Turing Machine Project
COSC 312
Fall 2023

Base-10 Addition Turing Machine

Introduction

This Turing Machine is designed to preform addition on two base-10 numbers. The goal was to create a "user-friendly" machine with an input alphabet (Σ) that mirrors the common usage of numbers (as opposed to binary or unary). The tape alphabet (Γ) includes additional symbols to prevent re-reading or re-summing digits that have already been evaluated. There are 29 states in total after simplification (the rough draft had over 40 states).

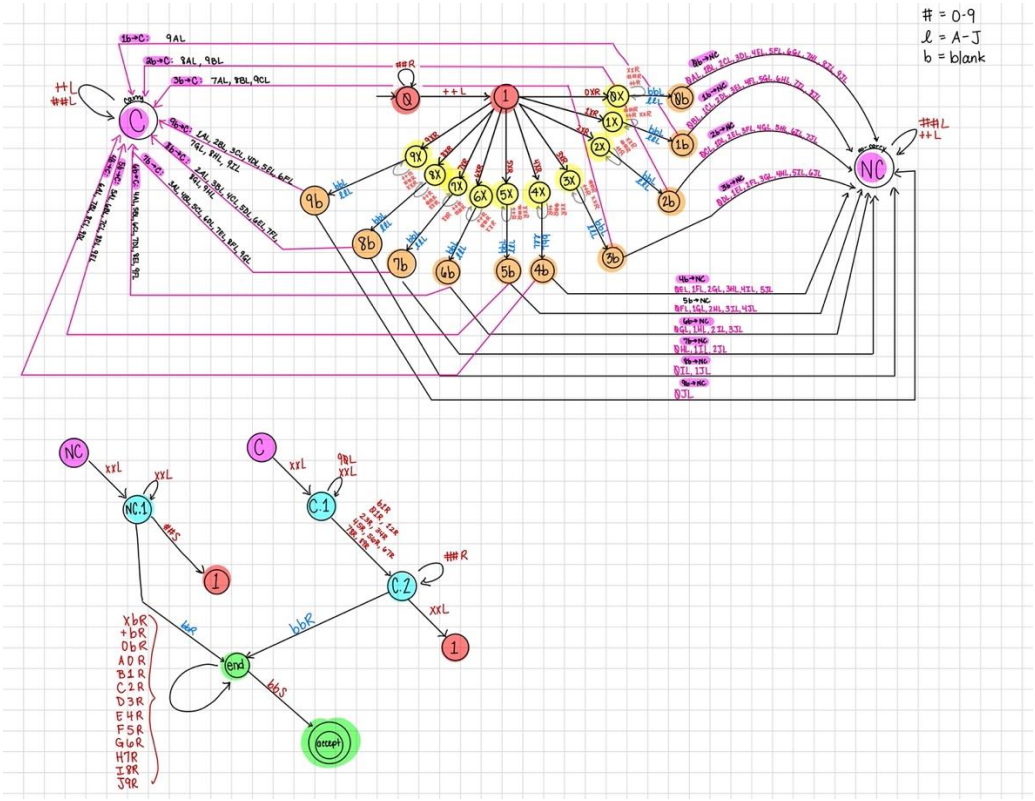
Σ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +}

Γ = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, X, A, B, C, D, E, F, G, H, I, J}

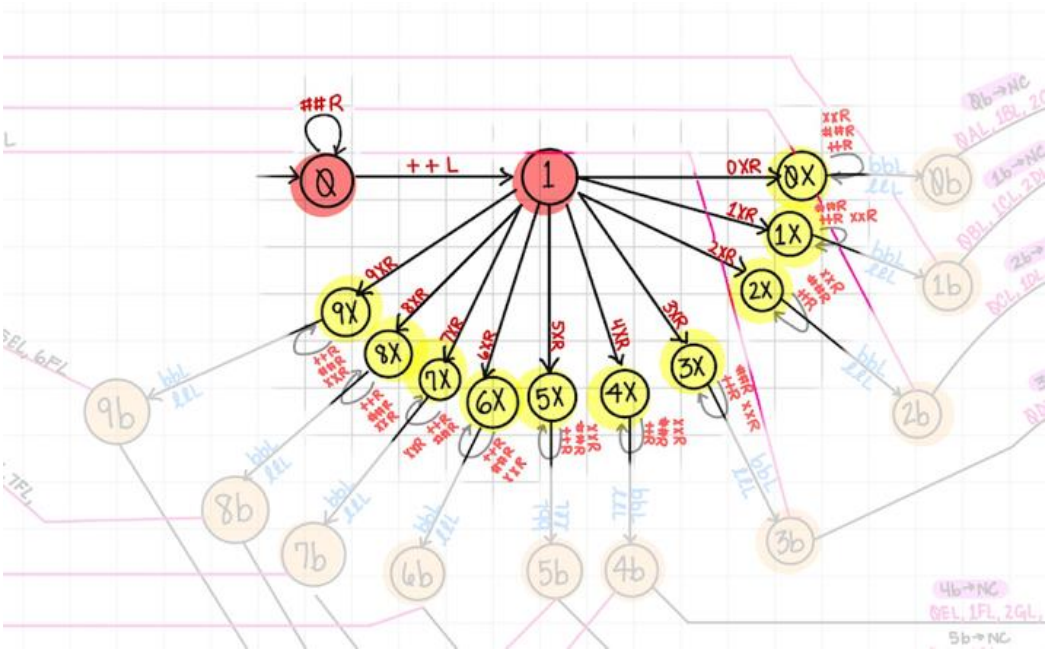
Breakdown

The handwritten version was chosen for this breakdown due to the shorthand making it less cluttered. The key for the shorthand is as follows: # represents the digits 0 through 9, ℓ (a cursive L) represents the letters A through J, and b represents a blank/space.

Turing Machine Project
COSC 312
Fall 2023



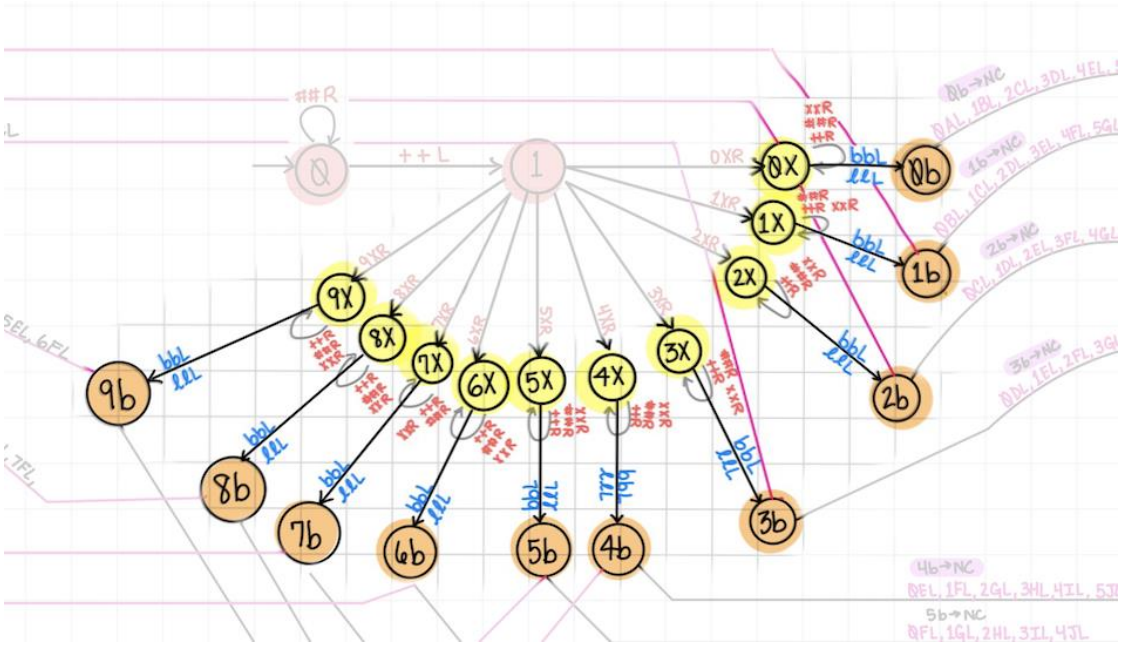
(Figure 1) – The Full Turing Machine (hand-drawn) version.



(Figure 2) – Step 1: Initialization

Step 1: Initialization

In the initial phase (the red states), the machine focuses on adding the one's place. It traverses the input until it encounters the + symbol, indicating it has passed the one's place (and moving to state 1), so it redirects to the left to read in the one's place. From there it will overwrite the digit with an X so the machine will not re-read the digit later.



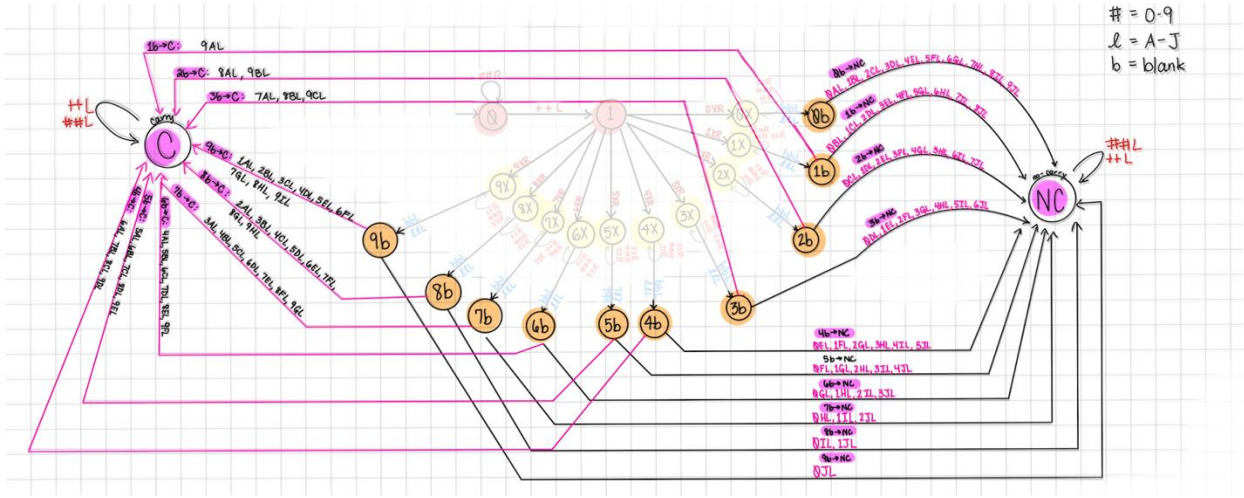
(Figure 3) – Step 2: Processing the second number

Step 2: Processing the second number

The machine transitions to one of the 10 yellow states based on the digit it reads, remaining in that state until it reaches the one's place of the second number. The yellow and orange states handle the process similarly to the initial states (0 and 1 in red). The machine traverses to a blank, indicating it passed the one's place of the second number. A ##R loop on

the yellow states prevents adding the digit from the one's place of the first number to the ten's or hundred's place of the second number. The bbL transition from yellow to orange states enables the machine to read the one's place digit after passing the blank, transitioning accordingly.

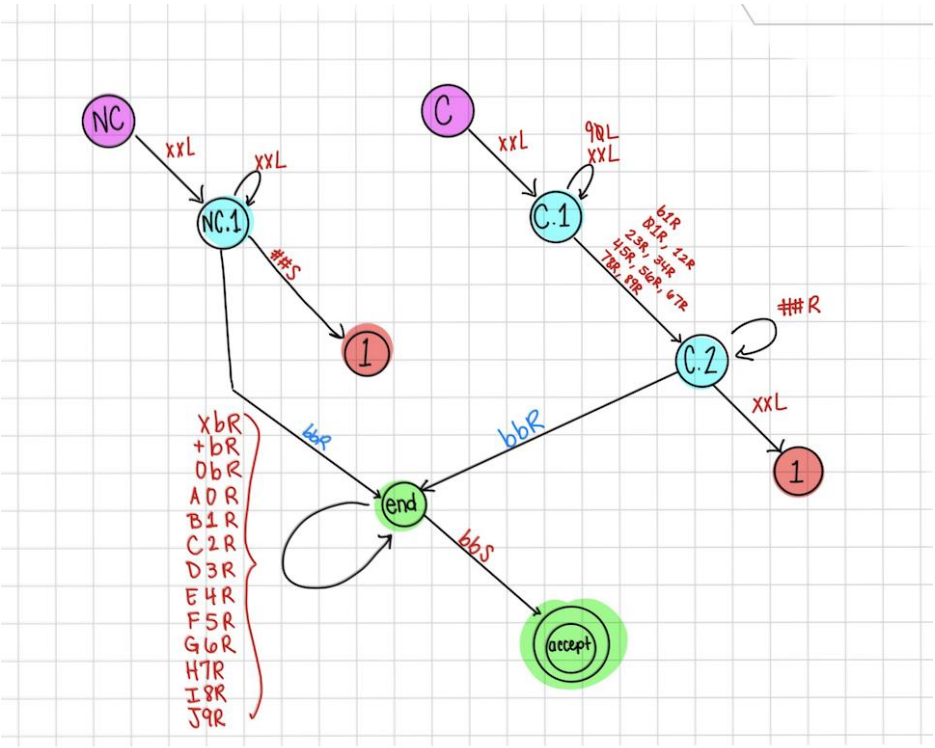
Two additional components in these transitions include the XXR and ++R loops, facilitating the machine's ability to loop back and add the digit from the ten's or hundred's place of the first number. For instance, if the tape is 10X+11E, the machine skips over the X and + to reach the second number. The ℓ ℓL replaces the bbL transition when the machine returns for the ten's or hundred's places, as illustrated in the example 1XX+13F, where the machine stops at F, turns left to read 3, and transitions accordingly.



(Figure 4) – Step 3: Digit Addition

Step 3: Digit Addition

This section features orange states representing the digits read from the first number. The value to write is determined by the digit read from the second number, transitioning to either a non-carry state ("NC") or a carry state ("C"). The write value is represented by letters A through J, corresponding to the numbers 0-9. Loop transitions on both C and NC states allow the machine to go left and sum the next digit from the first number.



(Figure 4) – Step 4 and Step 5

Step 4: The carrying problem

In instances where the sum of two digits yields a carry, the machine addresses this by transitioning to the carry state (labeled "C" and

highlighted in pink). Using X as a sentinel value, the machine signals it has reached the first number and moves to state C.1. It then traverses over X's until encountering a digit to which the carried-over 1 is added. An exception occurs when reaching the digit 9, resulting in another carry. In this case, the machine writes a 0, moves left on the tape to carry over to the next digit (C.1 to C.2 transition), and resumes the process. Once the carry is resolved, the machine proceeds to sum the next digit. Having moved the tape left, the machine returns to the right until encountering an X, moving left, and returning to state 1.

In the absence of a carry, the machine, indicating it reached the first number using X as a sentinel value, transitions to state NC.1. It passes over X's until reaching the digit to be summed, transitioning back to state 1.

Step 5: The Final Step (Figure 5)

Upon reaching step 4 and moving left on the first number to find another digit to sum, encountering a blank signifies the completion of digit summation. The bbR transitions from NC.1 and C.1 to the "end" state symbolize this conclusion. At this point, all digits are represented by letters. The loop on the "end" state converts symbols either to blanks (for cleanup) or back to the appropriate digits. Once the machine traverses right, finding a blank, it enters the "accept" state, leaving the answer on the tape.

Turing Machine Project
COSC 312
Fall 2023

How to Run it

The input to the TM machine does have some restrictions and rules.

1. The number of the longest length should be the leftmost number. If both numbers are of the same length, then it doesn't matter.
2. The second number should be (length of first number) + 1. The gaps of the second number should be filled in with leading 0s. For example, if the length of the first number is 4, and the second number is 33 (length 2), then the input should be 0033. You can see the input table for more examples.
3. There should be no spaces/blanks in the input (except before and after, of course).
4. There is no limit to the number of digits.

The machine will start in state 0 with the head at the first digit of the input. When the TM hits the accept state, the sum overwrites the second number and the rest of the input is replaced with blanks, leaving only the sum.

Input	Output
999+0037	1036
900+0100	1000
987+0453	1440
6+02	8
12345678+000012345	12358023

Turing Machine Project
COSC 312
Fall 2023

3+08	11
999+0999	1998
2003+01996	3999