Adrian A. Gerbaud H.
Samantha Rack

Project 3 Report

The language(s) platform used for testing

We wrote our Turing Machine in C++ to make use of the predefined STL classes. We especially took advantage of the STL vector class and C++ strings. We tested the program with different Turning Machine descriptions, including the ones given, some from class notes, as well as the three that we ourselves created and all worked successfully. The platform for testing used were the student machines.

A high level description of the program including major data structures

The implementation of a Turning Machine was itself not very difficult. The program works as follows. First, the text file that describes the behavior of the Turing Machine is read and parsed. The alphabets, states, and end states are kept in either vectors of strings or strings. A vector is also used to keep the input Tape that is given. A vector of vectors of strings was used to keep a table of the transitions. This table maintained transition structures for each state, tape character pair. A valid flag was maintained with each of these structures indicating whether or not there is a transition for that pair. This flag was also used to ensure the machine was deterministic – if the valid flag was already set and a transition for that pair was read, then the machine description provided was non-deterministic. Additional error checking was done to be sure that the format and content of the description produced a valid Turing Machine.

As long as we have not exceeded the maximum number of steps and we have not entered either the accept or reject state, we continue to advance our machine and print the trace. We advance by following the information in our transition table and changing the input tape if needed. If either the accept or reject state is reached, we halt and print the result.

Description of the program was tested

Turing Machine 1:

The first Turing Machine example executes binary addition for arbitrarily sized operators. The initial tape (input) is of the form {0,1}{0,1}*+{0,1}{0,1}*, where the + in the middle is the character. Examples are 101+1111111, 0+1, etc. The output maintains the original length of the input tape (without blank characters), and is the binary solution for the addition with leading zeros. The machine description can be found in ourTM_1.txt.

The machine begins by marking the first bit of the input string (a = 0, b = 1). Then, it traverses to the end of the first input string (i.e. The character immediately preceding “+”), and marks it (all of the marked characters in the first string of bits are marked “c”). Depending on if this is a 0 or 1, the machine chooses its next state, and then traverses to the end of the input before blank characters. At this point, the machine adds the two least significant bits and puts this value in the last position and marks it to differentiate it from un-added values (e = 0, f = 1). Then, the machine continues by returning to the second least significant bit of the first number, marks it, and adds it to the second least significant bit of the second number. The states to which the machine transitions depend on whether of not there is a carry bit resulting from any addition step. In order for the machine to terminate, there are additional states that handle the cases where the input “runs out” of bits in either string. The final step of the machine is to convert all c's, e's, and f's back to binary digits for the output.

A large number and variety of inputs were used to test this Turing Machine because of the extensive number of transitions in the transition function. The most useful inputs for checking the validity of the machine were inputs where: (1) the first number has more digits than the second, (2) the second number has more digits than the first, (3) the numbers have the same number of digits, and (4) only 1 digit in each of the numbers. With these cases, the inputs were complex enough to sometimes require carry bits and sometimes not. Examples of the inputs used: (1) 1011+101, (2) 0+1111, (3) 10101+01101, and (4) 1+0. The input tape can be found in ourTM_1-input.txt. *NOTE: Other, longer bit strings were added during testing, but these were not included because of the extensive trace.*

```
Input Tape:
4
1,0,1,1,+,1,0,1
0,+,1,1,1,1
1,0,1,+,0,0,1
1,+,0
```

```
Output:
()q0(1,0,1,1,+,1,0,1)
(b)q1(0,1,1,+,1,0,1)
(b,0)q1(1,1,+,1,0,1)
(b,0,1)q1(1,+,1,0,1)
(b,0,1,1)q1(+,1,0,1)
(b,0,1)q2(1,+,1,0,1)
(b,0,1,c)q3(+,1,0,1)
(b,0,1,c,+)q3(1,0,1)
(b,0,1,c,+,1)q3(0,1)
(b,0,1,c,+,1,0)q3(1)
(b,0,1,c,+,1,0,1)q3()
(b,0,1,c,+,1,0)q4(1)
(b,0,1,c,+,1)q5(0,e)
(b,0,1,c,+)q6(1,0,e)
(b,0,1,c)q6(+,1,0,e)
(b,0,1)q32(c,+,1,0,e)
(b,0)q32(1,c,+,1,0,e)
(b,0,c)q7(c,+,1,0,e)
```

```
(b,0,c,c)q7(+,1,0,e)
(b,0,c,c,+)q7(1,0,e)
(b,0,c,c,+,1)q7(0,e)
(b,0,c,c,+,1,0)q7(e)
(b,0,c,c,+,1)q8(0,e)
(b,0,c,c,+)q5(1,e,e)
(b,0,c,c)q6(+,1,e,e)
(b,0,c)q32(c,+,1,e,e)
(b,0)q32(c,c,+,1,e,e)
(b)q32(0,c,c,+,1,e,e)
(b,c)q3(c,c,+,1,e,e)
(b,c,c)q3(c,+,1,e,e)
(b,c,c,c)q3(+,1,e,e)
(b,c,c,c,+)q3(1,e,e)
(b,c,c,c,+,1)q3(e,e)
(b,c,c,c,+)q4(1,e,e)
(b,c,c,c)q5(+,e,e,e)
(b,c,c)q23(c,+,e,e,e)
(b,c)q23(c,c,+,e,e,e)
(b)q23(c,c,c,+,e,e,e)
()q23(b,c,c,c,+,e,e,e)
(0)q26(c,c,c,+,e,e,e)
(0,0)q26(c,c,+,e,e,e)
(0,0,0)q26(c,+,e,e,e)
(0,0,0,0)q26(+,e,e,e)
(0,0,0,0,0)q26(e,e,e)
(0,0,0,0)q27(0,e,e,e)
(0,0,0)q28(0,0,e,e,e)
(0,0,0,1)q29(0,e,e,e)
(0,0,0,1,0)q29(e,e,e)
(0,0,0,1,0,0)q29(e,e)
(0,0,0,1,0,0,0)q29(e)
(0,0,0,1,0,0,0,0)q29()
(0,0,0,1,0,0,0,0, )q30()
ACCEPT

()q0(0,+,1,1,1,1)
(a)q1(+,1,1,1,1)
()q2(a,+,1,1,1,1)
(0)q33(+,1,1,1,1)
(0,0)q33(1,1,1,1)
(0,0,1)q33(1,1,1)
(0,0,1,1)q33(1,1)
(0,0,1,1,1)q33(1)
(0,0,1,1,1,1)q33()
(0,0,1,1,1,1, )q30()
ACCEPT

()q0(1,0,1,+,0,0,1)
(b)q1(0,1,+,0,0,1)
```

```
(b,0)q1(1,+,0,0,1)
(b,0,1)q1(+,0,0,1)
(b,0)q2(1,+,0,0,1)
(b,0,c)q3(+,0,0,1)
(b,0,c,+)q3(0,0,1)
(b,0,c,+,0)q3(0,1)
(b,0,c,+,0,0)q3(1)
(b,0,c,+,0,0,1)q3()
(b,0,c,+,0,0)q4(1)
(b,0,c,+,0)q5(0,e)
(b,0,c,+)q6(0,0,e)
(b,0,c)q6(+,0,0,e)
(b,0)q32(c,+,0,0,e)
(b)q32(0,c,+,0,0,e)
(b,c)q3(c,+,0,0,e)
(b,c,c)q3(+,0,0,e)
(b,c,c,+)q3(0,0,e)
(b,c,c,+,0)q3(0,e)
(b,c,c,+,0,0)q3(e)
(b,c,c,+,0)q4(0,e)
(b,c,c,+)q10(0,f,e)
(b,c,c)q9(+,0,f,e)
(b,c)q2(c,+,0,f,e)
(b)q2(c,c,+,0,f,e)
()q2(b,c,c,+,0,f,e)
(0)q34(c,c,+,0,f,e)
(0,0)q34(c,+,0,f,e)
(0,0,0)q34(+,0,f,e)
(0,0,0,0)q34(0,f,e)
(0,0,0,0,0)q34(f,e)
(0,0,0,0)q35(0,f,e)
(0,0,0,0,1)q33(f,e)
(0,0,0,0,1,1)q33(e)
(0,0,0,0,1,1,0)q33()
(0,0,0,0,1,1,0, )q30()
ACCEPT

()q0(1,+,0)
(b)q1(+,0)
()q2(b,+,0)
(0)q34(+,0)
(0,0)q34(0)
(0,0,0)q34()
(0,0)q35(0)
(0,0,1)q33()
(0,0,1, )q30()
ACCEPT
```

Turing Machine 2:

The second Turing machine takes a binary string that ends with a # sign and will reverse it. The machine begins by moving the string forward in the tape to allow for space in order to be able to reverse the string. After allocating that space, the algorithm will progressively take the first letter in the string and move it to the left until it is in the last position of the reversed string. It will do this repeatedly until the string is reversed. It is important to note that leading # signs after the reversed string have not been erased by the machine, but erasing them is trivial. The machine description can be found in ourTM_2.txt.

The three inputs chosen for testing represent strings that have an even number of characters, an odd number of characters, a palindrome, an d non-palindromes. Additionally, there is an input that represents only one character, to show that it will reverse this correctly as well. The input tape can be found in ourTM_2-input.txt.

Input Tape:

```
4
0,0,0,1,1,1,#
1,1,1,#
1,0,1,0,#
0,#
```

Output:

```
()q0(0,0,0,1,1,1,#)
( )q1(0,0,1,1,1,#)
( ,0)q1(0,1,1,1,#)
( ,0,0)q1(1,1,1,#)
( ,0,0,1)q1(1,1,#)
( ,0,0,1,1)q1(1,#)
( ,0,0,1,1,1)q1(#)
( ,0,0,1,1,1,#)q2()
( ,0,0,1,1,1)q3(#,0)
( ,0,0,1,1)q3(1,#,0)
( ,0,0,1)q3(1,1,#,0)
( ,0,0)q3(1,1,1,#,0)
( ,0)q3(0,1,1,1,#,0)
( )q3(0,0,1,1,1,#,0)
()q3( ,0,0,1,1,1,#,0)
( )q0(0,0,1,1,1,#,0)
( , )q1(0,1,1,1,#,0)
( , ,0)q1(1,1,1,#,0)
( , ,0,1)q1(1,1,#,0)
( , ,0,1,1)q1(1,#,0)
( , ,0,1,1,1)q1(#,0)
( , ,0,1,1,1,#)q2(0)
( , ,0,1,1,1,#,0)q2()
( , ,0,1,1,1,#)q3(0,0)
( , ,0,1,1,1)q3(#,0,0)
( , ,0,1,1)q3(1,#,0,0)
( , ,0,1)q3(1,1,#,0,0)
( , ,0)q3(1,1,1,#,0,0)
( , )q3(0,1,1,1,#,0,0)
( )q3( ,0,1,1,1,#,0,0)
( , )q0(0,1,1,1,#,0,0)
( , , )q1(1,1,1,#,0,0)
( , , ,1)q1(1,1,#,0,0)
( , , ,1,1)q1(1,#,0,0)
( , , ,1,1,1)q1(#,0,0)
( , , ,1,1,1,#)q2(0,0)
( , , ,1,1,1,#,0)q2(0)
( , , ,1,1,1,#,0,0)q2()
( , , ,1,1,1,#,0)q3(0,0)
( , , ,1,1,1,#)q3(0,0,0)
( , , ,1,1,1)q3(#,0,0,0)
( , , ,1,1)q3(1,#,0,0,0)
( , , ,1)q3(1,1,#,0,0,0)
( , , )q3(1,1,1,#,0,0,0)
( , )q3( ,1,1,1,#,0,0,0)
( , , )q0(1,1,1,#,0,0,0)
( , , , )q4(1,1,#,0,0,0)
( , , , ,1)q4(1,#,0,0,0)
( , , , ,1,1)q4(#,0,0,0)
( , , , ,1,1,#)q5(0,0,0)
( , , , ,1,1,#,0)q5(0,0)
( , , , ,1,1,#,0,0)q5(0)
( , , , ,1,1,#,0,0,0)q5()
( , , , ,1,1,#,0,0)q6(0,1)
( , , , ,1,1,#,0)q6(0,0,1)
( , , , ,1,1,#)q6(0,0,0,1)
( , , , ,1,1)q6(#,0,0,0,1)
( , , , ,1)q6(1,#,0,0,0,1)
( , , , )q6(1,1,#,0,0,0,1)
( , , )q6( ,1,1,#,0,0,0,1)
( , , , )q0(1,1,#,0,0,0,1)
( , , , , )q4(1,#,0,0,0,1)
( , , , , ,1)q4(#,0,0,0,1)
( , , , , ,1,#)q5(0,0,0,1)
( , , , , ,1,#,0)q5(0,0,1)
( , , , , ,1,#,0,0)q5(0,1)
( , , , , ,1,#,0,0,0)q5(1)
( , , , , ,1,#,0,0,0,1)q5()
( , , , , ,1,#,0,0,0)q6(1,1)
( , , , , ,1,#,0,0)q6(0,1,1)
( , , , , ,1,#,0)q6(0,0,1,1)
( , , , , ,1,#)q6(0,0,0,1,1)
( , , , , ,1)q6(#,0,0,0,1,1)
( , , , , )q6(1,#,0,0,0,1,1)
( , , , )q6( ,1,#,0,0,0,1,1)
( , , , , )q0(1,#,0,0,0,1,1)
( , , , , , )q4(#,0,0,0,1,1)
( , , , , , ,#)q5(0,0,0,1,1)
( , , , , , ,#,0)q5(0,0,1,1)
( , , , , , ,#,0,0)q5(0,1,1)
( , , , , , ,#,0,0,0)q5(1,1)
( , , , , , ,#,0,0,0,1)q5(1)
( , , , , , ,#,0,0,0,1,1)q5()
( , , , , , ,#,0,0,0,1)q6(1,1)
( , , , , , ,#,0,0,0)q6(1,1,1)
( , , , , , ,#,0,0)q6(0,1,1,1)
( , , , , , ,#,0)q6(0,0,1,1,1)
( , , , , , ,#)q6(0,0,0,1,1,1)
( , , , , , )q6(#,0,0,0,1,1,1)
( , , , , )q6( ,#,0,0,0,1,1,1)
( , , , , , )q0(#,0,0,0,1,1,1)
( , , , , , , )q7(0,0,0,1,1,1)
( , , , , , )q12( ,#,0,0,1,1,1)
( , , , , , ,0)q13(#,0,0,1,1,1)
( , , , , , ,0,#)q14(0,0,1,1,1)
( , , , , , ,0)q12(#,#,0,1,1,1)
( , , , , , )q12(0,#,#,0,1,1,1)
( , , , , )q12( ,0,#,#,0,1,1,1)
( , , , , ,0)q13(0,#,#,0,1,1,1)
( , , , , ,0,0)q13(#,#,0,1,1,1)
( , , , , ,0,0,#)q14(#,0,1,1,1)
( , , , , ,0,0,#,#)q14(0,1,1,1)
( , , , , ,0,0,#)q12(#,#,1,1,1)
( , , , , ,0,0)q12(#,#,#,1,1,1)
( , , , , ,0)q12(0,#,#,#,1,1,1)
( , , , , )q12(0,0,#,#,#,1,1,1)
( , , , )q12( ,0,0,#,#,#,1,1,1)
( , , , ,0)q13(0,0,#,#,#,1,1,1)
( , , , ,0,0)q13(0,#,#,#,1,1,1)
( , , , ,0,0,0)q13(#,#,#,1,1,1)
( , , , ,0,0,0,#)q14(#,#,1,1,1)
( , , , ,0,0,0,#,#)q14(#,1,1,1)
( , , , ,0,0,0,#,#,#)q14(1,1,1)
( , , , ,0,0,0,#,#)q9(#,#,1,1)
( , , , ,0,0,0,#)q9(#,#,#,1,1)
( , , , ,0,0,0)q9(#,#,#,#,1,1)
( , , , ,0,0)q9(0,#,#,#,#,1,1)
( , , , ,0)q9(0,0,#,#,#,#,1,1)
( , , , )q9(0,0,0,#,#,#,#,1,1)
( , , )q9( ,0,0,0,#,#,#,#,1,1)
( , , ,1)q10(0,0,0,#,#,#,#,1,1)
( , , ,1,0)q10(0,0,#,#,#,#,1,1)
( , , ,1,0,0)q10(0,#,#,#,#,1,1)
( , , ,1,0,0,0)q10(#,#,#,#,1,1)
( , , ,1,0,0,0,#)q11(#,#,#,1,1)
( , , ,1,0,0,0,#,#)q11(#,#,1,1)
( , , ,1,0,0,0,#,#,#)q11(#,1,1)
( , , ,1,0,0,0,#,#,#,#)q11(1,1)
( , , ,1,0,0,0,#,#,#)q9(#,#,1)
( , , ,1,0,0,0,#,#)q9(#,#,#,1)
( , , ,1,0,0,0,#)q9(#,#,#,#,1)
( , , ,1,0,0,0)q9(#,#,#,#,#,1)
( , , ,1,0,0)q9(0,#,#,#,#,#,1)
( , , ,1,0)q9(0,0,#,#,#,#,#,1)
( , , ,1)q9(0,0,0,#,#,#,#,#,1)
( , , )q9(1,0,0,0,#,#,#,#,#,1)
( , )q9( ,1,0,0,0,#,#,#,#,#,1)
( , ,1)q10(1,0,0,0,#,#,#,#,#,1)
( , ,1,1)q10(0,0,0,#,#,#,#,#,1)
( , ,1,1,0)q10(0,0,#,#,#,#,#,1)
( , ,1,1,0,0)q10(0,#,#,#,#,#,1)
( , ,1,1,0,0,0)q10(#,#,#,#,#,1)
( , ,1,1,0,0,0,#)q11(#,#,#,#,1)
( , ,1,1,0,0,0,#,#)q11(#,#,#,1)
( , ,1,1,0,0,0,#,#,#)q11(#,#,1)
( , ,1,1,0,0,0,#,#,#,#)q11(#,1)
( , ,1,1,0,0,0,#,#,#,#,#)q11(1)
( , ,1,1,0,0,0,#,#,#,#)q9(#,#)
( , ,1,1,0,0,0,#,#,#)q9(#,#,#)
( , ,1,1,0,0,0,#,#)q9(#,#,#,#)
( , ,1,1,0,0,0,#)q9(#,#,#,#,#)
( , ,1,1,0,0,0)q9(#,#,#,#,#,#)
( , ,1,1,0,0)q9(0,#,#,#,#,#,#)
( , ,1,1,0)q9(0,0,#,#,#,#,#,#)
( , ,1,1)q9(0,0,0,#,#,#,#,#,#)
( , ,1)q9(1,0,0,0,#,#,#,#,#,#)
( , )q9(1,1,0,0,0,#,#,#,#,#,#)
( )q9( ,1,1,0,0,0,#,#,#,#,#,#)
( ,1)q10(1,1,0,0,0,#,#,#,#,#,#)
( ,1,1)q10(1,0,0,0,#,#,#,#,#,#)
( ,1,1,1)q10(0,0,0,#,#,#,#,#,#)
( ,1,1,1,0)q10(0,0,#,#,#,#,#,#)
( ,1,1,1,0,0)q10(0,#,#,#,#,#,#)
( ,1,1,1,0,0,0)q10(#,#,#,#,#,#)
( ,1,1,1,0,0,0,#)q11(#,#,#,#,#)
( ,1,1,1,0,0,0,#,#)q11(#,#,#,#)
( ,1,1,1,0,0,0,#,#,#)q11(#,#,#)
( ,1,1,1,0,0,0,#,#,#,#)q11(#,#)
( ,1,1,1,0,0,0,#,#,#,#,#)q11(#)
( ,1,1,1,0,0,0,#,#,#,#,#,#)q11()
( ,1,1,1,0,0,0,#,#,#,#,#)q15(#)
ACCEPT

()q0(1,1,1,#)
( )q4(1,1,#)
( ,1)q4(1,#)
( ,1,1)q4(#)
( ,1,1,#)q5()
( ,1,1)q6(#,1)
( ,1)q6(1,#,1)
( )q6(1,1,#,1)
()q6( ,1,1,#,1)
( )q0(1,1,#,1)
( , )q4(1,#,1)
( , ,1)q4(#,1)
( , ,1,#)q5(1)
( , ,1,#,1)q5()
( , ,1,#)q6(1,1)
( , ,1)q6(#,1,1)
( , )q6(1,#,1,1)
( )q6( ,1,#,1,1)
( , )q0(1,#,1,1)
( , , )q4(#,1,1)
( , , ,#)q5(1,1)
( , , ,#,1)q5(1)
( , , ,#,1,1)q5()
( , , ,#,1)q6(1,1)
( , , ,#)q6(1,1,1)
( , , )q6(#,1,1,1)
( , )q6( ,#,1,1,1)
( , , )q0(#,1,1,1)
( , , , )q7(1,1,1)
( , , )q9( ,#,1,1)
( , , ,1)q10(#,1,1)
( , , ,1,#)q11(1,1)
( , , ,1)q9(#,#,1)
( , , )q9(1,#,#,1)
( , )q9( ,1,#,#,1)
( , ,1)q10(1,#,#,1)
( , ,1,1)q10(#,#,1)
( , ,1,1,#)q11(#,1)
( , ,1,1,#,#)q11(1)
( , ,1,1,#)q9(#,#)
( , ,1,1)q9(#,#,#)
( , ,1)q9(1,#,#,#)
( , )q9(1,1,#,#,#)
( )q9( ,1,1,#,#,#)
( ,1)q10(1,1,#,#,#)
( ,1,1)q10(1,#,#,#)
( ,1,1,1)q10(#,#,#)
( ,1,1,1,#)q11(#,#)
( ,1,1,1,#,#)q11(#)
( ,1,1,1,#,#,#)q11()
( ,1,1,1,#,#)q15(#)
ACCEPT
```

```
()q0(1,0,1,0,#)
( )q4(0,1,0,#)
( ,0)q4(1,0,#)
( ,0,1)q4(0,#)
( ,0,1,0)q4(#)
( ,0,1,0,#)q5()
( ,0,1,0)q6(#,1)
( ,0,1)q6(0,#,1)
( ,0)q6(1,0,#,1)
( )q6(0,1,0,#,1)
()q6( ,0,1,0,#,1)
( )q0(0,1,0,#,1)
( , )q1(1,0,#,1)
( , ,1)q1(0,#,1)
( , ,1,0)q1(#,1)
( , ,1,0,#)q2(1)
( , ,1,0,#,1)q2()
( , ,1,0,#)q3(1,0)
( , ,1,0)q3(#,1,0)
( , ,1)q3(0,#,1,0)
( , )q3(1,0,#,1,0)
( )q3( ,1,0,#,1,0)
( , )q0(1,0,#,1,0)
( , , )q4(0,#,1,0)
( , , ,0)q4(#,1,0)
( , , ,0,#)q5(1,0)
( , , ,0,#,1)q5(0)
( , , ,0,#,1,0)q5()
( , , ,0,#,1)q6(0,1)
( , , ,0,#)q6(1,0,1)
( , , ,0)q6(#,1,0,1)
( , , )q6(0,#,1,0,1)
( , )q6( ,0,#,1,0,1)
( , , )q0(0,#,1,0,1)
( , , , )q1(#,1,0,1)
( , , , ,#)q2(1,0,1)
( , , , ,#,1)q2(0,1)
( , , , ,#,1,0)q2(1)
( , , , ,#,1,0,1)q2()
( , , , ,#,1,0)q3(1,0)
( , , , ,#,1)q3(0,1,0)
( , , , ,#)q3(1,0,1,0)
( , , , )q3(#,1,0,1,0)
( , , )q3( ,#,1,0,1,0)
( , , , )q0(#,1,0,1,0)
( , , , , )q7(1,0,1,0)
( , , , )q9( ,#,0,1,0)
( , , , ,1)q10(#,0,1,0)
( , , , ,1,#)q11(0,1,0)
( , , , ,1)q12(#,#,1,0)
( , , , )q12(1,#,#,1,0)
( , , )q12( ,1,#,#,1,0)
( , , ,0)q13(1,#,#,1,0)
( , , ,0,1)q13(#,#,1,0)
( , , ,0,1,#)q14(#,1,0)
( , , ,0,1,#,#)q14(1,0)
( , , ,0,1,#)q9(#,#,0)
( , , ,0,1)q9(#,#,#,0)
( , , ,0)q9(1,#,#,#,0)
( , , )q9(0,1,#,#,#,0)
( , )q9( ,0,1,#,#,#,0)
( , ,1)q10(0,1,#,#,#,0)
( , ,1,0)q10(1,#,#,#,0)
( , ,1,0,1)q10(#,#,#,0)
( , ,1,0,1,#)q11(#,#,0)
( , ,1,0,1,#,#)q11(#,0)
( , ,1,0,1,#,#,#)q11(0)
( , ,1,0,1,#,#)q12(#,#)
( , ,1,0,1,#)q12(#,#,#)
( , ,1,0,1)q12(#,#,#,#)
( , ,1,0)q12(1,#,#,#,#)
( , ,1)q12(0,1,#,#,#,#)
( , )q12(1,0,1,#,#,#,#)
( )q12( ,1,0,1,#,#,#,#)
( ,0)q13(1,0,1,#,#,#,#)
( ,0,1)q13(0,1,#,#,#,#)
( ,0,1,0)q13(1,#,#,#,#)
( ,0,1,0,1)q13(#,#,#,#)
( ,0,1,0,1,#)q14(#,#,#)
( ,0,1,0,1,#,#)q14(#,#)
( ,0,1,0,1,#,#,#)q14(#)
( ,0,1,0,1,#,#,#,#)q14()
( ,0,1,0,1,#,#,#)q15(#)
ACCEPT

()q0(0,#)
( )q1(#)
( ,#)q2()
( )q3(#,0)
()q3( ,#,0)
( )q0(#,0)
( , )q7(0)
( )q12( ,#)
( ,0)q13(#)
( ,0,#)q14()
( ,0)q15(#)
ACCEPT
```

Turing Machine 3:

The third Turing machine example performs a subtraction of a unary encoding of two numbers. It is important to note that the first number must be greater than the second; otherwise this algorithm will not work. The input tape should be the two numbers, separated by a blank space. The algorithm works by removing a one from the beginning of the two numbers until the second number has no more ones to remove. After the algorithm is run, the first number will have the unary encoding of the subtraction and the second one will be all zeros. The machine description can be found in ourTM_3.xt.

The first input tests a small scale, regular subtraction of 2 – 1. The second one tests a subtraction of 3-3, to make sure the case where the answer is zero is taken care of. The third one attempts a subtraction with larger numbers to still make sure it works. All three cases perform as expected. The input tape can be found in ourTM_3-input.txt.

Input Tape:

```
3
1,1, ,1
1,1,1, ,1,1,1
1,1,1,1,1,1,1,1,1,1,1, ,1,1,1,1
```

Output:

```
3
1,1, ,1
1,1,1, ,1,1,1
1,1,1,1,1,1,1,1,1,1,1, ,1,1,1,1
()q1(1,1, ,1)
(0)q2(1, ,1)
(0,1)q2( ,1)
(0,1, )q3(1)
(0,1, ,0)q6()
(0,1, ,0, )q7()
ACCEPT

()q1(1,1,1, ,1,1,1)
(0)q2(1,1, ,1,1,1)
(0,1)q2(1, ,1,1,1)
(0,1,1)q2( ,1,1,1)
(0,1,1, )q3(1,1,1)
(0,1,1, ,0)q6(1,1)
(0,1,1, )q4(0,1,1)
(0,1,1)q4( ,0,1,1)
(0,1)q4(1, ,0,1,1)
(0)q5(1,1, ,0,1,1)
()q5(0,1,1, ,0,1,1)
(0)q1(1,1, ,0,1,1)
(0,0)q2(1, ,0,1,1)
(0,0,1)q2( ,0,1,1)
(0,0,1, )q3(0,1,1)
(0,0,1, ,0)q3(1,1)
```

```
(0,0,1, ,0,0)q6(1)
(0,0,1, ,0)q4(0,1)
(0,0,1, )q4(0,0,1)
(0,0,1)q4( ,0,0,1)
(0,0)q4(1, ,0,0,1)
(0)q5(0,1, ,0,0,1)
(0,0)q1(1, ,0,0,1)
(0,0,0)q2( ,0,0,1)
(0,0,0, )q3(0,0,1)
(0,0,0, ,0)q3(0,1)
(0,0,0, ,0,0)q3(1)
(0,0,0, ,0,0,0)q6()
(0,0,0, ,0,0,0, )q7()
ACCEPT

()q1(1,1,1,1,1,1,1,1,1,1,1, ,1,1,1,1)
(0)q2(1,1,1,1,1,1,1,1,1,1, ,1,1,1,1)
(0,1)q2(1,1,1,1,1,1,1,1,1, ,1,1,1,1)
(0,1,1)q2(1,1,1,1,1,1,1,1, ,1,1,1,1)
(0,1,1,1)q2(1,1,1,1,1,1,1, ,1,1,1,1)
(0,1,1,1,1)q2(1,1,1,1,1,1, ,1,1,1,1)
(0,1,1,1,1,1)q2(1,1,1,1,1, ,1,1,1,1)
(0,1,1,1,1,1,1)q2(1,1,1,1, ,1,1,1,1)
(0,1,1,1,1,1,1,1)q2(1,1,1, ,1,1,1,1)
(0,1,1,1,1,1,1,1,1)q2(1,1, ,1,1,1,1)
(0,1,1,1,1,1,1,1,1,1)q2(1, ,1,1,1,1)
(0,1,1,1,1,1,1,1,1,1,1)q2( ,1,1,1,1)
(0,1,1,1,1,1,1,1,1,1,1, )q3(1,1,1,1)
(0,1,1,1,1,1,1,1,1,1,1, ,0)q6(1,1,1)
(0,1,1,1,1,1,1,1,1,1,1, )q4(0,1,1,1)
(0,1,1,1,1,1,1,1,1,1,1)q4( ,0,1,1,1)
(0,1,1,1,1,1,1,1,1,1)q4(1, ,0,1,1,1)
(0,1,1,1,1,1,1,1,1)q5(1,1, ,0,1,1,1)
(0,1,1,1,1,1,1,1)q5(1,1,1, ,0,1,1,1)
(0,1,1,1,1,1,1)q5(1,1,1,1, ,0,1,1,1)
(0,1,1,1,1,1)q5(1,1,1,1,1, ,0,1,1,1)
(0,1,1,1,1)q5(1,1,1,1,1,1, ,0,1,1,1)
(0,1,1,1)q5(1,1,1,1,1,1,1, ,0,1,1,1)
(0,1,1)q5(1,1,1,1,1,1,1,1, ,0,1,1,1)
(0,1)q5(1,1,1,1,1,1,1,1,1, ,0,1,1,1)
(0)q5(1,1,1,1,1,1,1,1,1,1, ,0,1,1,1)
()q5(0,1,1,1,1,1,1,1,1,1,1, ,0,1,1,1)
(0)q1(1,1,1,1,1,1,1,1,1,1, ,0,1,1,1)
(0,0)q2(1,1,1,1,1,1,1,1,1, ,0,1,1,1)
(0,0,1)q2(1,1,1,1,1,1,1,1, ,0,1,1,1)
(0,0,1,1)q2(1,1,1,1,1,1,1, ,0,1,1,1)
(0,0,1,1,1)q2(1,1,1,1,1,1, ,0,1,1,1)
(0,0,1,1,1,1)q2(1,1,1,1,1, ,0,1,1,1)
(0,0,1,1,1,1,1)q2(1,1,1,1, ,0,1,1,1)
(0,0,1,1,1,1,1,1)q2(1,1,1, ,0,1,1,1)
```

```
(0,0,1,1,1,1,1,1,1)q2(1,1, ,0,1,1,1)
(0,0,1,1,1,1,1,1,1,1)q2(1, ,0,1,1,1)
(0,0,1,1,1,1,1,1,1,1,1)q2( ,0,1,1,1)
(0,0,1,1,1,1,1,1,1,1,1, )q3(0,1,1,1)
(0,0,1,1,1,1,1,1,1,1,1, ,0)q3(1,1,1)
(0,0,1,1,1,1,1,1,1,1,1, ,0,0)q6(1,1)
(0,0,1,1,1,1,1,1,1,1,1, ,0)q4(0,1,1)
(0,0,1,1,1,1,1,1,1,1,1, )q4(0,0,1,1)
(0,0,1,1,1,1,1,1,1,1,1)q4( ,0,0,1,1)
(0,0,1,1,1,1,1,1,1,1)q4(1, ,0,0,1,1)
(0,0,1,1,1,1,1,1,1)q5(1,1, ,0,0,1,1)
(0,0,1,1,1,1,1,1)q5(1,1,1, ,0,0,1,1)
(0,0,1,1,1,1,1)q5(1,1,1,1, ,0,0,1,1)
(0,0,1,1,1,1)q5(1,1,1,1,1, ,0,0,1,1)
(0,0,1,1,1)q5(1,1,1,1,1,1, ,0,0,1,1)
(0,0,1,1)q5(1,1,1,1,1,1,1, ,0,0,1,1)
(0,0,1)q5(1,1,1,1,1,1,1,1, ,0,0,1,1)
(0,0)q5(1,1,1,1,1,1,1,1,1, ,0,0,1,1)
(0)q5(0,1,1,1,1,1,1,1,1,1, ,0,0,1,1)
(0,0)q1(1,1,1,1,1,1,1,1,1, ,0,0,1,1)
(0,0,0)q2(1,1,1,1,1,1,1,1, ,0,0,1,1)
(0,0,0,1)q2(1,1,1,1,1,1,1, ,0,0,1,1)
(0,0,0,1,1)q2(1,1,1,1,1,1, ,0,0,1,1)
(0,0,0,1,1,1)q2(1,1,1,1,1, ,0,0,1,1)
(0,0,0,1,1,1,1)q2(1,1,1,1, ,0,0,1,1)
(0,0,0,1,1,1,1,1)q2(1,1,1, ,0,0,1,1)
(0,0,0,1,1,1,1,1,1)q2(1,1, ,0,0,1,1)
(0,0,0,1,1,1,1,1,1,1)q2(1, ,0,0,1,1)
(0,0,0,1,1,1,1,1,1,1,1)q2( ,0,0,1,1)
(0,0,0,1,1,1,1,1,1,1,1, )q3(0,0,1,1)
(0,0,0,1,1,1,1,1,1,1,1, ,0)q3(0,1,1)
(0,0,0,1,1,1,1,1,1,1,1, ,0,0)q3(1,1)
(0,0,0,1,1,1,1,1,1,1,1, ,0,0,0)q6(1)
(0,0,0,1,1,1,1,1,1,1,1, ,0,0)q4(0,1)
(0,0,0,1,1,1,1,1,1,1,1, ,0)q4(0,0,1)
(0,0,0,1,1,1,1,1,1,1,1, )q4(0,0,0,1)
(0,0,0,1,1,1,1,1,1,1,1)q4( ,0,0,0,1)
(0,0,0,1,1,1,1,1,1,1)q4(1, ,0,0,0,1)
(0,0,0,1,1,1,1,1,1)q5(1,1, ,0,0,0,1)
(0,0,0,1,1,1,1,1)q5(1,1,1, ,0,0,0,1)
(0,0,0,1,1,1,1)q5(1,1,1,1, ,0,0,0,1)
(0,0,0,1,1,1)q5(1,1,1,1,1, ,0,0,0,1)
(0,0,0,1,1)q5(1,1,1,1,1,1, ,0,0,0,1)
(0,0,0,1)q5(1,1,1,1,1,1,1, ,0,0,0,1)
(0,0,0)q5(1,1,1,1,1,1,1,1, ,0,0,0,1)
(0,0)q5(0,1,1,1,1,1,1,1,1, ,0,0,0,1)
(0,0,0)q1(1,1,1,1,1,1,1,1, ,0,0,0,1)
(0,0,0,0)q2(1,1,1,1,1,1,1, ,0,0,0,1)
(0,0,0,0,1)q2(1,1,1,1,1,1, ,0,0,0,1)
(0,0,0,0,1,1)q2(1,1,1,1,1, ,0,0,0,1)
```

```
(0,0,0,0,1,1,1)q2(1,1,1,1, ,0,0,0,1)
(0,0,0,0,1,1,1,1)q2(1,1,1, ,0,0,0,1)
(0,0,0,0,1,1,1,1,1)q2(1,1, ,0,0,0,1)
(0,0,0,0,1,1,1,1,1,1)q2(1, ,0,0,0,1)
(0,0,0,0,1,1,1,1,1,1,1)q2( ,0,0,0,1)
(0,0,0,0,1,1,1,1,1,1,1, )q3(0,0,0,1)
(0,0,0,0,1,1,1,1,1,1,1, ,0)q3(0,0,1)
(0,0,0,0,1,1,1,1,1,1,1, ,0,0)q3(0,1)
(0,0,0,0,1,1,1,1,1,1,1, ,0,0,0)q3(1)
(0,0,0,0,1,1,1,1,1,1,1, ,0,0,0,0)q6()
(0,0,0,0,1,1,1,1,1,1,1, ,0,0,0,0, )q7()
ACCEPT
```

Any Major difficulties Encountered or notable implementation decision

An important implementation decisions was first deciding which programming language to use. Working with a scripting language like Python or Perl would have made parsing through the text files easier. However, our familiarity with C++ and the STL made us choose C++. Using a vector of vectors of strings to create a table that will keep the transition information was especially important to be able to consistently access the transitions.

Most difficulty was experienced in developing the Turing Machines and successfully translating a complicated diagram into a machine description.