1. **Problem 1**

   Let $L_1 \subset L_2$.

   (a) a If L_1 is a regular language, then is L_2 necessarily a regular language?

   > **Solution:** No. Counterexample: L_1 only accepts the string 01, L_2 is the language 0^n1^n.

   (b) b If L_2 is a regular language, then is L_1 necessarily a regular language?

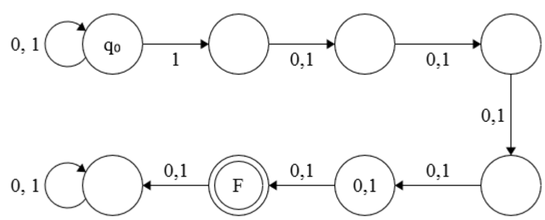
   > **Solution:** No. Counterexample: L_2 accepts all possible strings, L_1 is the language 0^n1^n.

2. **Problem 2**

   Consider a regular language L that accepts a string if the 6th to last bit is a 1 (over an alphabet of $\{0, 1\}$.

   (a) Construct an NFA that recognizes L.

   > **Solution:**



   (b) Argue that any DFA which recognizes L must have at least 64 states.

   > **Solution:** Suppose for the sake of contradiction that there exists a DFA N that recognizes L but has strictly less than 64 states. Consider the 6-bit binary prefix string of an input. There are $2^6 = 64$ different possibilities, so by pigeonhole principle there exists two prefix strings P_a and P_b that arrive at the same state in N. Since they are different strings, they must differ in at least one bit. Consider the greatest i for which P_a and P_b differ at bit i. Concatenate i 0's to both P_a and P_b. The augmented strings will arrive at the same state after N reads each prefix, and therefore will also arrive at the same state after N reads the suffix, which is identical in both strings. However, N should reject one and accept the other. Contradiction.

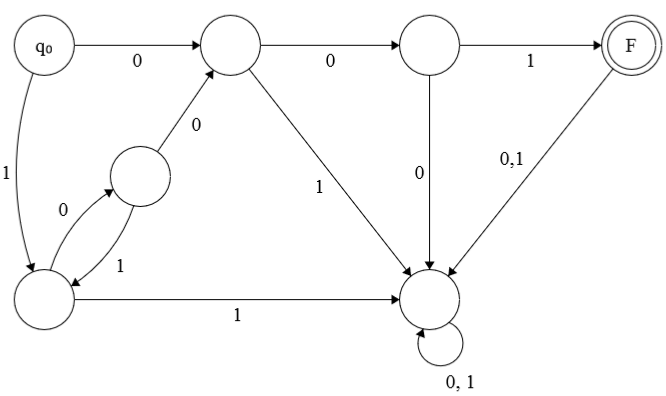
3. **Problem 3**

   Given a DFA for L, provide a formal construction for a DFA that recognizes L*. In other words, provide the 5-tuple that characterizes Kleene(L) $= \{w_1...w_k | k \geq 0$ and each $w_i \in L\}$.

Solution: Let the DFA that recognizes L be called N and the DFA that recognizes Kleene(L) be called M. We will construct M as follows. The alphabet Σ is unchanged. The finite set of states Q is the power set of the states in N. The initial state in M is the singleton of the initial state in N. If the initial state of N is not an accept state, then let the singleton of the initial state be an accept state in M. This allows M to accept the empty string. Also, let any state in M that "contains" an accept state in N be an accept state. Finally, the transition function of M applies the transition function of N to every state contained by the current state in M in order to generate a set of states (which are represented by a state in M). If the current state is an accept state, add the initial state to this set of states.

4. **Problem 4**

Draw a DFA that accepts the regular language represented by the regular expression ((10)*)001.



Solution: