# Exercises about finite automata

1. Let's have alphabet $\Sigma = \{a, b\}$. Construct a finite automaton, which recognizes the following language:

   a) $L(M) = L(\emptyset)$
   b) $L(M) = L(\emptyset^*)$
   c) $L(M) = L(\epsilon)$
   d) $L(M) = L(\epsilon^*)$
   e) $L(M) = L(\Sigma^*)$

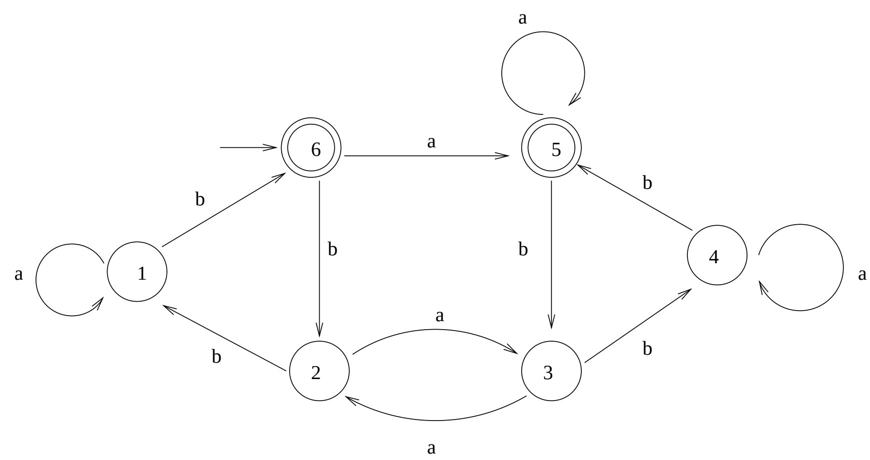
2. Create a finite automaton that describes the function of a lift moving between two floors. The lift may be either up or down. On both floors, there is a simple "come here" button, and in the lift there is an "up" and a "down" button. In addition, the lift has a door that can be opened and closed; the lift only moves if the door is closed. (The automaton does not have to have specific favorable states.)

   Input: pushing the buttons, opening and closing the door. States: up/down; door open/closed. Notation: U = up, D = down, o = door open, c = door closed.

3. Let $\Sigma = \{0, 1\}$. Construct a deterministic finite automaton $M$, which recognizes the following language

   a) $L(M) = \{w | \text{the length of } w \text{ is odd}\}$
   b) $L(M) = \{w | \text{number of 1s in } w \text{ is multiple of three}\}$
   c) $L(M) = \{w | w \text{ has even number of 1s and 0s}\}$

4. Create a minimum automaton that is equivalent to the following deterministic finite automaton.




5. Let's consider the comic in the lecture material about a nondeterministic automaton consulting the doctor. Which middle phases does the operation consist of? Does everything go as it should?

6. Invent at least one phenomenon in your everyday life, which can be described as a nondeterministic automaton. Draw the transition diagram of your automaton and try to determinize it!

7. Create a non-deterministic finite automat that tests whether alphabet $\{a, b\}$ contains the character string "abaa" and make it deterministic.

8. Give nondeterministic finite automata to accept the following languages. Try to take advantage of nondeterminism as much as possible.

   a) The set of strings over alphabet $\{0, 1, ..., 9\}$ such that the final digit has appeared before.
   b) The set of strings over alphabet $\{0, 1, ..., 9\}$ such that the final digit has *not* appeared before.
   c) The set of strings over alphabet $\{0, 1\}$ such that there are two 0's separated by a number of 1's that is a multiple of 4. Note that 0 is an allowable multiple of 4.

9. What can you say about time requirements of problems, which belong to the class of regular languages? What about their space requirements? (Hint:

Consider the program representation of a finite automaton.) Is there any difference if the corresponding automaton is deterministic or undeterministic?

10. Generating a regular language: "The Poem Automat" **2 points**

    Implement a program that creates rows according to the regular expression $(the)(\text{ATTR})^*$ SUBJ PRED the $(\text{ATTR})^*$ OBJ( ATTRIB$\cup\epsilon$) Languages ATTR, SUBJ, OBJ, PRED and ATTRIB consist of the following words, for example:

    ATTR = {fat, black}
    SUBJ = {cat, moon, fish}
    OBJ = {cat, moon, fish}
    PRED = {shines, watches, swims}
    ATTRIB = {in the sky, in the lake}

    Your program can now produce 'verses' like the following:

    The black cat watches the fat fish in the lake
    The moon swims in the sky

    NB! Specify that not all predicates may be followed by an object (e.g. swims).

    Expand the example as you wish with words that fit into the poem! Add the exclamations

    "What a (ATTR) (SUBJ)!"

    and the questions

    "Are you a (ATTR) (SUBJ)?"

    Hand in the code of your program and examples of your outputs (poems) to the instructor! Be ready to represent your program/poems in the art exhibition!