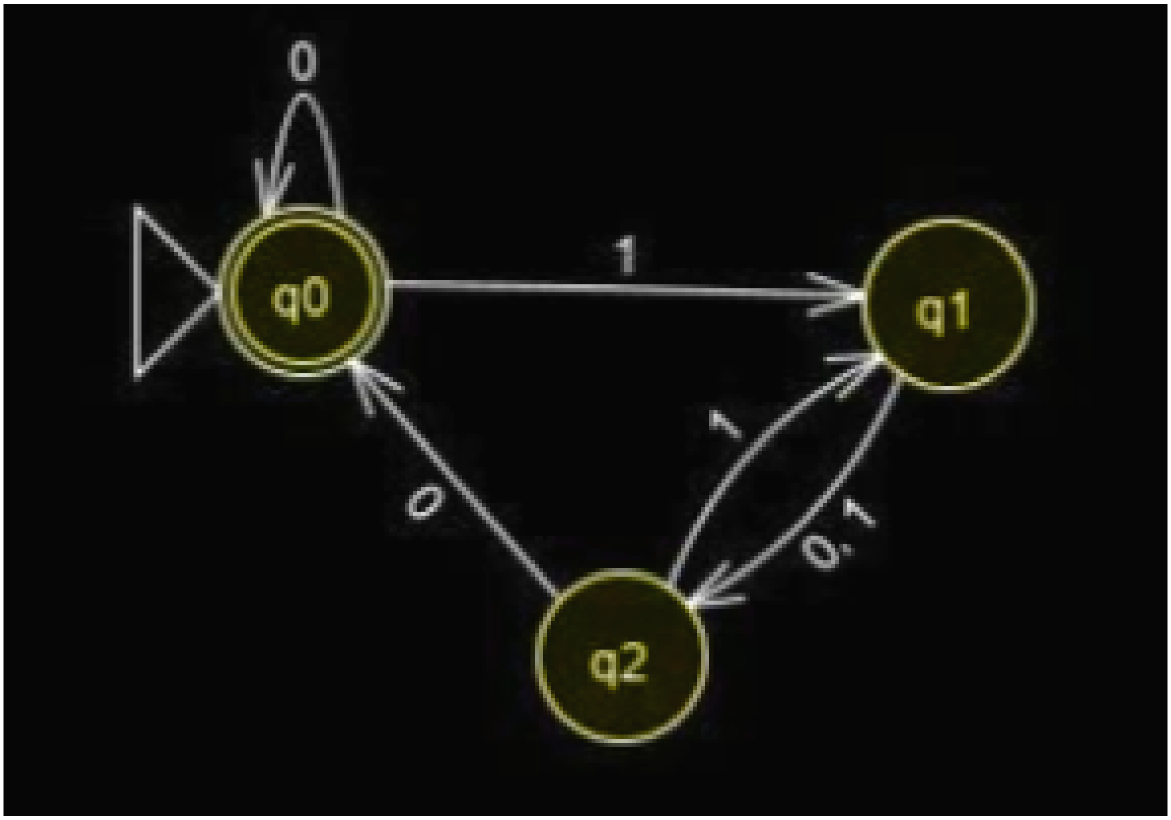
1. [6 pts] Answer questions for the following DFA M and give reasons for your answers.



1.1) Is $< M, 0100 > \in ADFA$? (Note: $ADFA$ represents the acceptance problem of DFA)

- **Yes, because if you plug in 0100 into the above DFA, the path taking is as follows: q0 > q0 > q1 > q2 > q0 > accepted.**

1.2) Is $< M, 011 > \in ADFA$?

- **No, because if you plug in the input string 011 into the above DFA, the path taking is as follows q0 > q0 > q1 > q2 > rejected.**

1.3) Is $< M > \in ADFA$?

- **No, because nothing is passed into the above DFA. The input should be <M, w> where w is some input string**

1.4) Is $< M, 0100 > \in AREX$?

- **No, because M is in DFA form which is not in the correct form for AREX.**

1.5) Is $< M > \in EDFA$?

- **No, because M is a DFA that is not empty so it is not in the set of EDFA**

1.6) Is $< M, M > \in EQDFA$?

- **Yes, because the two DFAs are passed into the EQdfa and M = M.**

2. [7 pts] Assume $\Sigma = \{0, 1\}$. Given a DFA C, does there exist an algorithm to decide whether $L(C) = \Sigma*$? Express this problem as a language denoted as $ALLDFA$ and prove that $ALLDFA$ is decidable (Hints: build a Turing decider).

The language for the above problem is M = {<C> | C is a DFA and L(C) = Σ∗}. M is decidable. We can use proof by constructions to determine this. TM F "on input <C>"

1. **We can build a DFA D that accepts Σ∗**
2. **We can run TM E (EQ DFA) on <C, D> and we will get the following**
   a. **If E accepts, F is accepted**
   b. **If E rejects, F is rejected**

3. [7 pts] Given two DFAs A and B, we consider the problem of deciding whether L(A) (language of A) is a subset of L(B). Express this problem as a language denoted as $SUBDFA$ and prove that $SUBDFA$ is decidable.

If L(A) is in the subset L(B), the following must also be true: L(A) intersected with !L(B) must be empty. This is true because if if L(A) is in !L(B) that means that it is not in L(B) which makes L(A) not in the subset of L(B). We can make a language V = {" On input <A, B> where A and B are regular expressions"}

1. **We can construct a DFA M such that L(M) = !L(B) intersected with L(A)**
2. **We must run TM T on input W, where T decides EmptyDFA**
3. **If T accepts, then it is accepted and if T rejects, then it is rejected.**