1. What problem the machine tackled. Was it a recognizer, decider, or computation problem?

This machine is a computation Turing Machine. It takes an input string over the alphabet {x, y, z} and outputs a new string with every x removed, preserving the order of the remaining characters. The machine always halts in the accept state once the whole input has been run.

2. What, if any, was the reference from which the problem solved by the machine was drawn.

There was no reference for this problem. I designed the Turing Machine based on the idea of tape editing operations used in basic Turing Machines.

3. How does the machine work, in words.

The machine uses two tapes:

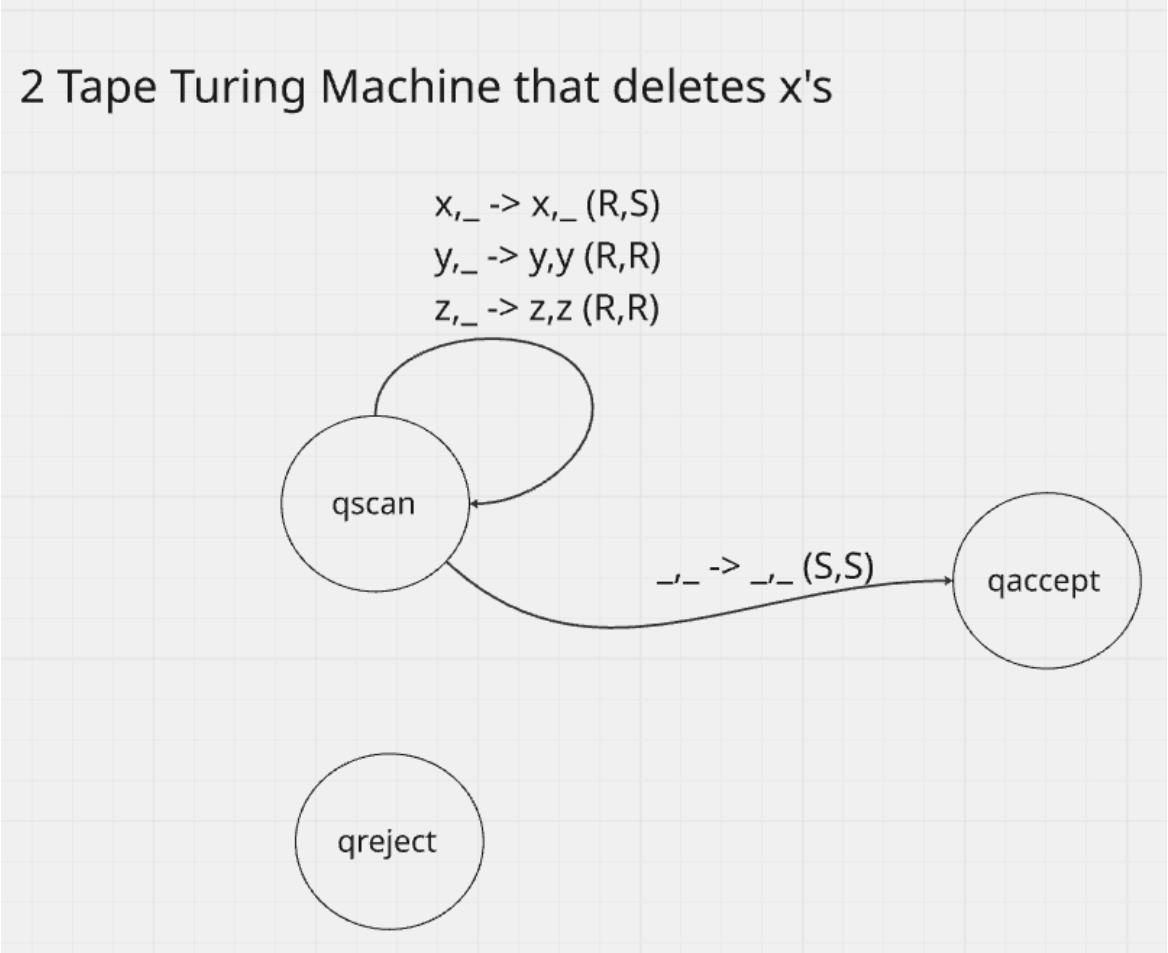
- Tape 1 contains the original input string
- Tape 2 starts blank and is used to build the output

The machine scans Tape 1 from the left:

- If it reads x. It ignores it and continues
- If it reads y or z, it copies that character onto Tape 2 and advances the Tape 2 head
- This continues until Tape 1 reaches the end
-

The machine always halts because Tape 1 is scanned exactly once from left to right

4. A state diagram.



5. How did you verify correct operation.

I verified the correct operation by running a test file with a variety of input string including: Strings with no x's, strings with one x, string with many x's in different positions. I ran the machine in my simulator and confirmed that the final tape always contained the input string with all of the x's removed. On my github repo there is a TM2-24418873.txt file with all of the transitions and information needed for it to run the TM2-24418873-tape.txt file with all of the test cases. The output is in the results-TM2-24418873-tape.txt