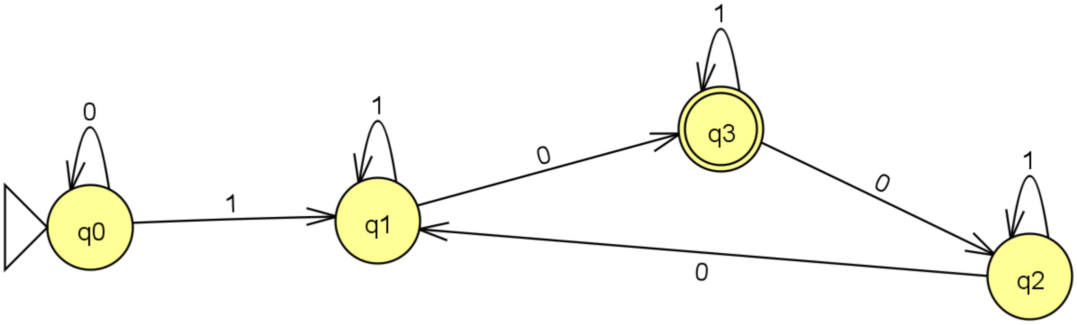
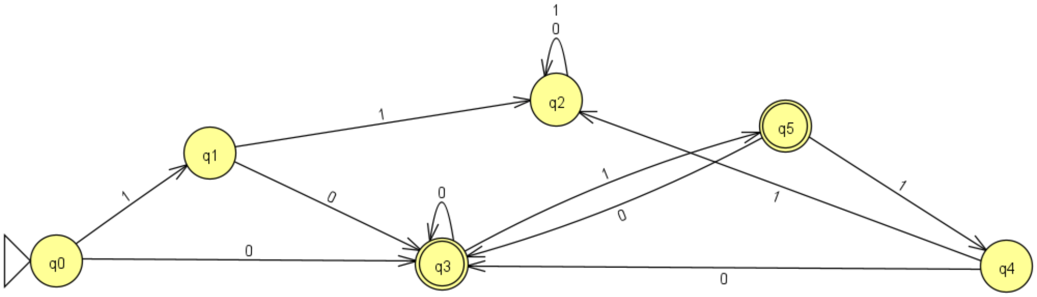
1. For each of the following languages, use JFLAP (http://www.jflap.org) to draw the state diagram of a deterministic finite automaton (DFA) that accepts the language. Also give the formal definition of the DFA in terms of the 5-tuple (Q, Σ, δ, q_0, F). *20 points*

a) $L_1 = \{w \in \{0,1\}^* | w$ starts with a 1 and its number of 0s is a multiple of three$\}$



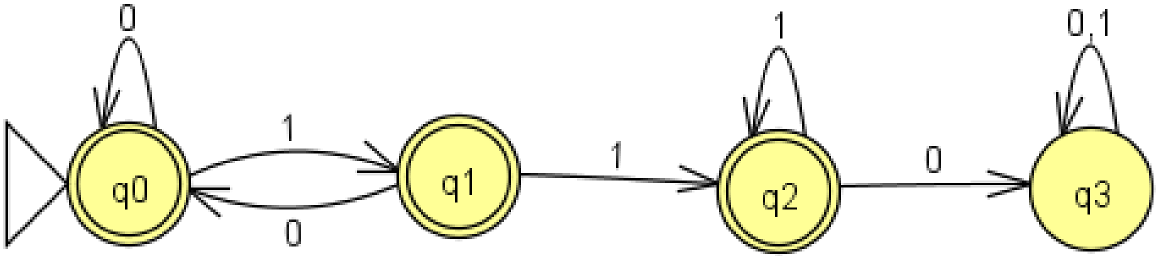
- Q = q0, q1, q2, q3
- Σ = (0,1)
- δ = dfa above
- Q0 = q0
- F = q1

b) $L_2 = \{w \in$
$\{0,1\}^* |$ in which every 1 is either immediately preceded or immediately followed by 0. such as 0110, 101, 0$\}$



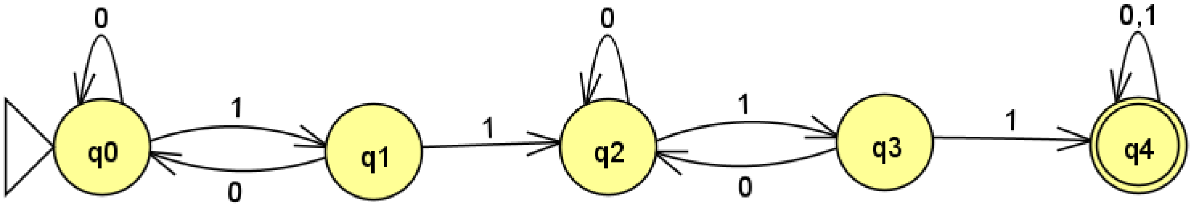
- Q = q0, q1, q2, q3, q4, q5
- Σ = (0,1)
- δ = dfa above
- Q0 = q0
- F = q3,q5

c) $L_3 = \{w \in \{0,1\}^* | w \text{ does not start with } 110\}$



- Q = q0, q1, q2, q3,
- Σ = (0,1)
- δ = dfa above
- Q0 = q0
- F = q0, q1, q2

d) $L_4 = \{w \in \{0,1\}^* | \text{ in which substring 11 occurs at least twice}\}$

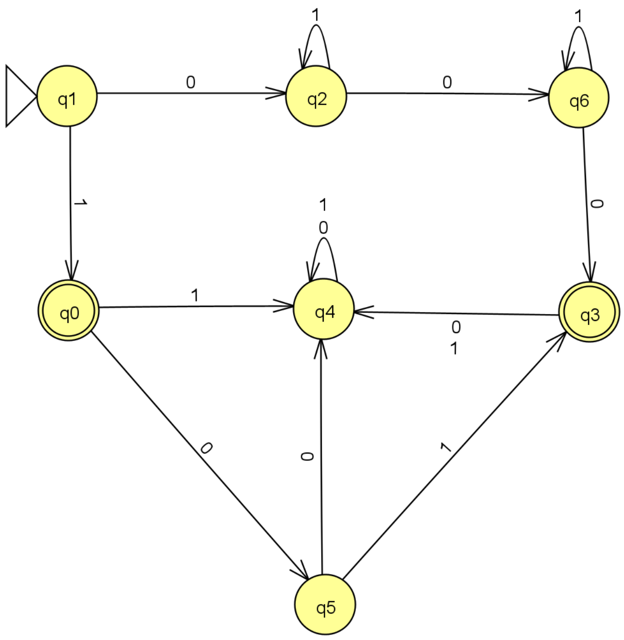


- Q = q0, q1, q2, q3, q4
- Σ = (0,1)
- δ = dfa above
- Q0 = q0
- F = q4

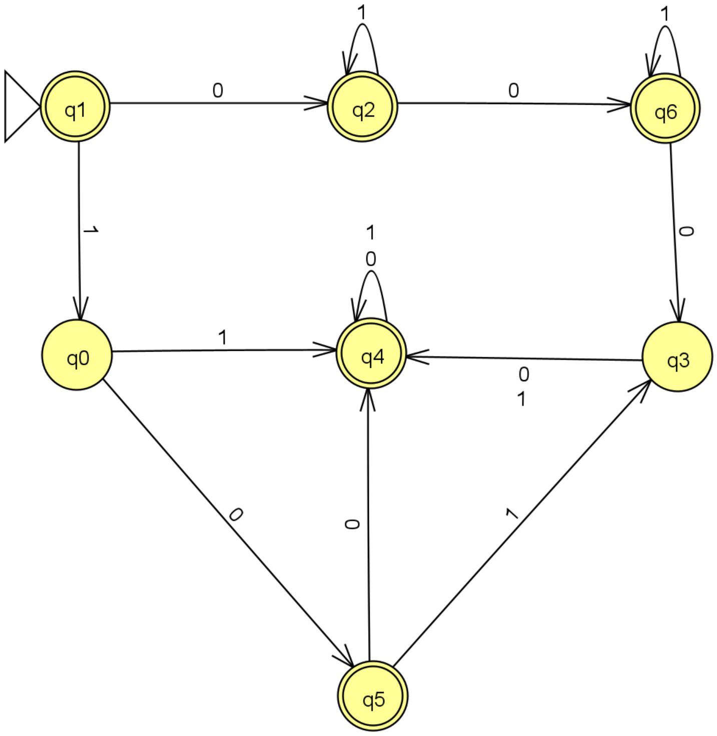
2. The language below is a complement to a simpler language. First, identify the simpler language and give the state diagram of the DFA that recognizes it. Then, use it to give the state diagram of the DFA that recognizes the language below. *10 points*

$$L_5 = \{w \in \{0,1\}^* | w \text{ is any string except } 1, 101, \text{ and } 000\}$$

W is not any string except 1, 101, and 000


q1
q2
q6
q0
q4
q3
q5
0
0
1
1
1
1
0
0
1
0
1
0
0
1

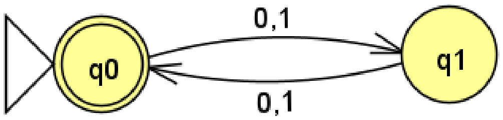
ANSWER


q1
q2
q6
q0
q4
q3
q5
0
0
1
1
1
1
0
0
1
0
1
0
0
1

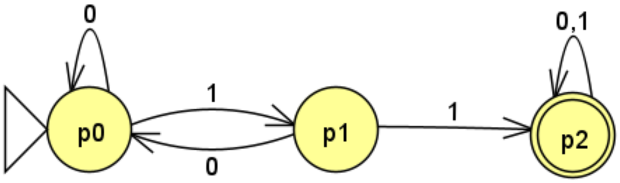
3. The language below is the intersection of two simpler languages. First, identify the simpler languages and give the state diagrams of the DFAs that recognize them. Then, use the product construction from the proof of Theorem 1.25 in the book to build a DFA that recognizes the language specified below; give its state diagram before and after simplification if there are any unneeded states or states that can be combined. *10 points*

$$L_6 = \{w \in \{0,1\}^* | w \text{ has even length AND do not contain substring } 11\}$$

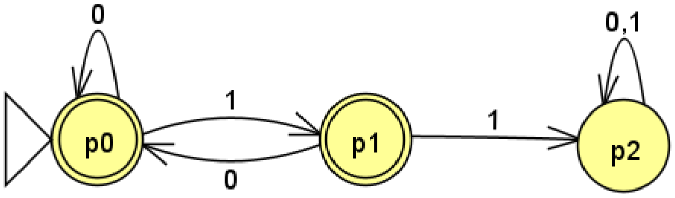
W has an even length



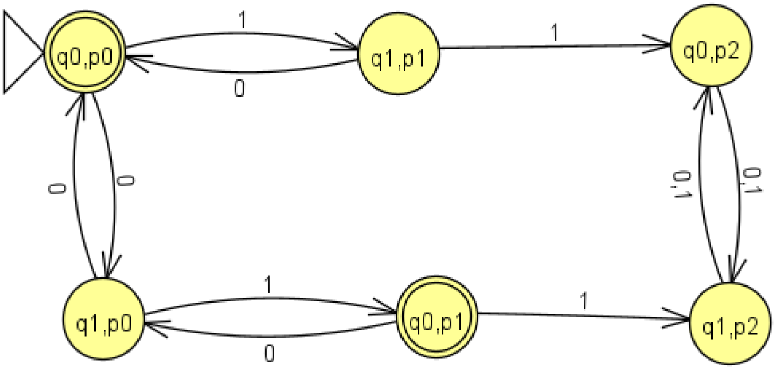
W contains substring 11



W does not contain substring 11



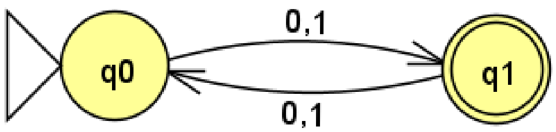
ANSWER



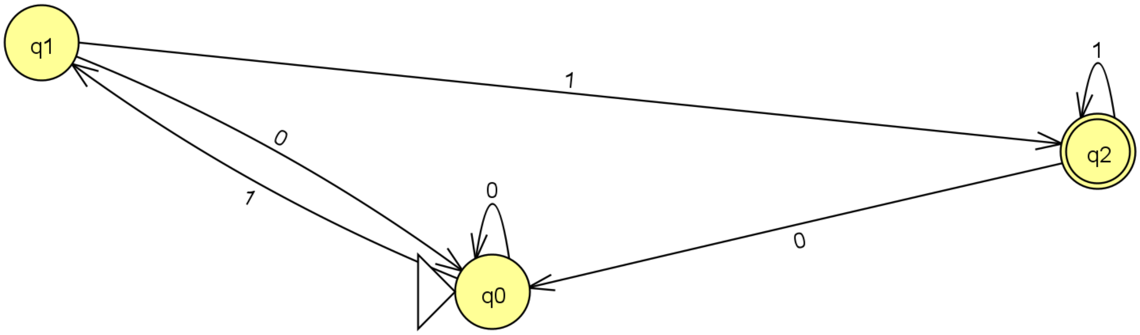
4. The language below is the union of two simpler languages. First, identify the simpler languages and give the state diagrams of the DFAs that recognize it. Then, use the product construction from the proof of Theorem 1.25 (PDF pp.66) to give the state diagram of the DFA that recognizes the language below. *10 points*

$$L_7 = \{w \in \{0,1\}^* | w \text{ has odd length OR end with } 111\}$$

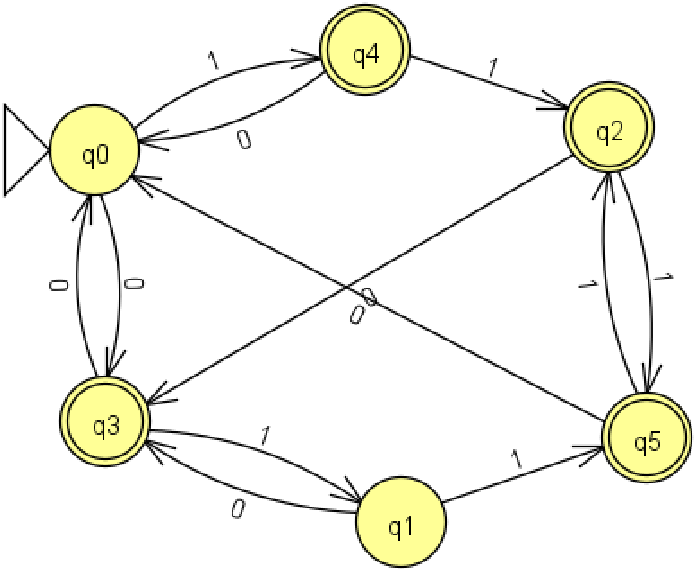
W has an odd length



W ends with 11

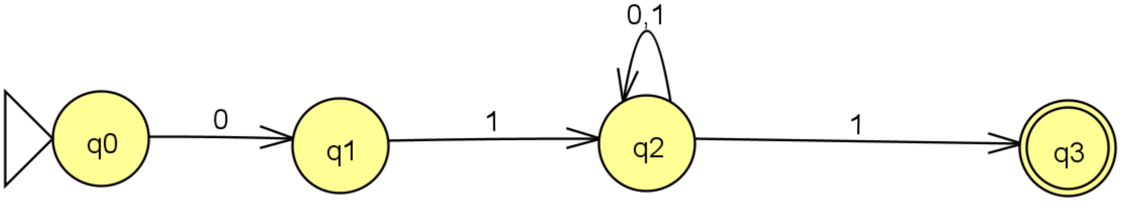


ANSWER



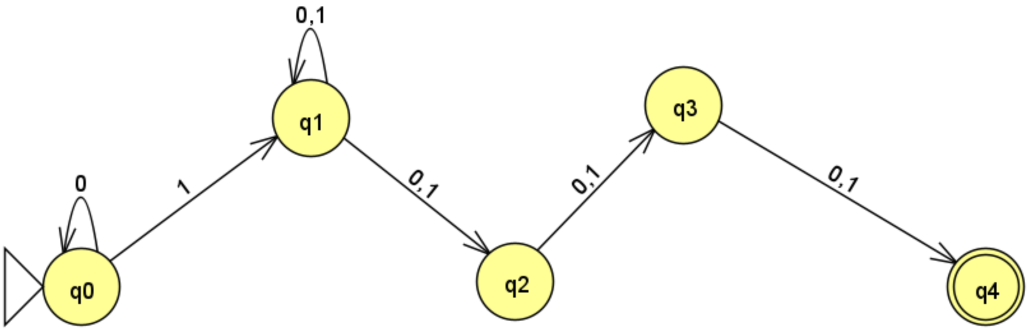
5. For each part, give the formal definition *and* state diagram (http://www.jflap.org) for a nondeterministic finite automaton (NFA) that accepts the specified language. *20 points*

a) $L_8 = \{w \in \{0,1\}^* | \mathit{01} \text{ is a prefix of } w \text{ and } \mathit{1} \text{ is a suffix of } w\}$



- Q = q0, q1, q2, q3
- Σ = (0,1)
- δ = nfa above
- Q0 = q0
- F = q3

b) $L_9 = \{w \in \{0,1\}^* | \text{ forth} - \text{to} - \text{the} - \text{last symbol of w is 1 }\}$

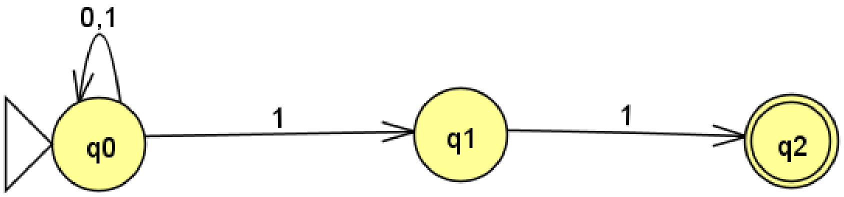


- Q = q0, q1, q2, q3,q4
- Σ = (0,1)
- δ = nfa above
- Q0 = q0
- F = q4

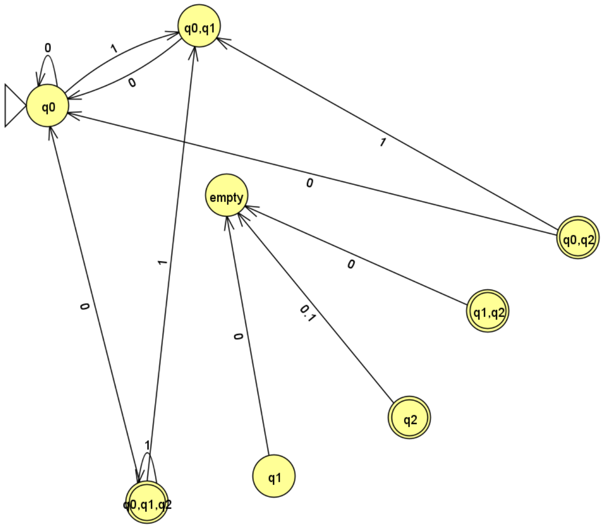
6. First, give the state diagram (http://www.jflap.org) for the NFA that recognizes the language below using no more than 3 states. Next, use the powerset construction from the proof of Theorem 1.39 in the book to convert the NFA into a DFA. If there are any unneeded states or states that can be combined, you may simplify your DFA, but show your DFA's state diagram before and after simplification. *10 points*

$$L_{10} = \{w \in \{0,1\}^* | 11 \text{ is a suffix of } w\}$$

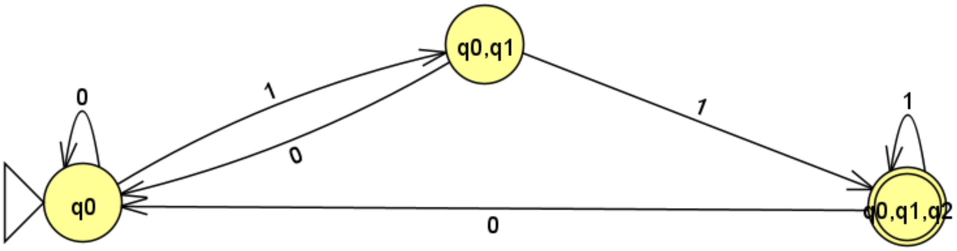
NFA:

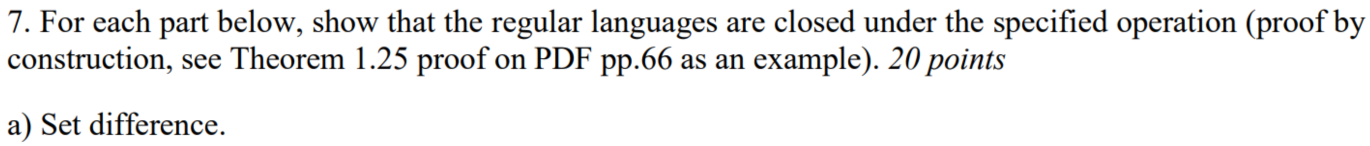


DFA



DFA Simplified:





7. For each part below, show that the regular languages are closed under the specified operation (proof by construction, see Theorem 1.25 proof on PDF pp.66 as an example). *20 points*

a) Set difference.

We must prove that regular languages are closed under set differenc. We need to show that if we have two regular languages, L1 and L2, whos set differential, L1 - L2, is also a regular language.

Proof by construction:

Assume that L1 & L2 are regular languages. This means that there exists deterministic finite automata (DFAs) that recognize them. We then onstruct a DFA that recognizes the set difference L1 - L2.

Start with DFAs for both L1 and L2:

- DFA1 for L1 = (Q1, Σ, δ1, q1, F1)
- DFA2 for L2 = (Q2, Σ, δ2, q2, F2)

We'll create a new DFA for L1 - L2 as follows:

The states of the new DFA is the Cartesian product of the states of DFA 1 and 2, denoted as Q = Q1 × Q2.
The alphabet Σ remains the same for the new DFA.

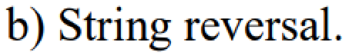
The transition function δ will be defined as follows:

- δ((q1, q2), a) = (δ1(q1, a), δ2(q2, a)) for all q1 in Q1, q2 in Q2, and a in Σ.
- The start state of the new DFA will be the pair (q1, q2) where q1 is the start state of DFA1, and q2 is the start state of DFA2, i.e., (q1, q2) = (q1, q2).

The set of accepting states F for the new DFA will be defined as follows:

- F = {(q1, q2) | q1 in F1 and q2 not in F2}

We have constructed a DFA that recognizes the set difference L1 - L2. The DFA has a finite # of states and satisfies the properties of a DFA, making it a regular language. Therefore, regular languages are closed under set difference.



b) String reversal.

We must prove that regular languages are closed under string reversal. We need to show that if we have a regular language L, then its reversal, L'R, must also be a regular language.

Proof by construction:

L will be a regular language recognized by a DFA M = (Q, Σ, δ, q0, F). We will to construct a DFA that recognizes the reversal of L, denoted as L'R. We start with the DFA M that recognizes L. Create a new DFA M'R = (Q, Σ, δ'R, q0, F'R) to recognize L'R:

Alphabet Σ remains the same.
The Σ remains the same.
Q0 will be the same.

The set of accepting states F'R is the set of states from which the original DFA M could reach an accepting state in one or more transitions.

- The transition function δ'R as follows: $\delta'R(q, a) = \{p \mid \delta(p, a) = q \text{ for some } a \text{ in } \Sigma\}$

The new DFA M'R recognizes the reversal of L, L'R. Because, we have constructed a DFA for L'R, which has a finite number of states AND it satisfies the properties of a DFA, we have shown that regular languages are closed under string reversal.

Therefore, regular languages are closed under both set difference and string reversal.