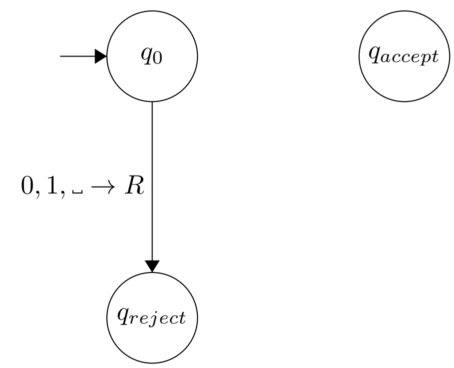
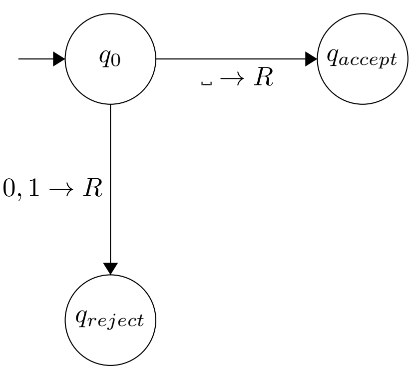
Problem 1

Total: 40 points (10 points each)
Draw the state-transition diagram for a Turing Machine M that *decides* each of the following languages. That is, M is supposed to accept all strings in the language and reject all strings not in the language, but it can never loop forever on any input (your TM **must** halt)! Assume that the input alphabet is $\Sigma = \{0, 1\}$ and the tape alphabet $\Gamma = \Sigma \cup \{\textvisiblespace\}$, where $\textvisiblespace$ is the blank symbol.

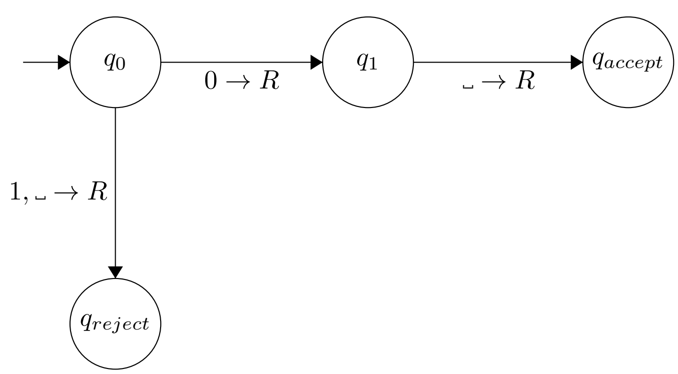
1. $A = \varnothing = \{\}$ Solution:



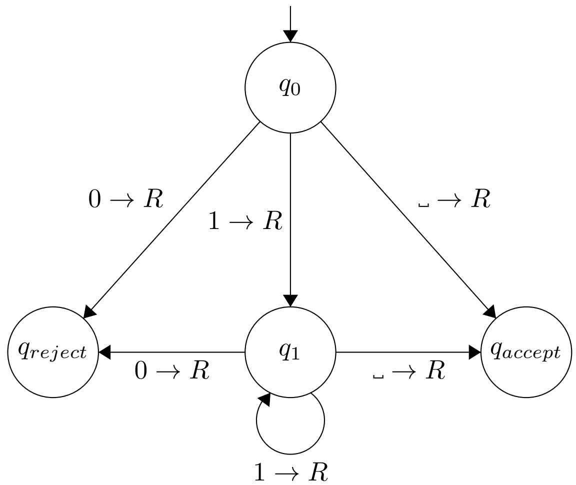
2. $B = \{\epsilon\}$ Solution:



3. $C = \{0\}$ Solution:



4. $D = \{1\}^*$ Solution:



Problem 2

Total: 30 points

Show a Turing Machine M that decides the following language.

$$E = \{yy \mid y \in \{0,1\}^*\}$$

1. Describe in English the idea of how it works. It must be detailed enough as an *algorithm.* **5 points**
   - *Bad:* M operates by finding the midpoint of the input string.
   - *Good:* Explain **how** M finds the midpoint.
2. Show all the transitions by drawing the state diagram (label all states!). *Also, indicate groups of states with their purposes.* For example, states $q_n, \ldots, q_m$ finds the midpoint of the input string. You may omit the reject state. **10 points**
3. In addition, M must restore the input tape to its original state, i.e., any markers you have used should revert back to the original input symbols. **5 points**

Solution - Idea: We will need following additional characters - a, b, α, β, X, and Y.

We will split the spring into two halves by marking the first half with a, b and the second half with α, β.

1. At the start, we are at the leftmost 0 or 1. Replace it with a (if 0) or b (if 1) and go to the rightmost 0 or 1. *Note:* If we can't find the rightmost 0 or 1, i.e., we reach a symbol $\sqcup, \alpha,$ or β, then the length of the input string is odd and it will be rejected.
2. Now, we are at the rightmost 0 or 1. Replace it with α (if 0) or β (if 1) and go back to the leftmost 0 or 1.
3. Repeat 1-2 until all 0s and 1s are marked.
4. Return to the left end of the tape.

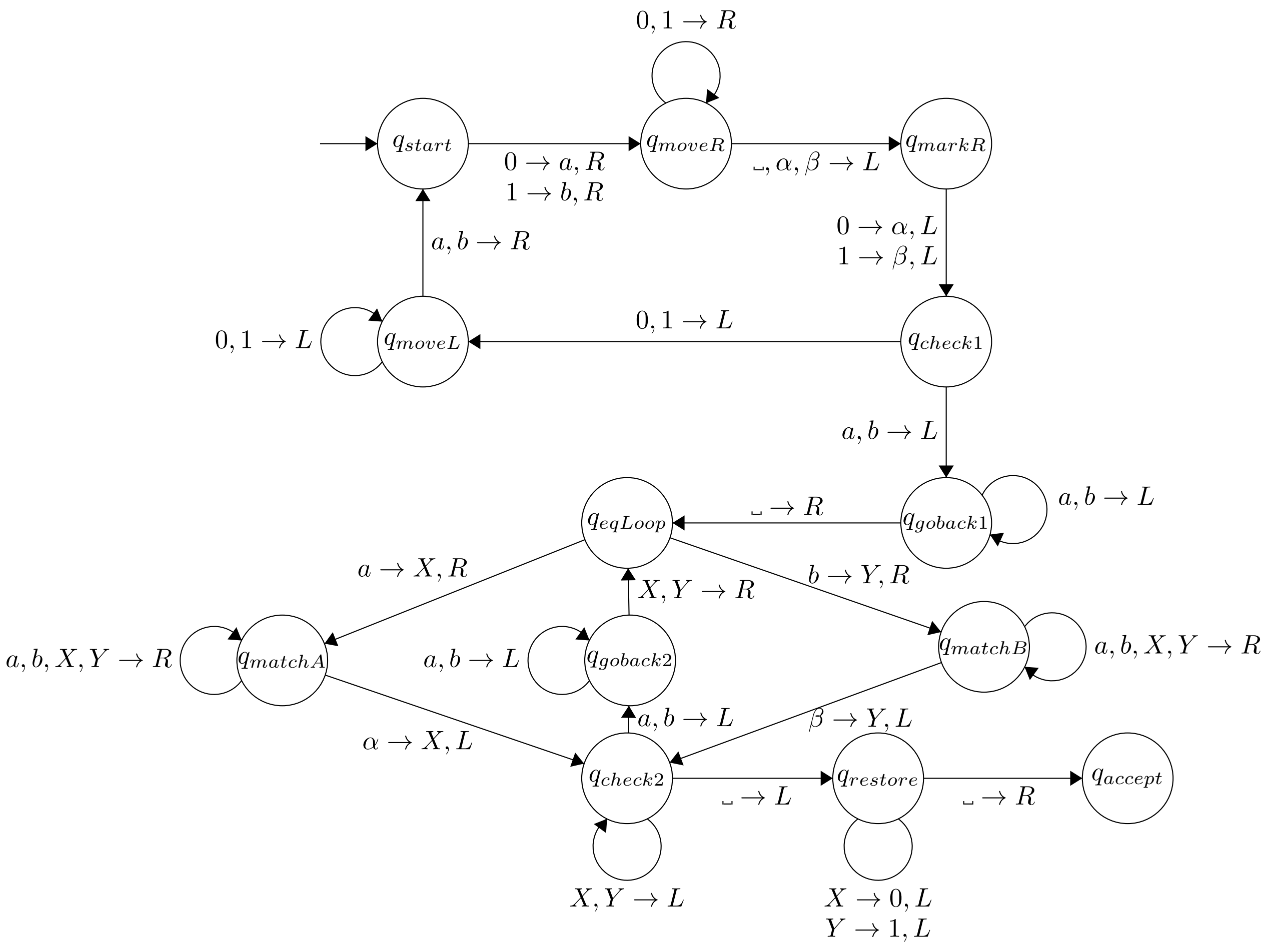
Now we can easily distinguish between halves of the string. Next, we will compare these two halves.

1. Replace (cross off) the leftmost a with X or b with Y. Remember which symbol was crossed off and move to the leftmost α or β.
2. If the leftmost α or β matches the crossed-off symbol from the left half of the string, then cross it off with X (if α) or Y (if β) as well and return to the leftmost a or b. If it was NOT a match, then reject the string

3. Repeat 1-2 until all symbols are crossed off.

Since we already performed the length check, our TM should reach ␣ on the tape if there is no mismatch in the string, upon which the string should be accepted after the tape is restored with the input string.

Solution - State diagram:



Explanation: To simplify the state diagram, the tape is open-ended on both sides.

The rectangular loop at the top is the first phase described in the Idea part, splitting the input strings into two halves. At q_{check1}, the TM determines whether to continue the first phase or move to the next phase.

The second phase starts with q_{eqLoop} where the TM will loop to match the corresponding symbols. q_{matchA} is the designated state to remember that we have seen a and must match it to α and q_{matchB} remembers that we have seen b and attempt to match it to β. The matched symbols are crossed off from the tape by replacing them with $*$. At q_{check2}, if all symbols are crossed off, then the left half of the string must equal the right half of the string so the TM restores the input string on the tape with $q_{restore}$ before entering the accept state.

To simplify the diagram, some transitions and the reject state were hidden. Make no mistake! The vanilla TM is **deterministic** and thus requires transitions for *all* state and symbol pairs and *all* transitions *to* the reject state to be defined. Below is the list of omitted transitions from the diagram.

1. $q_{start}␣ \rightarrow q_{accept}R$: An empty string is a member of E. There is a transition from q_{start} to q_{accept} on a blank symbol.

2. $q_{markR}a \rightarrow q_{reject}R$ and $q_{markR}b \rightarrow q_{reject}R$: Odd-length strings will cause the TM to enter the reject state.

3. $q_{matchA}\beta \rightarrow q_{reject}R$ and $q_{matchB}\alpha \rightarrow q_{reject}R$: Mismatch in the string will cause the TM to enter the reject state.