\* 1. There are two languages *L*1 and *L*2 over the alphabet {0, 1}. Words of *L*1 are described by the regular expression 0\*1\*0\*1\*0\*, words of *L*2 are described by the regular expression (01+10)\*. Construct a finite automaton which

a) accepts language *L*1 ∪ *L*2, b) accepts language *L*1 ∩ *L*2.

\* 2. The automaton A1 resp. A2 accepts the language *L*1 resp. *L*2 over alphabet A, |A| = *k*. Each of A1 and A2 has *n* internal states. We have to determine whether *L*1 ∩ *L*2 is finite. What is the asymptotic complexity of the task?

\* 3. The alphabet is {*a*, *b*, *c*, *d*}. A critical string over this alphabet is a such one which begins and ends with the symbol *b* and moreover its Hamming distance from the pattern *abbbcdabbcdab* is greater than 2. Construct a finite automaton which will detect in a text the occurences of all critical strings.

\* 4. There is a finite automaton A which can detect in a text all its substrings which Levenshtein distance form a given pattern *p* is less than a given value *k*. Automaton A contains some epsilon-transitions. Construct an example of this automaton for |*p*| = 6 and *k* = 3. Remove all epsilon-transitions and write down the resulting automaton.

\* 5. We define reduced Levenshtein distance of the words *v*, *w* (over alphabet A) to be the minimum number of edit operations which will transform word *v* into word *w*. In this case, we consider as edit operations only the operations Insert and Delete. Construct a NFA without epsilon-transitions which will be able to detect in a text any string which reduced Levenshtein distance form the pattern *abaabacc* is exactly 2.

6. Let us denote by symbol d(*x*, *y*) the Levenshtein distance between words *x* and *y*. There are three words *u*, *v*, *w* and it holds that d(*u*, *v*) = *d*1, d(*v*, *w*) = *d*2. What are possible values of d(*u*, *w*) in relation to *d*1 and *d*2? The alphabet is the same for all words involved.

7. Construct a NFA which will detect in a text any element of the set of all continuous substrings of the pattern

*abcdefghijklmnopqrstuvwxzy*.

8. Let us denote by the symbol HD(*v*, *w*) resp. LD(*v*, *w*) the Hamming resp. Levenshtein distance between the words *v* and *w* over the alphabet A. Decide which of the following may be true for some words *v* and *w* which length is at least 5.

a) HD(*v*, *w*) < LD(*v*, *w*), b) HD(*v*, *w*) = LD(*v*, *w*), c) HD(*v*, *w*) > LD(*v*, *w*).

9. Write down all words over alphabet {*a*, *b*, *c*} which Levenshtein distance form the word *aba* is exactly a) 1, b) 2.

10. There is a pattern *p* and a string *q*. The string *q* was obtained from *q* by applying exactly one of the two operations:

SWAP (= swapping of two immediately neigbouring symbols)

REWRITE (= substitution of a single symbol by another symbol of the alphabet)

Construct a NFA which will detect in a text any occurence of *q* when *p* = *abbaac* and alphabet is {*a*, *b*, *c*}.

11. Alphabet A = {0, 1}. The language *L*1 contains all words over A which contain the continuous substring 00 exactly once. The language *L*2 contains all words over A which contain the continuous substring 11 at least once. Construct a finite automaton which will detect in a text over A all occurences of all words of the language

a) *L*1 ∩ *L*2, b) *L*1 ∪ *L*2.

12. There are two finite sets M1 and M2 of words over alphabet A. Language L consists of all words *w* over A for which holds that at least one prefix of w is in set M1 and at least one suffix of w is in M2. The whole word is considered to be its own prefix and also its own suffix. Describe an algorithm which will construct an automaton accepting L. Describe a concrete example for |M1| = |M2| = 2.

13. An alphabet A, a pattern *p* over A and a fixed positive integer *k* are given. Describe an algorithm which will print out all words over A which Hamming distance form *p* is exactly *k*. What is the asymptotic complexity of this algorithm?

14. An alphabet A, a pattern *p* over A and a fixed positive integer *k* are given. Describe an algorithm which will print out all words over A which Levenshtein distance form *p* is at most *k*. What is the asymptotic complexity of this algorithm?