Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
M. Ahmadi, P. Bamberger, H. Ghodselahi, Y. Maus, P. Schneider

Theoretical Computer Science - Bridging Course
Summer Term 2017
Sample Solution Sheet 2

Exercise 1 (8 points)

Construct DFAs that recognize the following languages. Drawing the state diagrams is sufficient. The alphabet is $\Sigma = \{0, 1\}$.

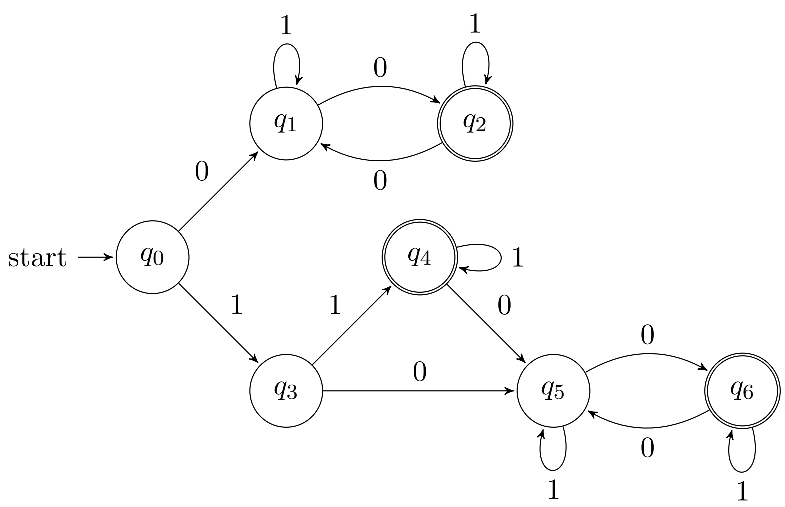
(a) (3 points) $L_1 = \{w \mid |w| \geq 2 \text{ and } w \text{ contains an even number of zeros}\}$.

(b) (2 points) $L_2 = \{w \mid w \text{ contains exactly two ones}\}$.

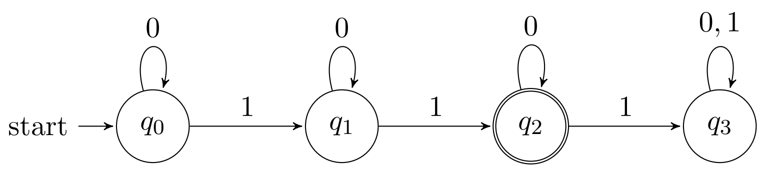
(c) (3 points) $L_3 = \{w \mid w \text{ has an odd number of zeros and ends with } 1\}$.

Solution

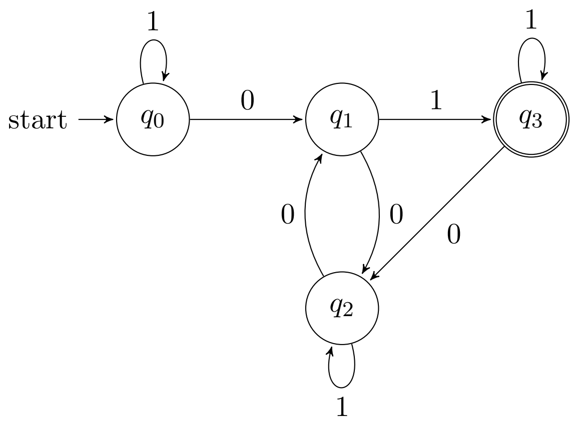
(a)



(b)



(c)



Exercise 2 (2+3 points)

Let L, L_1, L_2 be regular languages. Show that both $\overline{L} := \Sigma^* \setminus L$ and $L_1 \cap L_2$ are regular as well by constructing the corresponding DFAs.
Remark: No need for drawing state diagrams. Show how a DFA for the language in question can be constructed presuming the existence of DFA for L, L_1, L_2.

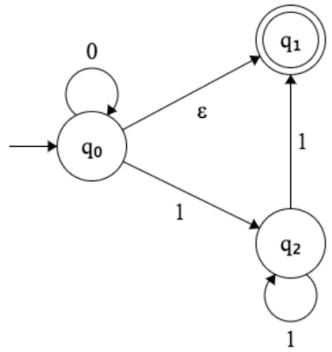
Solution

Let $M = (Q, \Sigma, \delta, q_0, F)$ be the DFA recognizing L. We define the DFA $\overline{M} := (Q, \Sigma, \delta, q_0, \overline{F})$ by inverting the set of accepting states of M, i.e. $\overline{F} := Q \setminus F$. We show that $\overline{M}$ recognizes $\overline{L}$.
If $w \in \overline{L}$, then $w \notin L$ and so M halts in an non accepting state q when processing w. $\overline{M}$ will halt in the same state (because we only changed the set of accepting states), but here q is an accepting state. Analogously, if $w \notin \overline{L}$, then $w \in L$ and so M halts in an accepting state when processing w. $\overline{M}$ will again halt in the same state, but here q is a non accepting state. So we have that $\overline{M}$ halts in an accepting state when processing w if and only if $w \in \overline{L}$. Thus $\overline{M}$ recognizes the language $\overline{L}$ which is therefore regular.

For proving the regularity of $L_1 \cap L_2$, we construct the product automaton like done in the lecture (Theorem 1.25. p. 30) for $L_1 \cup L_2$, with the difference that we set $F := F_1 \times F_2$ as the set of accepting states, where F_1 and F_2 are the sets of accepting states of the DFAs for L_1 and L_2.

Exercise 3 (7 points)

Consider the following NFA.



(a) (2 points) Give a formal description of the NFA by giving the alphabet, state set, transition function, start state and the set of accept states.

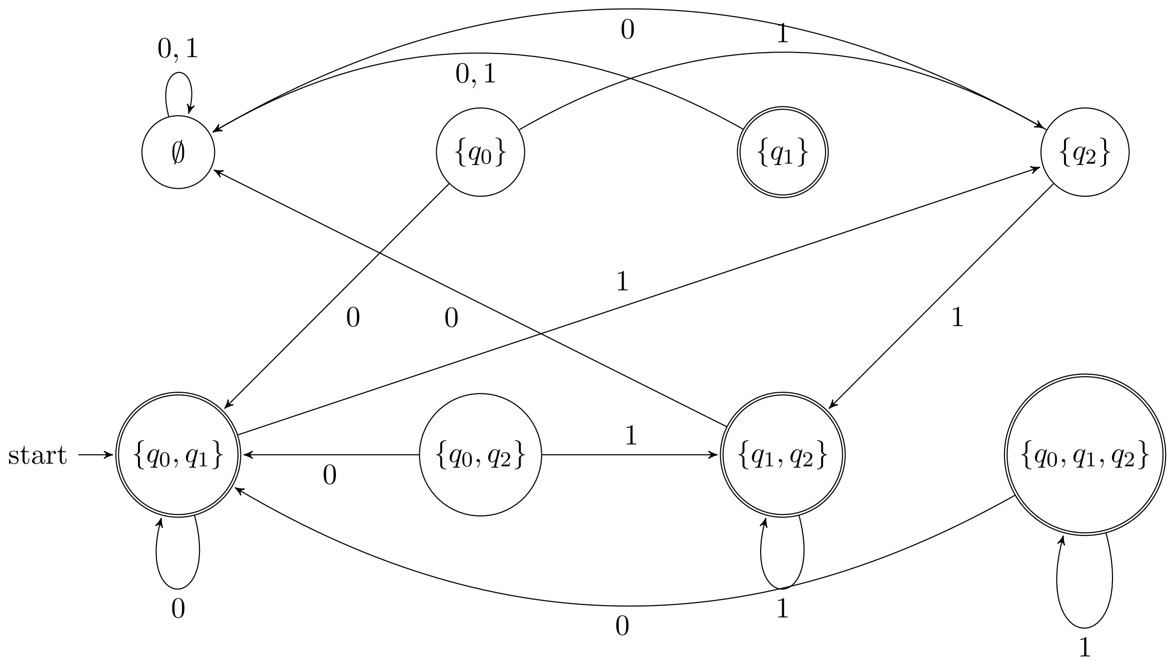
(b) (5 points) Construct a DFA which is equivalent to the above NFA by drawing the corresponding state diagram.

Solution

(a) The set of states is $Q = \{q_0, q_1, q_2\}$; the alphabet $\Sigma = \{0, 1\}$; the initial state is q_0; the set of accept states is $F = \{q_1\}$; the transition function is shown in the following table.

	q_0	q_1	q_2
0	q_0	$\emptyset$	$\emptyset$
1	q_2	$\emptyset$	q_1, q_2
ϵ	q_1	$\emptyset$	$\emptyset$

(b)



If we leave out nodes with no path leading into it, he have