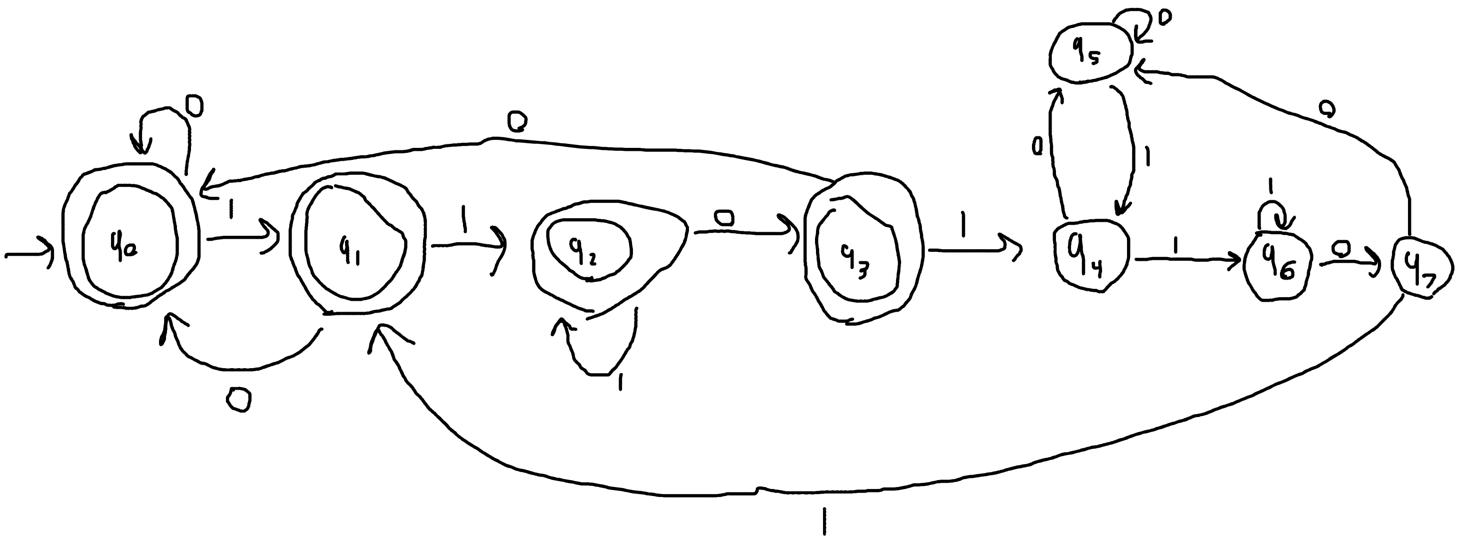
Question 1

Thursday, September 26, 2024 5:25 PM

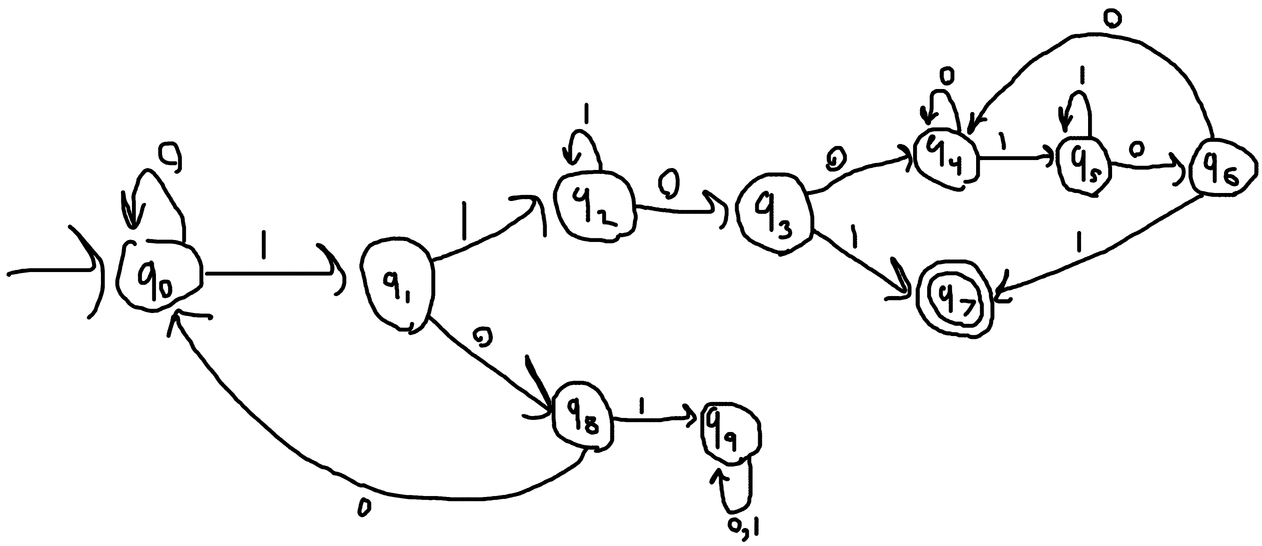
Q1 Devise a DFA that accepts $L = \{s \in \{0,1\}^* : \text{The number of occurrences of 1101 in } s \text{ is zero or even}\}$. Note that the occurrences of 1101 can overlap, i.e., 1101101 has 2 occurrences of 1101 that are overlapping and hence $1101101 \in L$. However, $1101101101 \notin L$. **[4 points]**



Question 2

Thursday, September 26, 2024 7:18 PM

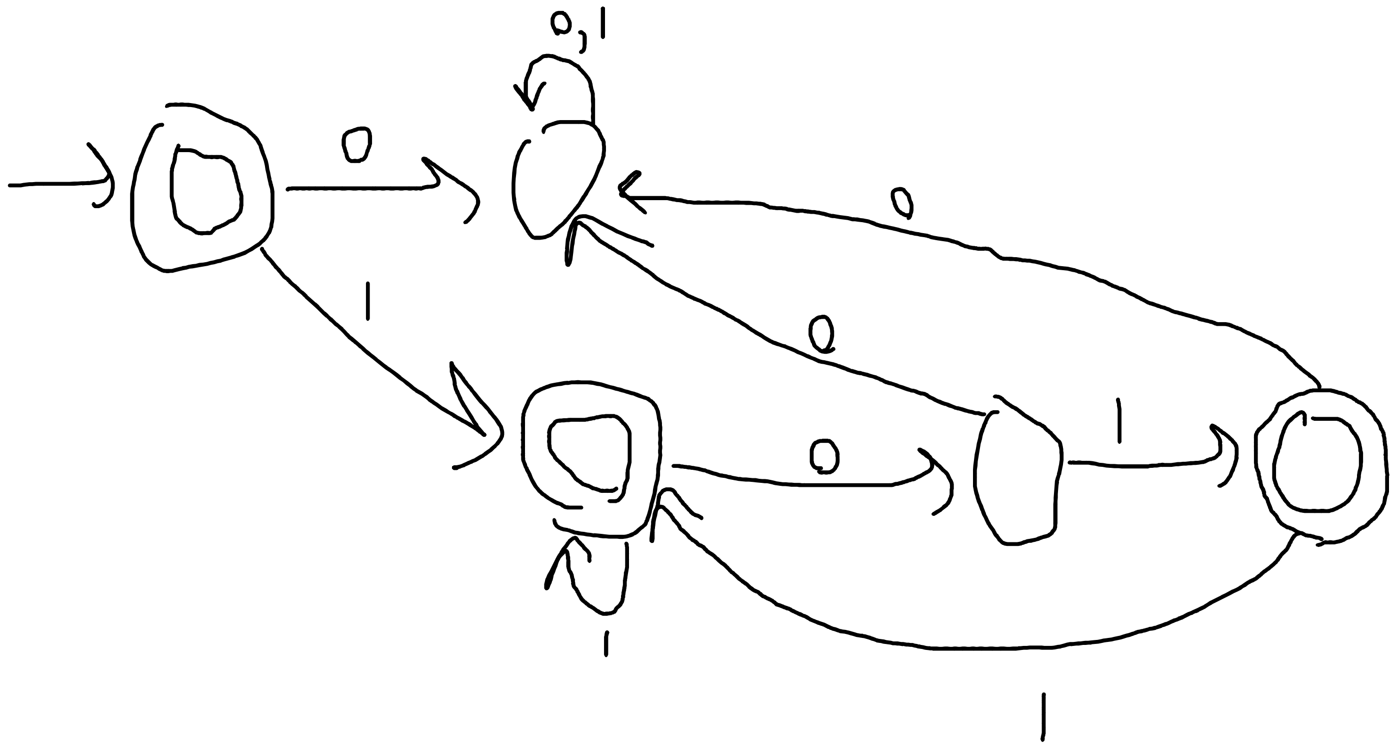
Q2 Devise a DFA that accepts the language of all binary strings s such that: (a) s contains 110 and 101 as substrings; and (b) the first occurrence of 110 precedes the first occurrence of 101. **[4 points]**



Question 3

Thursday, September 26, 2024 11:11 PM

Q3 Devise a DFA that accepts $L = \{1, 101, 1101\}^*$. **[4 points]**

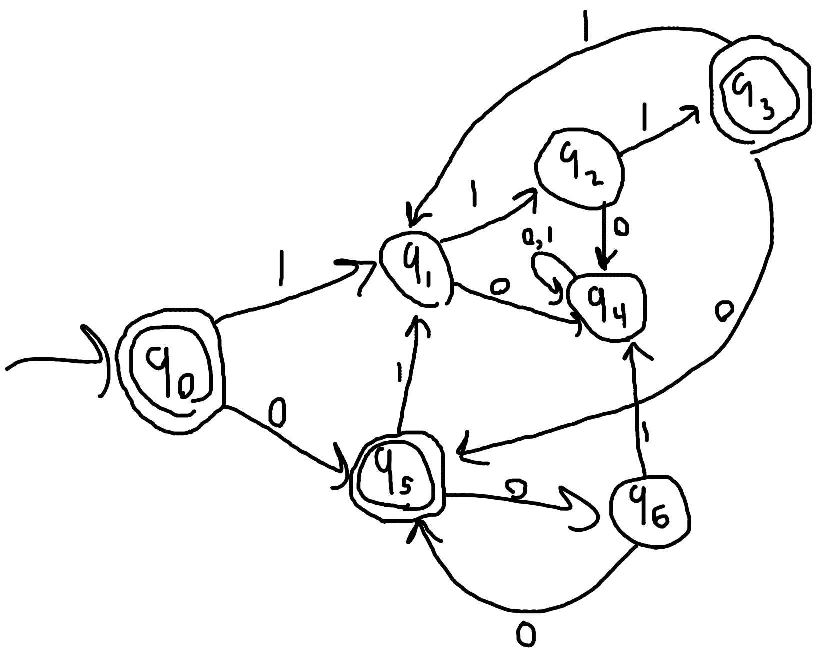


Question 4

Thursday, September 26, 2024 11:12 PM

Q4 Devise a DFA that accepts: **[4 points]**

$L = \{s \in \{0,1\}^* :$ every zero run in s is of odd length, and every one run in s has a length that is a multiple of $3\}$.



Question 5

Thursday, September 26, 2024 11:12 PM

Q5 You are given two black boxes that implement two DFAs. You can interact with the two boxes (individually) by presenting to it any string (of your choice) and the boxes will respond with accept or reject. You can interact with the two boxes any number of times by presenting input strings, but you have no way to look into the transition diagrams of the implemented DFAs. It it is know that the implemented DFAs are both 10 state DFAs, can you determine in finite time conclusively that the two DFAs are identical? [**4 points**]

Our definition of identical is being able to feed the same string into both black boxes (DFAs) and returning a corresponding reject or fail state for both DFAs, testing all combinations is the method to prove they're identical.

However, since our language isn't specified, obviously proving a reject or accept is a lot easier with just combinations of 0's or 1's (binary string) in a DFA than proving all possible combinations for a language containing an infinite number of characters. On that note, if we have an infinite language, it would not be possible to prove two DFAs are identical in finite time. This is because the number of possible combinations is $\lim_{n\to\infty}(n^0 + n^1 + \cdots + n^9) = \infty$.

But, assuming our language consists of binary strings (0's and 1's) and we're bound to 10 state DFAs, then the number of possible combinations is $2^0 + 2^1 + \cdots + 2^9$, which results in a finite time to conclude that the two DFAs are identical.