# Test 2 for CS 3350 Automata, Fall 2023

1–3. Let us consider a finite automaton that checks whether a soup is hot or cold. Let us consider an alphabet consisting of two symbols: $w$ (for "$w$arm up"), and $\ell$ (for "$\ell$eave on the table"). This automaton has two states:

- the final state $h$ (for "$h$ot"), and
- the starting state $c$ (for "$c$old").

Transitions are as follows:

- from the state $h$, $w$ leads to $h$, while $\ell$ lead to $c$;
- from the state $c$, $w$ leads to $h$, while $\ell$ leads to back to $c$.

This automaton accepts the word $w\ell w$.

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 $w\ell w$ 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 $w\ell w$ 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 $C \to w\ell C$, $H \to \varepsilon$, and $C \to wH$.

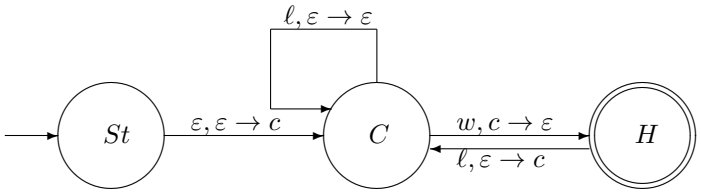
4. Use a general algorithm to construct a (non-deterministic) pushdown automaton that corresponds to this grammar.
5. Show, step by step, how the word $w\ell w$ will be accepted by this automaton.
6. Transform this grammar into Chomsky normal form.

7-8. Show, step by step:

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

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

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




This pushdown automaton accepts the word $w\ell w$. Use the general algorithm to show how this word will be generated in the corresponding context-free grammar.