

# Discrete Event Systems

## Exercise Sheet 1

### 1 Finite Automata

In this exercise you are asked to design your first finite automata. Try to minimize the number of states of your machines.

- a) Consider the alphabet  $\{0,1\}$ . Implement an automaton which accepts the following strings:  
At first there are zero or more '1's, followed by zero or more '0's. This in turn is followed by an arbitrary (non-zero) number of '1's.
- b) Design an automaton which decides whether a number is divisible by three. Assume that the digits of the number are inserted sequentially, that is, the number '135' is inserted as '1', '3' and finally '5'. How many states do you need? (Hint: Cross sum!)

### 2 “Mais im Bundeshuus”

It is Wednesday morning and the seven members of the Swiss Federal Council meet to decide about an important topic: In order to decrease the expenses of education, should only women be allowed to study at ETH?

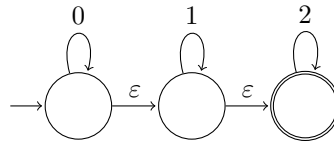
- a) Assume that the seven members vote one after another. Further, assume that there is no abstention of voting. Design an automaton which accepts the ballot if and only if the majority of the members voted in favor of the proposition.
- b) Extend your automaton for the case of abstentions of voting. The automaton should accept if and only if more members voted in favor of the proposition than against it.

### 3 Nondeterministic Finite Automata

- a) Consider the alphabet  $\{a, b\}$ . Construct an NFA that accepts all strings containing the substring  $abba$  at least twice. (This means that words containing  $abbaabba$  as a substring should also be accepted!)
- b) Construct an NFA which accepts the following regular expression:  $(00 \cup (0(0 \cup 1)^*))^*$ .
- c) Construct an NFA accepting  $1^*0^*1^+$  with as few states as possible. (cf. Exercise 1.1.a)
- d) Consider a machine  $M := (Q, \Sigma, \delta, q_0, Q)$ . Is it possible to make a statement about the strings being accepted by  $M$ ? Does it make a difference whether  $M$  is deterministic or not?

## 4 De-randomization

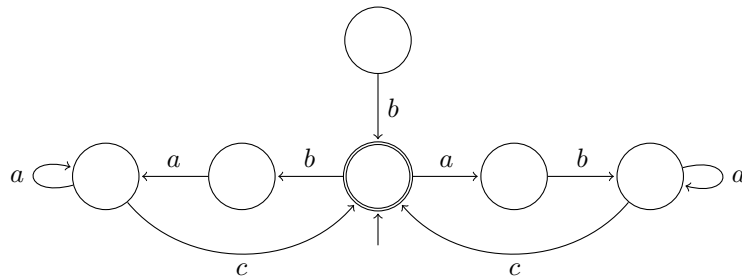
- a) Give a regular expression for the following NFA and construct an equivalent NFA *without*  $\epsilon$ -transitions.



- b) Finally, transform the machine into a deterministic automaton.

## 5 States Minimization

Simplify the following automaton. Explain why your changes are allowed. Finally, give the corresponding regular expression.



## 6 “Regular” Operations in UNIX

In this exercise you are asked to provide a UNIX command to output all lines in a file ending with “password” or “passwort”, followed by an unknown number (potentially zero) of vowels.

**Turn page for general information on the course and the exercises!**

## General Information

Please find below the most important information about the course and the exercises.

### Time and Place

- Lecture by Prof. L. Vanbever, Prof. R. Wattenhofer and Prof. L. Thiele, Thursday, 13:15–15:00, ETZ E6.
- Exercises after the course, *i.e.* Thursday, 15:15–16:45, ETZ E6.

### Exam Requirements

There are *no requirements* for attending the final exam from our side. However, if you need a Testat due to special circumstances, you need to hand in solutions to at least 80% of the exercises.

### Exercise Proceedings

Before every lecture, we will publish a new exercise sheet on the course website<sup>1</sup>. This exercise sheet is intended to be solved *during the exercise session* on Thursday where tutor(s) will be available to assist you and to answer potential questions. After every exercise session we will provide a detailed sample solution for the respective exercise sheet on the course website.

You can hand in your solutions for correction after the exercise session on a *voluntary basis*. This is not mandatory since a *Testat is not required* to be admitted to the final exam.

### Lecture Material

The lecture slides will also be made available on the course website in due time.

---

<sup>1</sup><http://dcg.ethz.ch/lectures/des/>