Automata, Computability, and Formal Languages
Spring 2025, Test 3

1. The following Turing machine replaces small letter a and b with capital one:

   - start, – → moving, R (here, – means blank)
   - moving, $a \to A$, R
   - moving, $b \to B$, R
   - moving, – → back, L
   - back, $A \to$ L
   - back, $B \to$ 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.

2. Arithmetic operations on Turing machines:

   a Design a Turing machine that subtracts 4 from a binary number.

   b Trace your Turing machine, step-by-step, on the example of the decimal number 14.

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

Turn over, please

3. The following finite automaton checks whether a sequence of letters has only capital letters. This automaton has two states: the starting state s which is also final, and the state e ("error"). Transitions are as follows:

- from the state s, symbols A and B lead back to s, while a and b lead to e;
- from the state e, each symbol leads back to e.

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 AB.

4. Give the formal definition of a feasible algorithm, and an explanation of 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.

These examples should be different from what we had in class, in homeworks, in last year's solutions. To make sure that your examples are different, use some numbers that have personal meaning to you – e.g., your birthday or digits from your car's licence plate.

5. 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?

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

7. Prove that the halting problem is not algorithmically solvable.

8. Give definitions of a decidable (recursive) language and of a semi-decidable (recursively enumerable, Turing-recognizable) language. Give an example of a decidable language and an example of a language which is semi-decidable but not decidable.

9. Prove that the cubic root of 24 is not a rational number.