CSc 30400 Introduction to Theory of Computer Science

1st Homework Set

Due Date: 2/23

Instructions:

- This homework set covers Sections 1.1 and 1.2. (DFAs and NFAs)
- Submit your solutions by email or hand them in class **before the beginning of the class!** If your electronic solutions consist of more than one files, compress all your files into one file and send only the zipped file.
- Preparing your solutions using JFLAP earns a 5% credit, thus you are very encouraged to use it. However be careful: There are some questions where JFLAP can automatically prepare the answer instead of you. I have no way to distinguish between an automatically generated answer and a self prepared one, so any correct answer gets full points. However I strongly recommend that you don't use JFLAP's automatic feature. In the exam you are on your own and you have to be familiar with this type of questions. If you need help you can use JFLAP's help steps.
- Exercises are divided into three sections: Practice, Easy and Hard. Practice questions just help you understand the material. Most of the questions are in the section "easy questions" (though not all easy questions have the same difficulty). Hard questions are a bit tougher than the easy ones. All the questions earn the same amount of points unless otherwise stated.

1 Practice questions

P 1. The formal definition of a DFA M is $(\{q_1, q_2, q_3, q_4, q_5\}$, {u,d}, δ, q_3, $\{q_3\})$, where δ is given by table 1. Give the state diagram of this machine.

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

Table 1: The transition function δ of exercise 1.

P 2. Figure 1 presents the state diagrams of a NFA M_1 and a NFA$_\varepsilon$ M_2. Answer the following questions about each of these machines.

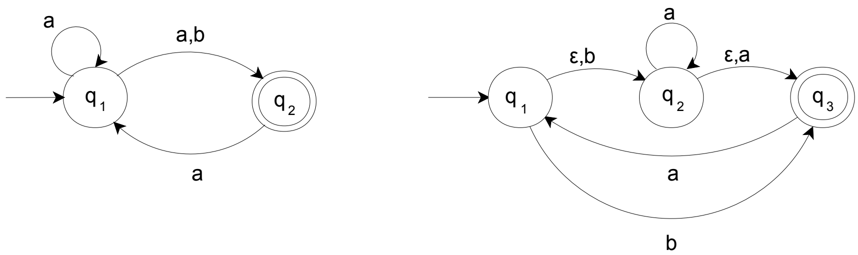

Figure 1: M_1 and M_2 of exercise 2.

(a) What is the start state?

(b) What is the set of accepting states?

(c) What sequence of sets of states does the machine go through on input aaab?

(d) Does the machine accept the string aaab?

(e) Does the machine accept the string ε?

2 Easy questions

E 1. Design DFA on the alphabet $\Sigma = \{0, 1\}$ that recognizes the language:

$L_1 = \{000,\ 010,\ 100,\ 110\}$

$$
\begin{aligned}
L_2 &= \{w \in \Sigma^* : w \text{ starts with } 010\} \\
L_3 &= \{w \in \Sigma^* : w \text{ contains the substring } 011\} \\
L_4 &= \{w \in \Sigma^* : w \text{ ends with } 111\} \\
L_5 &= \{w \in \Sigma^* : w \text{ starts with 1 and contains } 10\} \\
L_6 &= \{w \in \Sigma^* : w \text{ contains exactly two 1s}\} \\
L_7 &= \{w \in \Sigma^* : w \text{ contains at least two 1s}\} \\
L_8 &= \{w \in \Sigma^* : w \text{ contains two consecutive 1s}\} \\
L_9 &= \{w \in \Sigma^* : w \text{ contains an even number of 1s and ends with } 0\} \\
L_{10} &= \{w \in \Sigma^* : w \text{ contains a number of 1s congruent to } 1 \mod 3\}
\end{aligned}
$$

E 2. Design NFA$_\varepsilon$ on the alphabet $\Sigma = \{0, 1\}$ that recognizes the language:

$L_1 = \{w \in \Sigma^* : w \text{ starts with 1 and ends with } 10\}$

$L_2 = \{w \in \Sigma^* : |w| \text{ is even but not a multiple of three}\}$

$L_3 = \{w \in \Sigma^* : w$ contains a pair of 1s that are separated by an odd number of symbols$\}$

$L_4 = \{w \in \Sigma^* : w$ contains an even number of 1s, an odd number of 0s and do not contain the substring $01\}$

$L_5 = \{w \in \Sigma^* : w \text{ contains exactly two 1s and exactly three 0s}\}$

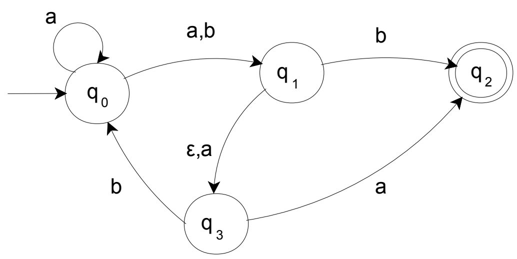

Figure 2: The NFA$_\varepsilon$ of exercise 3.

E 3. For the NFA$_\varepsilon$ of figure 2 first construct an equivalent NFA, from this an equivalent DFA (use the slides about NFA$_\varepsilon$ - NFA - DFA equivalence). Now construct the minimum equivalent DFA (use the slides about minimizing DFAs).

E 4. You decide to protect your property with a sophisticated alarm system. In order to enter, someone has to punch in a secret 4-digit number. For simplicity we will assume that the only allowed digits are $\{1, 2, 3, 4, 5\}$ and that the secret number is 3143.

Your goal for this problem is to design a DFA that works in the same way as the alarm system (where accepting an input is equivalent to opening the door). Specifically:

- The input, in the alphabet $\{1, 2, 3, 4, 5\}$, tells you the digits that were pressed in the order that this happened.
- Obviously, your DFA must accept the string 3143, even if more buttons are pressed afterwards.
- If a wrong digit is entered, the alarm system should restart, i.e. go back to expecting 3143.
- If the system restarts three times, the door should lock permanently.

For example:

- 3143234 should be accepted.
- 423143, 343143, 31423153143 should be accepted.
- 313143 should not be accepted (the second 3 causes a restart, so it's not considered part of the secret) but 3133143 should be accepted.
- 2223143 should not be accepted (three restarts happen).

3 Hard questions

H 1. Problem 1.31 of book: For any string $w = w_1 w_2 \cdots w_n$ the **reverse** of w, written $w^{\mathcal{R}}$, is the string w in reverse order, $w_n \cdots w_2 w_1$. For any language L, let $L^{\mathcal{R}} = \{w^{\mathcal{R}} | w \in L\}$. Show that if L is regular, so is L^R. (Hint: take the DFA that decides L and make it work "backwards")

H 2. Problem 1.38 of book: An **all**-NFA M is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ that accepts $x \in \Sigma^*$ if *every* possible state that M could be in after reading input x is a state from F. Note, in contrast, that an ordinary NFA accepts a string if *some* state among these possible states is an accept state. Prove that all-NFAs are equivalent with DFAs. Hint: First observe that a DFA is an all-NFA (why?). For the opposite direction, from each all-NFA you should construct an equivalent DFA.