Automata, Computability, and Formal Languages
Fall 2022, Test 3

1-2. Let L be language of all the words that contain equally many digits 0, 1, 2, and 3:

$$L = \{\Lambda, 0123, 3012, \ldots, 00112233, 01302132, \ldots\}.$$

Prove that this language is not context-free.

3. The following Turing machine takes words consisting of a's and b's, and replaces small a's with capital A's:

- start, – → work, R (here, – means blank)
- work, a → A, R
- work, b → R
- work, – → back, L
- back, A → L
- back, b → L
- back, – → halt

Trace it on the example of the word ab. Explain how each step will be represented if we interpret the Turing machine as a finite automaton with two stacks.

4. Arithmetic operations on Turing machines:

a Design a Turing machine that subtracts 2 from a unary number. Assume that the original number is greater than or equal to 2.

b Trace your Turing machine, step-by-step, on the example of the number 2.

c Why in Turing machines (and in most actual computers) the representation of a binary number starts with the least significant digit?

5. The following finite automaton describes strings that end in 0:

- the starting state e; this state means that either we have not read any symbols yet or that the last read symbol was not 0; and

- the final state f meaning that the last symbol we read was 0.

Transitions are as follows:

- from the state e, symbol 0 leads to the state f and symbol 1 leads back to the state e;
- from the state f, symbol 1 leads to the state e and 0 leads back to the state f.

Use the general algorithm to transform this finite automaton into a Turing machine. Show, step-by-step, how your Turing machine will accept the string 110.

6. Give the formal definition of a feasible algorithm, and an explanation what practically feasible means. Give two examples different from what we had in class:

- an example of a computation time which is formally feasible, but not practically feasible, and
- an example of a computation time which is practically feasible but not formally feasible.

7. What is P? What is NP? What does it means for a problem to be NP-hard? NP-complete? Give brief definitions. Give an example of an NP-complete problem: explain what is the input, what is the desired output. Is P equal to NP?

8. Prove that the square root of 24 is not a rational number.

9. Formulate the halting problem. Prove that it is not possible to check whether a given program halts on given data.

10. Formulate Church-Turing thesis. Is it a mathematical theorem? Is it a statement about the physical world?