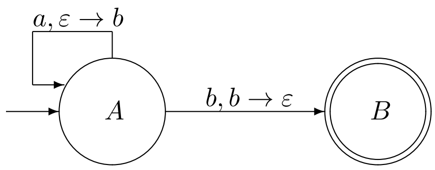
Automata, Computability, and Formal Languages
Fall 2025, Test 3

1. Show, step by step:

   1. how the stack-based algorithm will transform the expression $1-(1-2)$ into a postfix expression, and then

   2. how a second stack-based algorithm will compute the value of this expression.

2. Let us consider the following pushdown automaton:



This pushdown automaton accepts the word ab as follows:

	a	b
A	A	B
	b	

Use the general algorithm to show how this word will be generated in the corresponding context-free grammar.

3. Arithmetic operations on Turing machines:

   a Design a Turing machine that adds 4 to a binary number.

   b Trace your Turing machine, step-by-step, on the example of the decimal number 6, which is 110 in binary. Explain how each step will be represented if we interpret the Turing machine as a finite automaton with two stacks.

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

4. The following finite automaton accepts only words that end with a period. This automaton has two states: start state s and final state f, and three symbols: a, b, and the period. From both states, letters a and b lead to s, and the period . lead to 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 3-symbol string "*ab.*".

5. 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.

6. 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?

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

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

9. 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.

10. Prove that the cubic root of 16 is not a rational number.