

Solution to Homework 8

Tasks: In the corresponding lecture, we described an algorithm that, given a finite automaton, produces a context-free grammar – a grammar that generate a word if and only if this word is accepted by the given automaton.

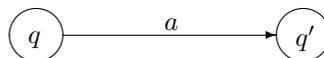
1. On the example from the automaton B from Homework 1.4, show how this algorithm will generate the corresponding context-free grammar. Similarly to Homework 3, assume that we only have symbols 1, a , and $?$.
2. On the example of a word 111 accepted by this automaton, show how the tracing of acceptance of this word by the finite automaton can be translated into a generation of this same word by your context-free grammar.

Reminder. This automaton has two states n and w . n is the starting state, and it is also the only final state. The transitions are as follows:

- from n , any digit leads to n , any other symbol leads to w ;
- from w , every symbol leads back to w .

Solution to Task 1. The general algorithm for transforming FA into CFG is as follows:

- To each state q of the FA, introduce a new variable Q .
- The variable corresponding to the starting state will be the starting variable of the new CFG.
- For each transition of the finite automaton



we add a rule $Q \rightarrow aQ'$.

- For each final state f of the FA, we add a rule $F \rightarrow \varepsilon$.

By applying this general algorithm to this FA, we get a CFG with two variables N and W , three terminal symbols 1, a , and $?$, the starting variable N and the following rules:

$$\begin{aligned} N &\rightarrow 1N \\ N &\rightarrow aW \\ N &\rightarrow ?W \\ W &\rightarrow 1W \\ W &\rightarrow aW \\ W &\rightarrow ?W \\ N &\rightarrow \varepsilon \end{aligned}$$

Solution to Task 2. Derivations in this grammar follow, step-by-step, the way the original finite automaton accepts a word. The word 111 is accepted by the original finite automaton as follows:

- we start in the start state n ; this corresponds to the starting variable N ;
- then, we use the fact that once we are in the state n and we see the symbol 1, then we move to the state n ; this transition corresponds to the rule $N \rightarrow 1N$, so the generation so far is:

$$\underline{N} \rightarrow 1N;$$

- then, we use the fact that once we are in the state n and we see the symbol 1, then we go back to the state n ; this transition corresponds to the rule $N \rightarrow 1N$, so generation so far is

$$\underline{N} \rightarrow 1\underline{N} \rightarrow 11N;$$

- then, we again use the fact that once we are in the state n and we see the symbol 1, then we go back to the state n ; this transition corresponds to the same rule $N \rightarrow 1N$, so generation so far is

$$\underline{N} \rightarrow 1\underline{N} \rightarrow 11\underline{N} \rightarrow 111N;$$

- we have read all the symbols of the word, and we are in the final state n ; for the FA, this means that the word 111 is accepted; for CFG, we need to use the rule $N \rightarrow \varepsilon$ corresponding to the final state n ; thus, we get the following derivation of the word 111:

$$\underline{N} \rightarrow 1\underline{N} \rightarrow 11\underline{N} \rightarrow 111\underline{N} \rightarrow 111.$$

So, we have indeed derived the word 111 in the grammar.