**ASSIGNMENT FOR THE COURSE**
**THEORETICAL COMPUTER SCIENCE**
**THEORY OF ALGORITHMS**

(1) Prove that $\{ab, aba\}^* = \{\varepsilon\} \cup \{a\}\{ba, baa\}^*\{b, ba\}$ (i.e. prove 2 inclusions: "$\subseteq$" and "$\supseteq$").

(2) Build a DFA which recognizes the language $\{0, 1\}^*$ in alphabet $\mathcal{A} = \{0, 1, 2\}$.

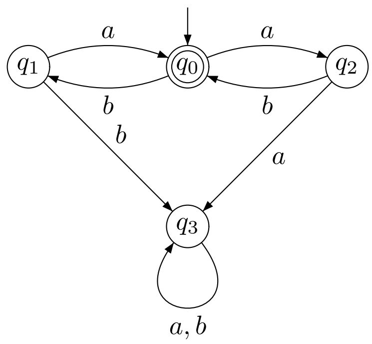
(3) Build a DFA which recognizes

$$L = \{w | w \in \{0, 1\}^* \text{ and the difference between the number of 0's and 1's in } w \text{ is even}$$

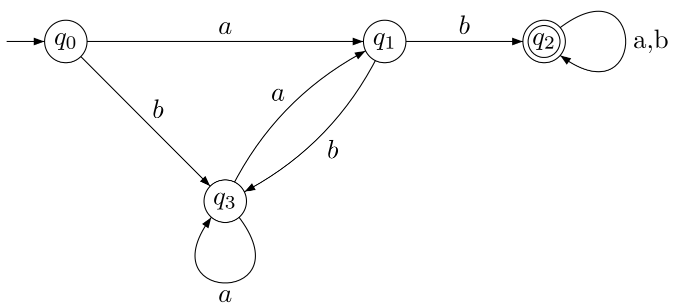
in $\mathcal{A} = \{0, 1\}$.

(4) Build a DFA for $L = \{w|$ in $w$ before $a$ always comes $b\}$ in $\mathcal{A} = \{a, b\}$.

(5) Describe the language recognized by the following DFA:




(6) Describe the language recognized by the following NFA:




(7) Using the algorithm considered in the lecture which allows one to build a DFA from a NFA, build a DFA for the language from Exercise (6).

(8) Construct a regular grammar generating $L = \{w | w \neq \varepsilon$ and the number of 1's divides by 3$\}$.

(9) Prove that every finite language not containing $\varepsilon$ is generated by a regular grammar.

(10) Prove that $L = \{w \in \{a, b\}^+ | w$ has equal number of occurrences of $a$ and $b\}$ is not regular.

(11) Prove that $L = \{ww^R | w \in \{a, b\}^+\}$ is not regular.

(12) Build a PDA recognizing $L = \{a^n b^n | n \in \omega\}$.

(13) Describe the language generated by the following CFG $\Gamma = \langle V, T, P, S\rangle$, where $V = \{S, A, B\}$, $T = \{a, b\}$, $P = \{S \to aB, S \to bA, A \to a, A \to aS, A \to BAA, B \to b, B \to bS, B \to ABB\}$.

(14) Construct a CFG which generates

$$L = \{xc^n | n \in \omega, x \in \{a,b\}^* \text{ and the number of } a\text{'s is } n \text{ or the number of } b\text{'s is } n\}$$

(15) Provide an example of context-free languages $L_1, L_2$ such that $L_1 \setminus L_2$ is not context-free.

(16) Prove that $L = \{ww | w \in \{0,1\}^*\}$ is not context-free.

(17) Prove that $L = \{a^i b^j c^k | i < j < k\}$ is not context-free.

(18) Prove that $f(x) = x!$ (where $0! = 1$) is primitive recursive.

(19) Prove that the function $rest(x, y)$ which outputs the reminder of division $x$ by $y$ (where $rest(x, 0) = x$) is primitive recursive.

(20) Prove that $f(x) = [\sqrt{x}]$ is primitive recursive.

(21) Describe the function which is the result of application of the minimization operator to $g(x, y, z) = |zy - x|$.

(22) Prove that the function $f$ such that $f(n)$ is the $n$-th Fibonacci number, is primitive recursive.

(23) Construct a Turing machine which performs transposition: $01^x q_1 01^y 0 \Rightarrow 01^y q_0 01^x 0$ *without building new cells to the left and to the right end of the tape.*

(24) Construct a Turing machine which performs duplication: $q_1 01^x 0 \Rightarrow q_0 01^x 01^x 0$.

(25) Construct a Turing machine which correctly computes the function $f(x, y) = x - y$ (in particular, if $x < y$ the machine does not halt).

A set $A$ is computable if its characteristic function $\chi_A$ is computable (where $\chi_A(x) = 1$ if $x \in A, \chi_A = 0$ if $x \notin A$).

(26) Prove that if $f(x)$ is computable then its graph $\Gamma_f$ is a computable set.

(27) Prove that if $f(x)$ is a monotone computable function then its range is a computable set.

(28) Prove that if $f(x, y)$ is computable and $g(x) = \mu y[f(x, y) = 0]$ then the graph of $g$ $\Gamma_g$ is computable.

(29) Prove that the set $A = \{x \in \omega | x \text{ is a perfect number}\}$ is computable (a number $x$ is perfect if the sum of all its devisors that are less than $x$ equals $x$).

(30) Prove that it is impossible to get the functions $s(x) = x + 1$ and $f(x) = 2x$ from the basic functions $o$ and $I^n_m$ using the operators of composition and primitive recursion.

(31) Let $e = a_0, a_1 a_2 a_2 \ldots$ be a representation of the number $e$ (Euler's number $2,718281\ldots$) as an infinite decimal fraction. Prove that $f(n) = a_n$ is computable.

(32) Prove that the problem 2-SAT is in $\mathcal{P}$.

(33) Let $L_1 \in \mathcal{P}$, $L_2$ be $\mathcal{NP}$-complete and $L_3 \notin \mathcal{NP}$. Characterize the languages $L_1 \cap L_2$, $L_1 \cup L_2$, $L_2 c L_3$ (where $c$ is a symbol not in $L_2$ or $L_3$), $\overline{L_3}$ using the following expressions:
   (a) definitely in $\mathcal{P}$;
   (b) definitely in $\mathcal{NP}$ (but perhaps not in $\mathcal{P}$ and perhaps not $\mathcal{NP}$-complete);
   (c) definitely $NP$-complete;
   (d) definitely not in $\mathcal{NP}$.

Explain your choice.

(34) Decide whether $\mathcal{P}$ is closed under union, intersection, concatenation and star. Give a proof of your choice.

(35) Decide whether $\mathcal{NP}$ is closed under union, intersection, concatenation and star. Give a proof of your choice.

(36) Build a nondeterministic Turing machine which recognizes $L = \{ww | w \in \{a, b\}^*\}$.