MAT 3670: Lab 2
Turing Machines and Bit Manipulation

Background

As explained in Chapter 2 of our text, the *bit* (i.e., the binary digit — 0 or 1) is the basic unit of information. For this lab, you will design and test several Turing machines which manipulate sequences of bits (also known as bit strings) in ways that are related to computer arithmetic.

Pre-Lab Exercises

Before arriving for this week's laboratory, carefully consider each of the following exercises. Time invested to obtain good designs, in the case of exercises 2 through 4, will pay off later, as little or no debugging may be needed during your time in the lab.

1. Refer to Figure 1 (on page 3), which depicts a two's complement system. Next to each circle, place the numerical value associated with the 4-bit code. For example, the code 0011 is used to represent 3 in this system. Similarly 1110 is used for -2. Later on, you will draw arrows on this diagram, but you don't need to worry about that now.

2. Design a Turing machine which will transform an input bit string into its complement, replacing 0's with 1's and 1's with 0's. Initially, the read/write head is under the leftmost bit of the input string. Upon completion, the read/write head should return to the leftmost bit and halt in a final state.

3. In binary arithmetic the idea of a *successor* is a fundamental idea — it's like adding one to a given integer. Let α be an arbitrary bit string. If α has only 1's, then the successor of α is the string with an equal number of 0's. Otherwise, there is at least one 0 bit within α, so we can focus our attention on the rightmost 0, writing $\alpha = \beta 01 \ldots 1$. In this case, the successor of α is $\beta 10 \ldots 0$. In other words, in the "tail end" of the string, the rightmost 0 is changed to a 1 and all of the remaining 1's get turned into 0's. For example, if $\alpha = 001010111$, its successor is 001011000.

   Design a Turing machine which will compute the successor of a bit string. As before, assume that the read/write head is initially positioned at the leftmost bit of the input string. Upon completion, the read/write head should be at the leftmost bit of the output string.

4. Design a Turing machine which combines the two previous machines. Given a bit string, your machine should first complement it, then obtain the successor of this complement. For example, given input string 0100, the desired result is 1100. (Since $0100 \to 1011 \to 1100$.) This operation is known as the *two's complement* operation: starting with n, the final result is denoted $TC(n)$.

Laboratory Exercises

1. Login and create a `lab2` folder.

2. Using JFLAP, enter your design for the complement Turing machine. Test it and make any necessary repairs. Save your final design in `lab2a.jff`.

3. Using JFLAP, enter your design for the successor Turing machine. Test it and make any necessary repairs. Save your final design in `lab2b.jff`.

4. Using JFLAP, enter your design for the complement/successor Turing machine. Test it and make any necessary repairs. Save your final design in `lab2c.jff`. Complete the table in Figure 2 using JFLAP's multiple run feature. Ensure your Turing machine produces correct results for each of these inputs.

5. Every row of Figure 2 provides two values: n and its two's complement value, $TC(n)$. Using the information from your table, draw an arrow from n to $TC(n)$ on the diagram of Figure 1. For example, since $TC(0110) = 1010$, an arrow from 0110 to 1010 appears.

6. Using your completed diagram, what can you say about the numerical value of $n + TC(n)$? (Use the associated decimal values, not the 4-bit patterns.) Examine all 16 arrows to see if your observation holds for each. Make an accurate statement about this, putting it in a file named `README`.

7. Adding to `README`, explain the designs of each of your Turing machines.

8. Before leaving the lab, please show me your completed diagram of Figure 1.

Submissions

Drop your `lab2` folder on the EIU submission icon.

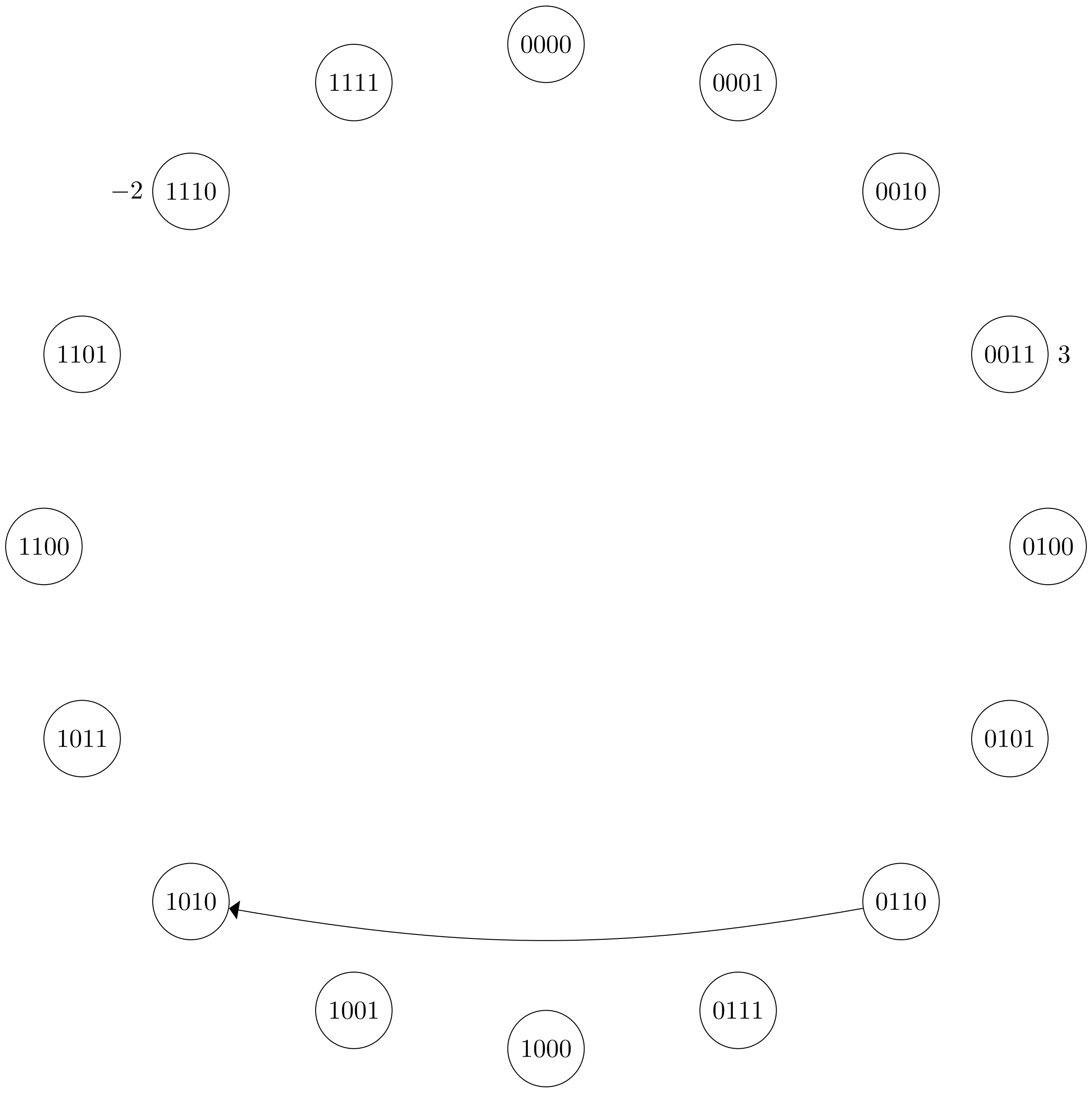

Figure 1: A diagram to help explain the 4-bit two's complement system. A portion of this lab asks you to add to this diagram. First, next to each circle place the numerical value associated with its 4-bit code. Then, for each value n, draw an arrow from n to $TC(n)$.

Input string	Output string
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

Figure 2: Test data for the complement/successor Turing machine.