Cody Lawson,

Sydney Shelby,

Abdulrehman Bhidya

COSC312

10-25-2018

Homework 8 (Turing Machine Design)

**Collaborators:**

**(1) names of collaborators and the roles for each in the design of the Turning Machine**

Sydney Shelby

* Multiplication (2\*2\*2\*2….)
* Unary to Base 10 Conversion

Cody Lawson

* Base 10 (0-99) to Unary Conversion

Abdulrehman Bidhya

* Comparison Operator
* LOG() format parsing

We all met together in order to glue the components together, test, and debug.

**Problem/Machine Information:**

Our turing machine will calculate the log (base 2) of any number x <= 99. Since floating point precision would take a ton of states to calculate, we decided to round the answer up to the closest integer. The input is treated just like a function call in C (name of function + parentheses with parameters).

Example inputs:

“LOG(5)” “LOG(22)” “LOG(95)”

All caps on the letters is a must, and the parameter must be an integer x so that 1 <= x < 100

Examples of bad inputs:

“LOG(100)” “log(2)”

We brought up the idea of solving log base 2 in class, and Dr. Berry seemed impressed and shared the idea with the class. Our inspiration for this project stems from the importance of powers of 2 in computer science.

The process our Turing Machine follows to calculate LOG(X) is as follows.

1. Pattern Match the input for LOG(X) and remove LOG()
2. Convert X from base 10 to unary and add an iteration marker, delimiter, and 2 0s to the right
3. Compare the left side to the right side,
   1. if the right side is greater move to step 5
   2. If the left side is greater move to step 4
4. Multiply the right side by 2 and add an additional iteration marker. Move to step 3
5. Delete everything except the iteration markers and convert them into base 10

Run Time Suggestions:

For larger numbers we recommend using the “No Delay” option when running the machine because as the input approaches 100 even the “No Delay” setting takes seconds to complete. It is okay to run the machine on a slower setting.

The hardware requirements to run our Turing Machine on STEM scale exponentially with the inputs. For example, we are able to run inputs from 64 to 99 with an i7 processor and 16GB of RAM. Without sufficient power, STEM may crash.

**Turing Machine State Descriptions:**

Start State: -100

States from -100 to -94 Parse “LOG”

States from 0 to 18 convert base 10 to unary

States from -94 to -50 formats the unary with an iteration marker (Y) and delimiter (X)

States from -50 to -46 compares the right side to the left

States -44,-43, -13, -12 are involved in converting Iteration markers into base 10

States -15 to -20 work to duplicate the right side

