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

This machine is a decider for the language:

$L = \{ w \in \{0,1\} \mid \text{no two adjacent characters are equal} \}^*$

This is a decider because it halts on every input to check whether it ends in a accept state or a reject state

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

There was no reference for this problem but I used the classic example of repetition of languages and adapted into a Turing Machine that checks if two adjacent characters are equal

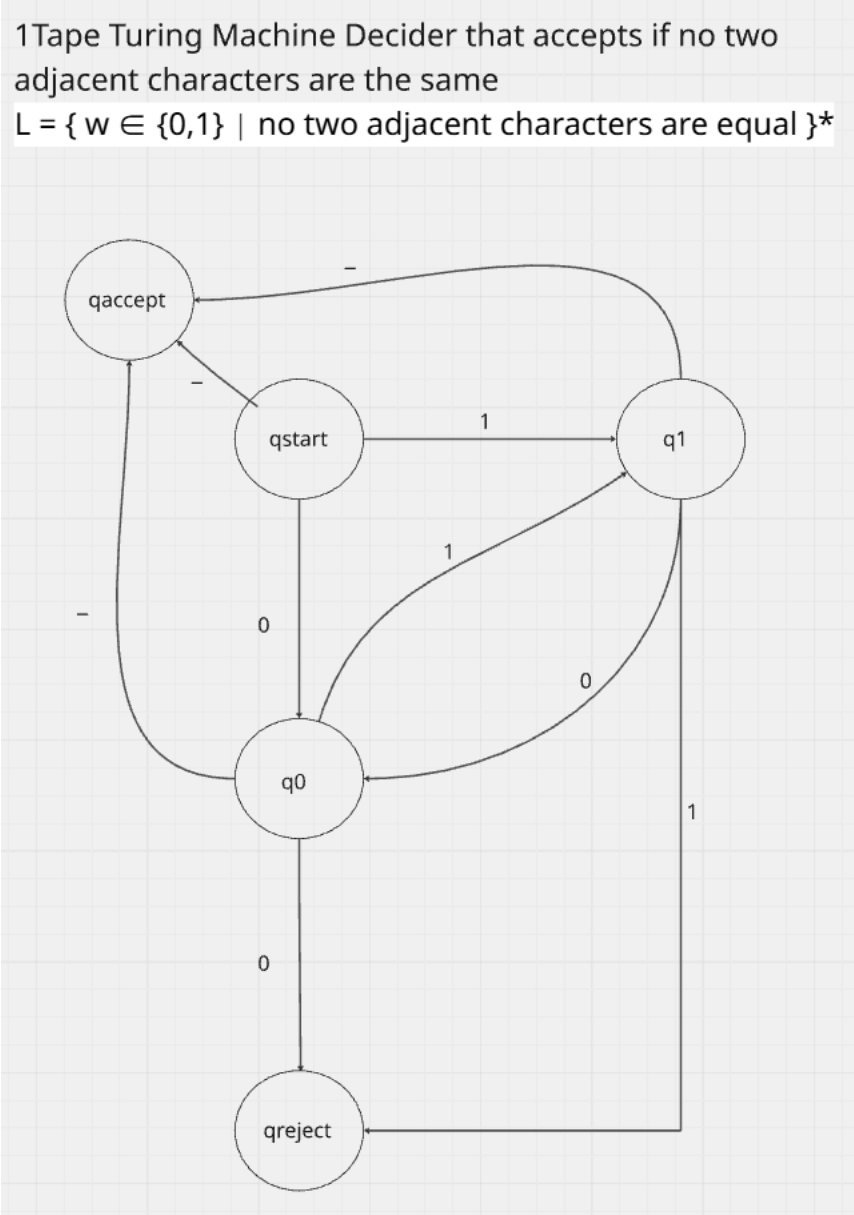
3. How does the machine work, in words.

The machine begins scanning the input from the left from tape.and keeps track of the previous symbol:

- If it sees the same symbol twice in a row it rejects it
- If it sees alternating symbols it continues scanning
- If it reaches the blank symbol _ without finding a repeated pair it accepts.

The machine always halts because it either detects two equal adjacent characters or reaches the end of the tape

4. A state diagram.




5. How did you verify correct operation.

I verified the correct operation by running a test file containing both valid and invalid string. I checked all strings with no adjacent repeats were accepted, and all strings with adjacent equal characters were rejected. On the github their is a TM1d-24418873.txt file with all of the transitions and information needed for it to run the TM1-24418873.txt file with all of the test cases. The output is in the results-TM1-24418873.txt