Theory of Computation, CSCI 438 spring 2022
More practice building Turing Machines, March 23rd

1. Create a Turing machine "procedure" that:
    a. Inserts a \$ (where \$∉Σ, the alphabet of the language) to the front of a tape, shifting the non-blank contents of the tape one place over to the right.
    b. Returns the read/write head to the front of the tape, pointing at the first input symbol after the \$ (the second cell of the tape).

    You can assume that this procedure will only be called at the beginning when there are only symbols from Σ or blank on the tape.

High Level Plan

Shift each character one place to the right, inserting a \$ in the first position. Return the read/write head to the second cell on the tape.

Low Level Plan

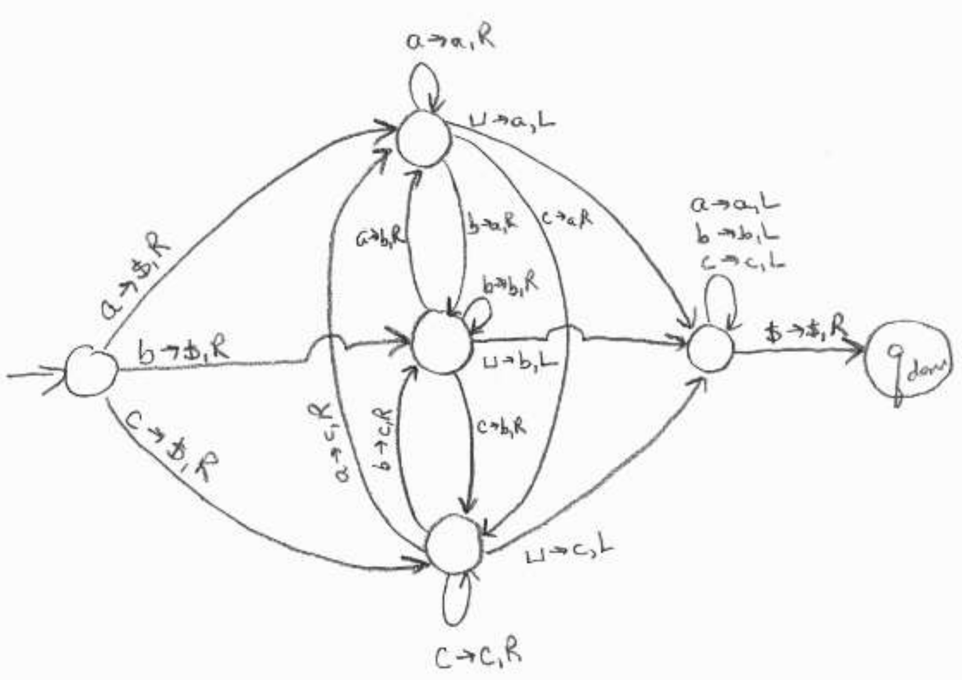
Read and remember the first character, replace it with a \$ and move right.
Loop:

> Traveling right until blank is reached, remember the current character and write the previously remembered character.

Travel left across a's and b's until hit \$, move right.

Machine

I'll use the alphabet Σ={a,b,c}. This can be modified for any alphabet.



2. Create a Turing Machine that accepts $L = \{0^{2\wedge n} : n\geq 0\}$. (This is 0 raised to the 2^n power) (This is example 3.7, page 171, in text.)

High Level Plan

Use the following algorithm with input m, where m is the length of the input string (i.e. the number of 0s).

```
loop
{
        if (m==0) reject
        else if (m==1) accept
        else if m is odd, reject
        else m:=m/2
}
```

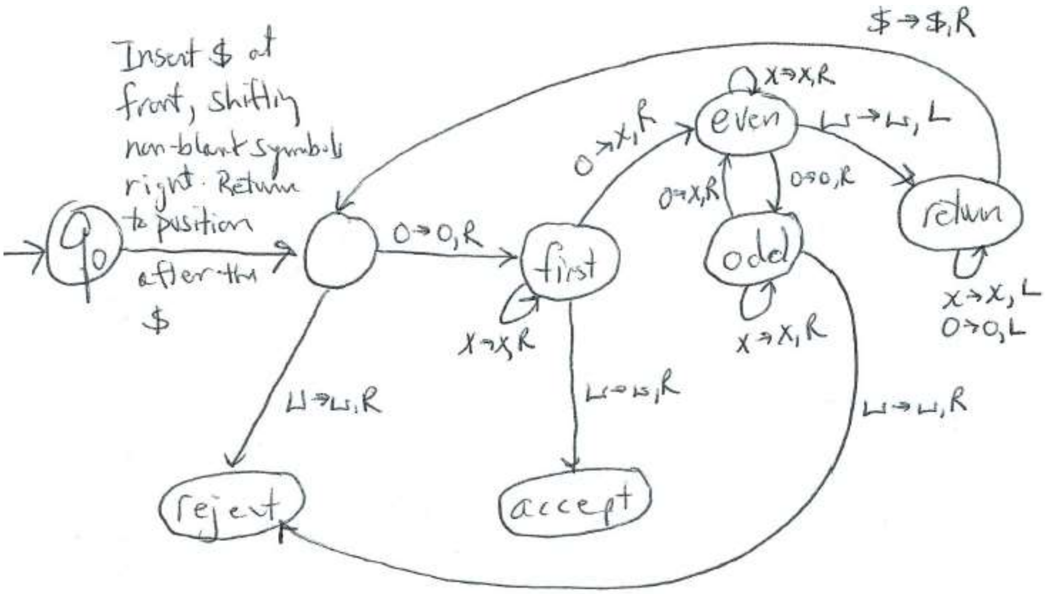
Low Level Plan

Insert a $ at the front of the tape, shifting all non-blank symbols one position to the right, and returning to point to the character to the right of the $.

```
Loop
{
        If blank, reject
        If only one 0 on the tape, ignoring Xs, accept

        // Remove half of the 0s, by replacing them with Xs
        Move right across the 0s and Xs, going from an even to an odd state,
        counting 0s, and replacing every other 0 with an X. If encounter blank in
        an odd state, reject. Else, when hit blank, continue.

        Return to the front of the tape.
}
```


Insert $ at front, shifting non-blank symbols right. Return to position after the $
q0
0→0,R
first
X→X,R
⊔→⊔,R
0→X,R
even
X→X,R
0→0,R
0→X,R
odd
X→X,R
⊔→⊔,R
⊔→⊔,L
return
X→X,L
0→0,L
$→$,R
⊔→⊔,R
reject
accept