for clarification.

Submission Format You must submit one zipped folder. This folder must contain one .txt file for each of the Turing machines you write. The .txt files must contain the Turing machine instructions and sufficient comments to explain your machine. Your machine instructions must follow the format explained in tutorials and in lectures. Save your .txt files with useful file names e.g. If you're writing a Turing machine to determine whether an integer is odd or even and the input is assumed to be unary, then save the file using a name like unary_oddeven.txt.

Turing Machine Code Format Ensure your Turing machine instructions are provided in a text file with the correct format. Keep the following in mind:

1. All instructions must be in the form: state,read,write,move,update;
2. Instructions are case sensitive
3. All instruction lines must end with a (semicolon) ; e.g. q0,@,@,R,q0;
4. All state names start with a q and are followed by an integer e.g. q12
5. In this assignment the only characters required are: 0,1,b,+,-
6. You may introduce other characters in your working
7. Only three possible move instructions: L,N,R
8. Final state is HALT;
9. Turing machine starts in a q0,@,@,R,... instruction
10. *File must end on a new line*
11. No other lines in the file can be empty
12. Order of the instructions in the file do not matter
13. Comment lines all begin with a (hashtag) #

An example is provided on Learn in the Turing section.

Unary Arithmetic

For these questions represent non-negative integers with unary representation.

(5 Points) Write a Turing machine to increment a unary number.

Input: a (possibly empty) string of 1s.

Output: a string of 1s corresponding to the successor of the input.

Example: Input = $_1$ Output = 1_1.

(5 points) Write a Turing machine to decrement a unary number.

Input: a (possibly empty) string of 1s.

Output: a string of 1s corresponding to the predecessor of the input.

Example: Input = 11 $_1$ Output = 1_1.

If input is 0 i.e. the empty string of 1s, then return 0 i.e. the empty string of 1s

(10 points) Write a Turing machine to sum two unary numbers.

Input: (possibly empty) strings of 1s separated by a + symbol.

Output: a string of 1s corresponding to the sum of the inputs.

Example: Input = $11111_1 + 11_1$ Output = 1111111_1.

(20 points) Write a Turing machine to calculate the (limited) difference of two unary integers.

Input: strings of 1s separated by a - symbol.

Output: a string of 1s corresponding to the sum of the inputs.

Example: Input = $11111_1 - 11_1$ Output = 111_1.

If $x \geq y$, then $x \dot{-} y = x - y$. Otherwise $x \dot{-} y = 0$.

(15 points) Write a Turing machine to determine whether a unary integer is odd/even.

- Input: a string of 1s.
- Output: a bit. 0 if odd, 1 if even.
- Example: Input = 11111_1 Output = 0.

(25 points) Write a Turing machine to determine whether one integer is less than another.

- Input: two strings of 1s separated by a blank cell.
- Output: a bit. 1 if first input smaller or equal, 0 if first input greater.
- Example: Input = $111b11111$ Output = 1.

(80 points) Write a Turing machine to decide if one integer divides another.

(100 points) Write a Turing machine to determine whether an integer is prime.

Binary Arithmetic

For these questions represent non-negative integers with binary representation.

(10 points) (Half-Adder) Write a Turing machine which adds two-bits.

Input: two bits $b_1 b_2$

Ouput: two bits sc according to the following table

b_1	b_2	sum	carry
0	0	0	0
1	0	1	0
0	1	1	0
1	1	0	1

(15 points) (Full-Adder) Write a Turing machine which adds two-bits with a carry bit.

Input: three bits $b_1 b_2$ and a carry c

Ouput: two bits sc according to the following table

b_1	b_2	c	sum	carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(25 points) Write a Turing machine to increment a binary number.

Input: a binary string.

Output: a binary string corresponding to the successor of the input.

Example: Input $= 11_2$ Output $= 100_2$.

(30 points) Write a Turing machine to decrement a binary number.

Input: a binary string.

Output: a binary string corresponding to the predecessor of the input.

Example: Input $= 100_2$ Output $= 11_2$.

(80 points) Write a Turing machine to sum two binary numbers.

Input: two binary strings separated by a + symbol

Output: a binary string corresponding to the sum of the inputs.

Example: Input $= 111_2 + 10_2$ Output $= 1001_2$.

(70 points) Write a Turing machine to calculate the (limited) difference of two binary integers.

Input: two binary strings separated by a - symbol

Output: a binary string corresponding to the limited difference of the inputs.

Example: Input = $1001_2 - 10_2$ Output = 111_2.

If $x \geq y$, then $x \dot{-} y = x - y$. Otherwise $x \dot{-} y = 0$.

(5 points) Write a Turing machine to determine whether a binary integer is odd/even.

- Input: a binary string.
- Output: a bit. 0 if the input is odd, 1 if the input is even.
- Example: Input = 11111_2 Output = 0.

(30 points) Write a Turing machine to determine whether one binary integer is less than another.

- Input: Two binary strings separated by a blank cell.
- Output: Single bit, 1 if the first input smaller or equal, 0 if the first input greater.
- Example: Input = 11111_2 11110_2 Output = 0.