

Turing Machine example to add two numbers

This is a Turing machine for binary addition for two numbers, x and y, on a tape in that order. For example (a restriction is that the leftmost digit must be a zero to work properly), call the first number on the tape x (the 0010) and the second number y (the 0011). The \$ character is at the start of the tape and separates the two numbers. The rest of the tape has blanks on it.

On tape initially: \$0010\$0011 // adding 2+3
 When done: \$0000\$0101 // equals 5

```
// algorithm repeatedly subtracts one from x, adds one to y until x is zero
repeat until x == 0, then HALT {
    // use T2 below, like a function call, start at leftmost $
    subtract 1 from x

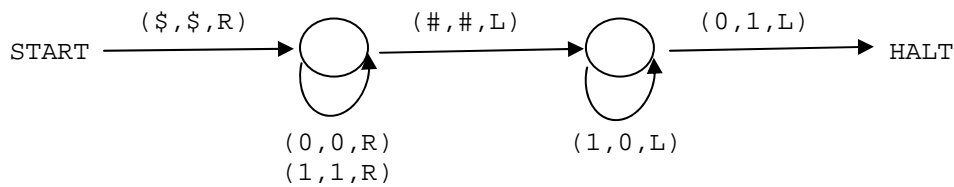
    // use T1 below, like a function call, start at middle $
    add 1 to y

    // simple TM to read and write the same character
    go back to the first $
}
```

The problem is solved by two Turing machines (functions), one to add one to a number and another to subtract one.

Algorithm for adding one (Turing Machine T1)

1. When starting at left end, walk to right end of number
2. Walking right to left, change all 1's to 0's
3. At first 0, change to 1



Algorithm for subtracting one (Turing Machine T2)

1. Take 1's complement, i.e., change all 1's to 0, all 0's to 1
2. Add one to the number
3. Take the 1's complement again

