Test 2 for CS 3350 Automata, Spring 2024

1–3. Let us consider a finite automaton that checks whether a test is challenging or too easy. Let us consider an alphabet consisting of two symbols: a (for "*a*dd a problem"), and d (for "*d*elete a problem"). This automaton has two states:

- the starting state c (for "*c*hallenging"), which is also final, and
- the state t (for "*t*oo easy").

Transitions are as follows:

- from the state c, d leads to t, while a lead back to c;
- from the state t, a leads to c, while d leads back to t.

This automaton accepts the word daa.

1. Show how the general algorithm will produce a context-free grammar that generates all the words accepted by this automaton – and only words generated by this automaton.
2. On the example of the word daa accepted by this automaton, show how the tracing of acceptance of this word by the finite automaton can be translated into a generation of this same word by your context-free grammar.
3. Show how the word daa can be represented as $uvxyz$ according to the Pumping Lemma for context-free grammars.

4-6. Let us consider the grammar with the starting variable C and the rules $C \to dT$, $C \to \varepsilon$, and $T \to aC$.

4. Use a general algorithm to construct a (non-deterministic) pushdown automaton that corresponds to this grammar.
5. Show, step by step, how the word da will be accepted by this automaton.
6. Transform this grammar into Chomsky normal form.

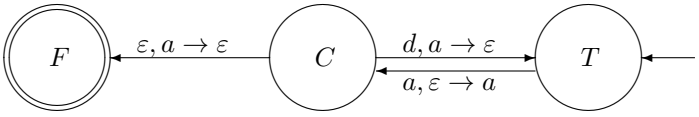
(Turn over, please.)

7-8. Show, step by step:

7. how the stack-based algorithm will transform the expression $3/(2/1)$ into a postfix expression, and then

8. how a second stack-based algorithm will compute the value of this expression (use integer operations in Java).

9-10. Let us consider the following pushdown automaton:



This pushdown automaton accepts the word *ada*. Use the general algorithm to show how this word will be generated in the corresponding context-free grammar.