Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
P. Bamberger, Y. Maus,

Theoretical Computer Science - Bridging Course
Summer Term 2017
Exercise Sheet 2

Hand in (electronically or hard copy) by 12:15 pm, Monday, November 06th, 2017

Exercise 1: Constructing DFAs *(2+2+2+2 Points)*

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

(a) $L_1 = \{w \mid |w| \geq 2 \text{ and } w \text{ contains an odd number of ones}\}$.

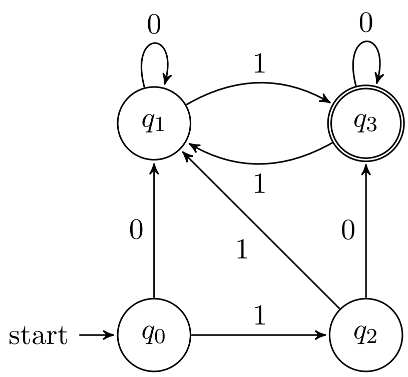
(b) $L_2 = \{w \mid w \text{ contains an even number of zeros}\}$.

(c) $L_3 = \{\text{in } w \mid w \text{ every zero is immediately followed by a one}\}$.

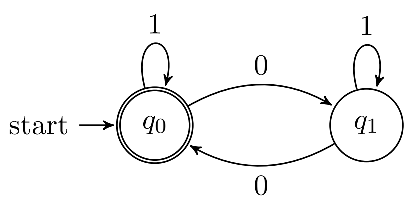
(d) $L_4 = \{w \mid w \text{ ends with } 01\}$.

Sample Solution

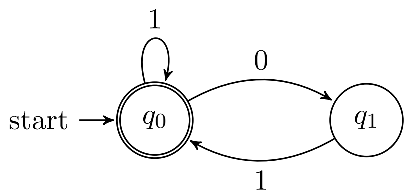
1.



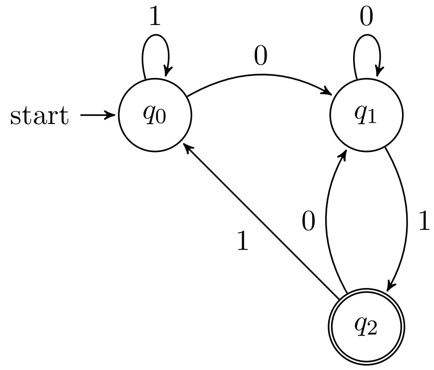
2.



3.



4.



Exercise 2: Kleene Star *(4 Points)*

For two languages L and K we define

$$L \cdot K := \{ww' \mid w \in L, w' \in K\}.$$

Use counter examples to show that none of the following equalities is valid for general languages K and L.

1. $K \cdot L = L \cdot K$
2. $(K \cdot K)^* = K^*$
3. $K^* \cdot K = K^*$
4. $(LK)^* = (L \cup K)^*$

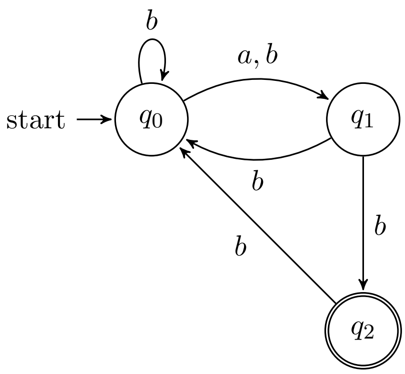
Sample Solution

1. $K = \{1\}$ and $L = \{0\}$, then $\{10\} = K \cdot L \neq L \cdot K = \{01\}$.
2. Let $K = \{1\}$, then $\{1^n \mid n \text{ is even}\} = (K \cdot K)^* \neq K^* = \{1^n \mid n \in NN\}$.
3. Let $K = \{1\}$. Then $\epsilon \in K^*$ but $\epsilon \notin K^* \cdot K$
4. Let $K = \{1\}$ and $L = \{0\}$, then all word in $(LK)^*$ have even length but the words in $(L \cup K)^*$ do not have to have even length, e.g., $1 \in (L \cup K)^*$ but $1 \notin (LK)^*$.

There are many examples in which the languages L and K contain more than one word!

Exercise 3: From NFA to DFA *(1+2+2 Points)*

Consider the following NFA.



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

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

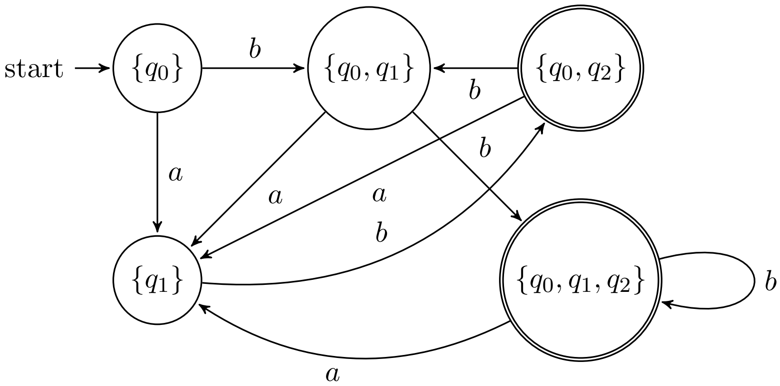
(c) Explain which language the automaton accepts.

Sample Solution

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

	q_0	q_1	q_2
a	$\{q_1\}$	$\emptyset$	$\emptyset$
b	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0\}$

(b) After performing the algorithm from the lecture we obtain the following DFA. All transitions which are not in the picture go to the garbage state $\emptyset$.



(c) The recognized language contains words of length at least two. Furthermore any a is immediately followed by a b. The number of b's after the last a must not be two.

Exercise 4: Union of regular Languages *(3 Points)*

Let L_1, L_2 be regular languages. Show that $L_1 \cup L_2$ is also regular without (explicitly!) using the proof which was presented in the lecture.

Sample Solution

As L_1 and L_2 are regular there are DFAs A_1 and A_2 with $L(A_1) = L_1$ and $L(A_2) = L_2$. Let q_1 and q_2 be the starting states of A_1 and A_2 respectively.
Now, we construct an ϵ-NFA A which consists of an additional start state q and the two DFAs A_1 and A_2. Besides all transistions from A_1 and A_2 we introduce two ϵ transitions: One from q to q_1 and one from q to q_2. The set of accepting states is the union of the accepting states of A_1 and A_2.
The constructed ϵ-NFA accepts a word if and only if it is accepted by A_1 or A_2, i.e., $L(A) = L(A_1) \cup L(A_2)$.
The algorithm of the lecture can be used to transfer A into a DFA; hence we have a DFA which accepts the language $L(A_1) \cup L(A_2)$, i.e., the language is regular.