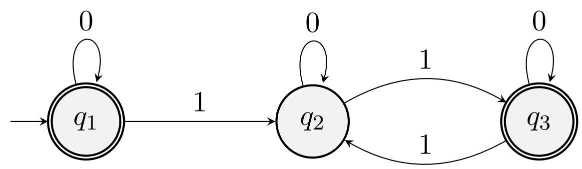
1. Find a regular expression for each of the following languages on $\{0, 1\}$:

   (a) Words not ending in 11.

   (b) Words of even length, length at least 3 and with second letter equal to 0.

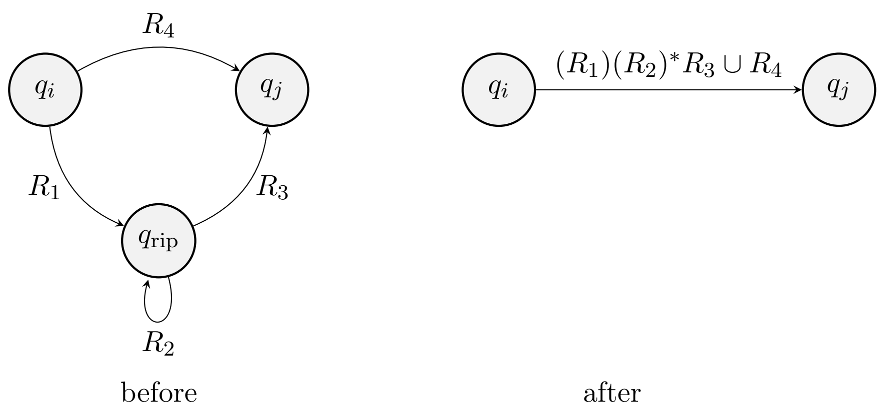
2. Let A be the following DFA:



   (a) Convert A into an NFA having (i) an initial states with no transitions leading into it, and (ii) a single accept state having no transitions of it. The resulting NFA should have five states, including the initial state q_0 and the unique accepting state q_4.

   (b) Rip out state q_1, and draw the resulting GNFA. The transitions will be labeled with regular expressions.

   (c) Recall the general prescription for ripping out a state given in the text:



   Letting $q_i = q_0$, $q_j = q_3$, and $q_{\text{rip}} = q_2$, identify R_1, R_2, R_3 and R_4. What regular expression labels the transition from q_0 to q_3 after ripping out q_2? What obvious simplification can you make involving the empty set?

   (d) Similarly, letting $q_i = q_3$, $q_j = q_3$, and $q_{\text{rip}} = q_2$, identify R_1, R_2, R_3 and R_4. What regular expression labels the transition from q_3 to q_3 after ripping out q_2?

   (e) Rip out q_2 and draw the resulting GNFA

   (f) Rip out q_3 to produce a GNFA with two states.

   (g) What is the resulting regular expression for the language accepted by A?

   (h) Give a simple description of $L(A)$ in words.

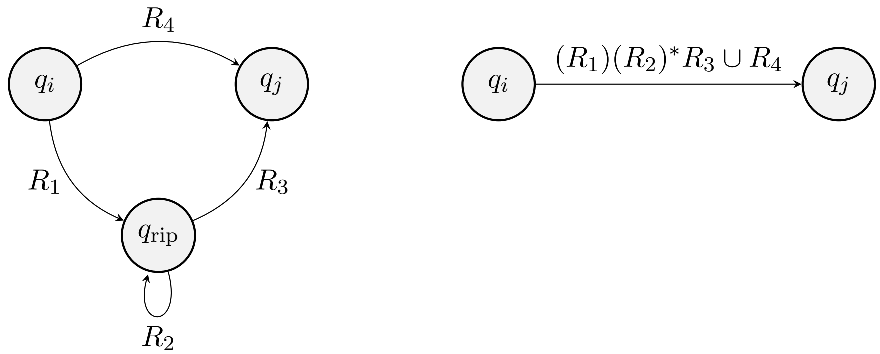
3. Negate the DFA from Problem the previous problem by exchanging accept and reject states to create the DFA A^c.

   (a) What is $L(A^c)$, in words.

(b) Convert A^c into an NFA having (i) an initial states with no transitions leading into it, and (ii) a single accept state having no transitions of it. The resulting NFA should have five states, including the initial state q_0 and the unique accepting state q_4.

(c) Rip out q_1.

(d) The resulting GNFA has states q_0, q_2, q_3, q_4. Recall the general prescription for ripping out a state given in the text:



before　　　　after

Letting $q_i = q_2$, $q_j = q_4$, and $q_{\text{rip}} = q_3$, identify R_1, R_2, R_3 and R_4. What regular expression labels the transition from q_2 to q_4 after ripping out q_3? How does that regular expression simplify?

(e) Rip out q_3.

(f) Rip out q_2.

(g) What is the resulting regular expression for $L(B) = L(A^c)$.

4. (Bonus) The text describes a method for converting an NFA with n states into a DFA with 2^n states. Show that this bound is roughly tight. Specifically, show that for every n there exists a language that can be recognized with an $(n+1)$-state NFA but cannot be recognized by a DFA with fewer than 2^n states.

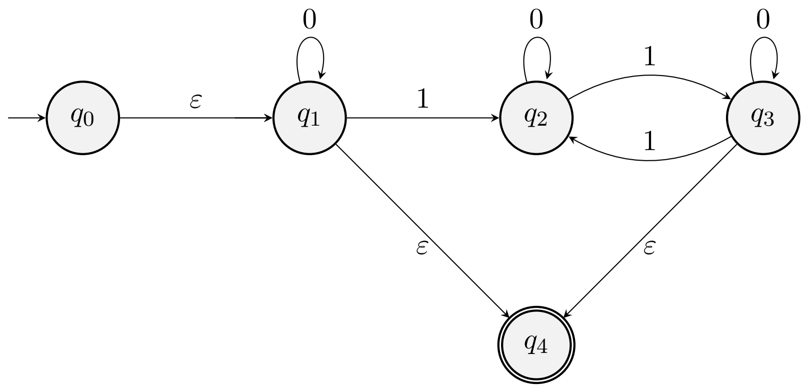
Solutions.

1. (a) $\Sigma^*(0 \cup 01)$.

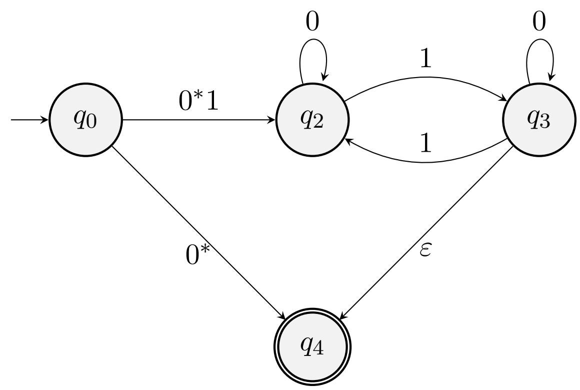
   (b) $(0 \cup 1)0(0 \cup 1)(0 \cup 1)((0 \cup 1)(0 \cup 1))$.

2.

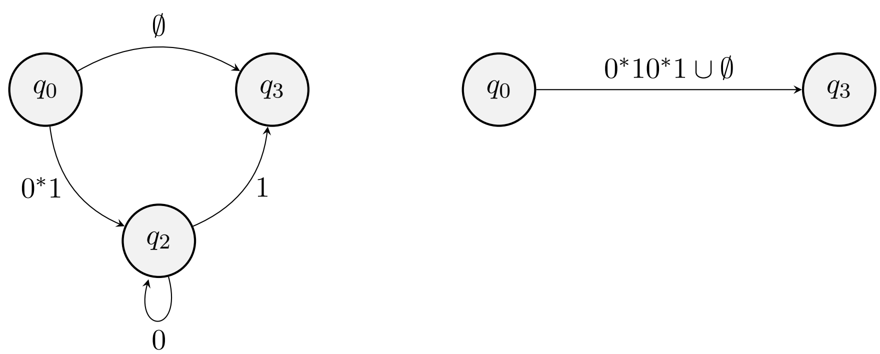
   (a)



(b) Rip out q_1:



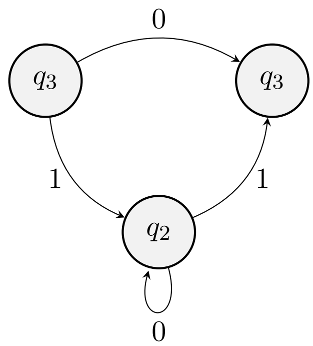
(c)



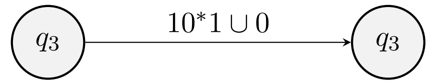
before after

So the transition from q_0 to q_3 is labeled with $0^*10^*1 \cup \emptyset$, which simplifies to 0^*10^*1.

(d)



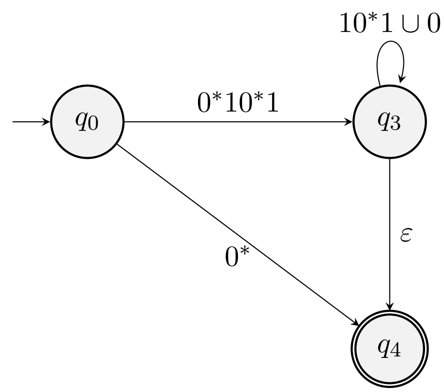
before



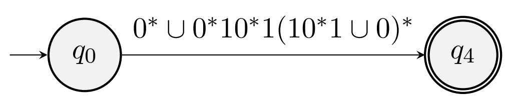
after

So the transition from q_3 to q_3 is labeled with $10^*1 \cup 0$.

(e) Rip out q_2:



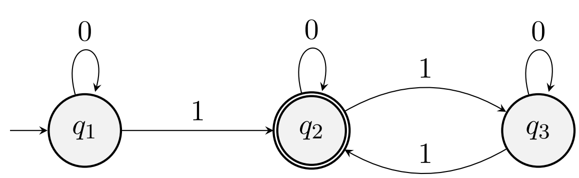
(f) Rip out q_3:



(g) The resulting regular expression is $0^* \cup 0^*10^*1(10^*1 \cup 0)^*$.

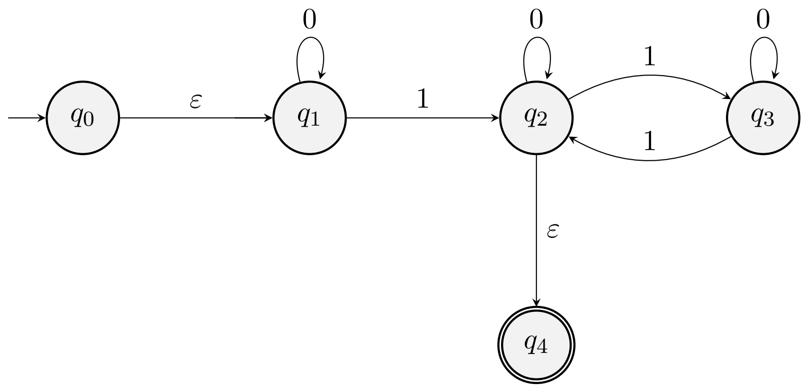
(h) The language $L(A)$ is all words having an even number of 1s.

3. Here is the DFA A^c:

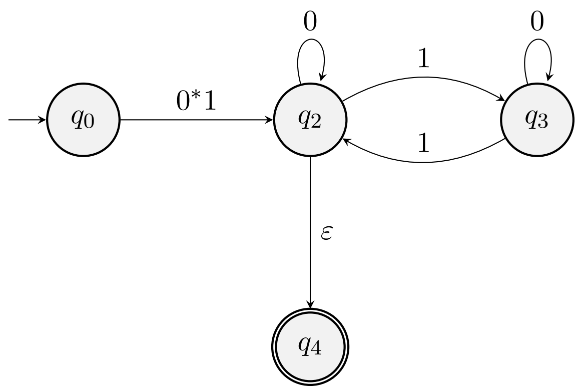


(a) All words having an odd number of 1s.

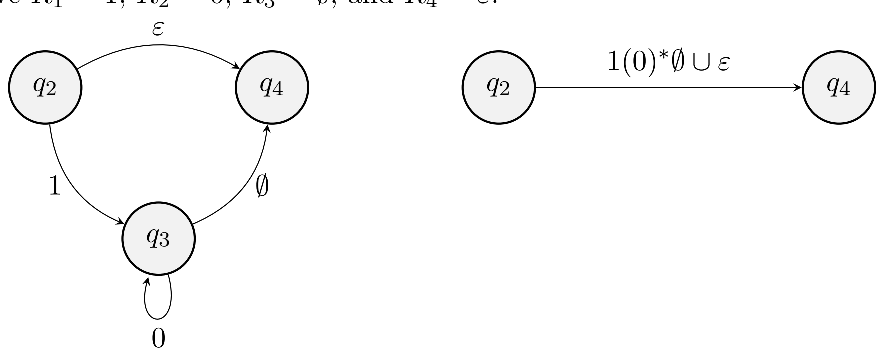
(b) Modification of A^c:



(c) Rip out q_1:



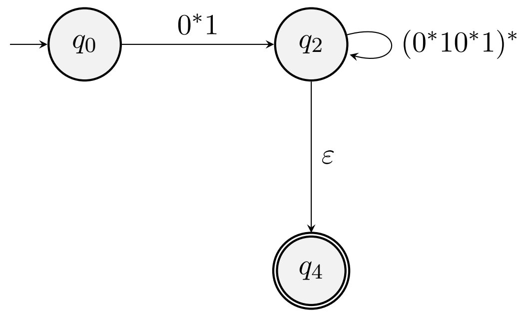
(d) We have $R_1 = 1$, $R_2 = 0$, $R_3 = \emptyset$, and $R_4 = \varepsilon$:



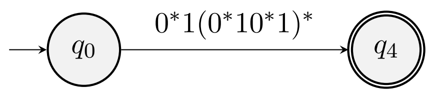
before after

The regular expression for the transition from q_2 to q_4 is $1(0)^*\emptyset \cup \varepsilon$, which simplifies to ε.

(e) Rip out q_3:



(f) Rip out q_2:



(g) The resulting regular expression is $0^*1(0^*10^*1)^*$.

4. See your instructor.