Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
Philipp Bamberger, Yannic Maus

Theoretical Computer Science - Bridging Course
Summer Term 2018
Exercise Sheet 2

for getting feedback submit (electronically) before the start of the tutorial on 5th of November 2018.

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

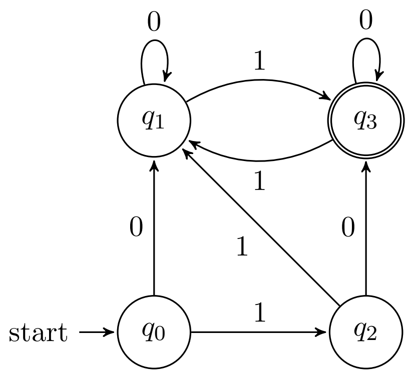
(b) $L_2 = \{w \mid w$ contains an even number of zeros$\}$.

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

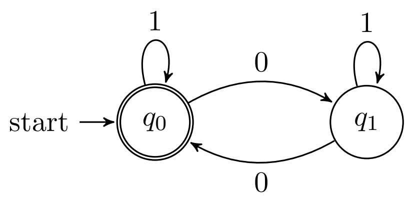
(d) $L_4 = \{w \mid w$ ends with 01$\}$.

Sample Solution

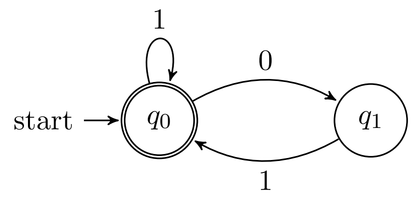
1.



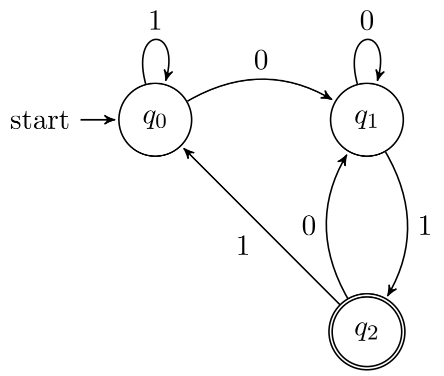
2.



3.



4.



Exercise 2: Maxstring

Let

$$maxstring(L) = \{w : w \in L \text{ and for all words } z \in \Sigma^* : z \neq \epsilon \Rightarrow wz \notin L\} \ .$$

(a) What is $maxstring(L_1L_2)$, where $L_1 = \{w \in \{a,b\}^* : w \text{ contains exactly one } a\}$ and $L_2 = \{a\}$?

(b) Explain how to prove that the regular languages are closed under *maxstring*.

Hint: Let L be a regular language. You need to prove that $maxstring(L)$ is regular as well.

Sample Solution

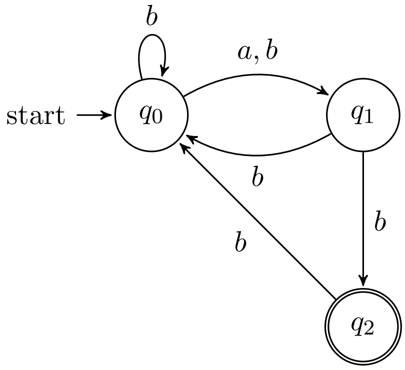
(a) L_1L_2. Let $w \in L_1L_2$ then it can be written as $w = w_1w_2$ with $w_1 \in L_1$, $w_2 \in L_2$, w_1 contains exactly one a and $w_2 = a$. Assume there is some $z \neq \epsilon$ with $w_1w_2z = w_1az \in L_1L_2$. As all words in L_1L_2 end with an a this implies $z = a$ and $w_1a \in L_1$. However, as w_1 contains exactly one a the word w_1a contains two a and is not contained in L_1, a contradiction.

The inclusion $maxstring(L_1L_2) \subseteq L_1L_2$ is by definition.

(b) Take a DFA A for the language L. Construct an DFA B for the language $maxstring(L)$ by taking A with a modified set of accepting states: Let the set of accepting states of B be the accepting states of A that do not have a path to an accepting state.

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.

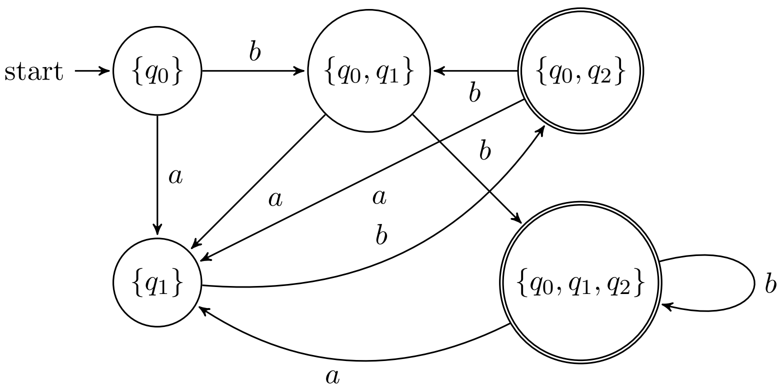
(c) Explain which language the automaton accepts.

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.