Name: __________________________

General comments:

- you are allowed up to 5 pages of handwritten notes;
- if you need extra pages, place your name on each extra page;
- the main goal of most questions is to show that you know the corresponding algorithms; so, if you are running out of time, just follow the few first steps of the corresponding algorithm.

Good luck!

1-2. Prove that the language \( L = \{s^{2n}q^n r^{2n}\} = \{\Lambda, ssqrr, sssqqrrrr, \ldots\} \) is not context-free.

3. Trace the following Turing machine on the example of the word 10: start, \( \rightarrow \) work, R (here, \( \rightarrow \) means blank); work, 0 \( \rightarrow \) 1, R; work, 1 \( \rightarrow \) 0, R; work, \( \rightarrow \) back, L; back, 0 \( \rightarrow \) L; back, 1 \( \rightarrow \) L; back, \( \rightarrow \) halt. Explain how each step will be represented if we interpret the Turing machine as a finite automaton with two stacks.

4. Design a Turing machine that subtracts 1000 (binary version of 8) from a binary number which is larger than or equal to 1000. Trace your Turing machine, step-by-step, on the example of the string 1001. 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 binary strings with at most one 1. This automaton has:

- the starting state \( s \) which is also final; this state means that we have not read any 1s yet;
- the state \( s1 \) meaning that we have read one 1; this state is also final; and
- the state \( s2 \) meaning that we have read two or more 1s.

Transitions are as follows:

- from the state \( s \), symbol 0 leads back to state \( s \) and symbol 1 leads to state \( s1 \);
- from state \( s1 \), symbol 0 leads back to state \( s1 \) and 1 leads to state \( s2 \);
- from state \( s2 \), both symbols 0 and 1 lead back to the state \( s2 \).

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 010.

6. Give the formal definition of a feasible algorithm. 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 6 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?