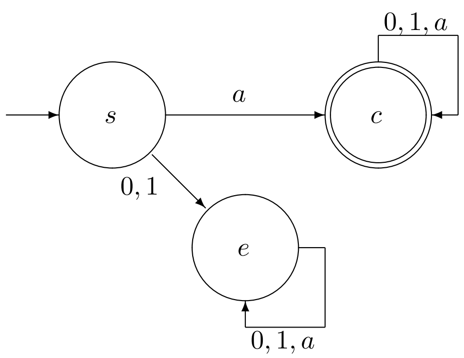
Solution to Homework 8

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 0, 1, and a.

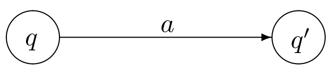
2. On the example of a word $a01$ accepted by this automaton, show how the tracing of acceptance of this word by the finite automaton can be translated into a generation of this same word by your context-free grammar.

Reminder. The automaton has the following form:



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 $Q \to aQ'$.

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

By applying this general algorithm to this FA, we get a CFG with 3 variables S, C, and E, three terminal symbols A, a, ansd 1, the starting variable S and the following rules:

$$S \to aC$$

$$S \to 0E$$

$$S \to 1E$$

$$C \to aC$$

$$C \to 0C$$

$$C \to 1C$$

$$E \to aE$$

$$E \to 0E$$

$$E \to 1E$$

$$C \to \varepsilon$$

Solution to Task 2. Derivations in this grammar follow, step-by-step, the way the original finite automaton accepts a word. The word $a01$ is accepted by the original finite automaton as follows:

- we start in the start state s; this corresponds to the starting variable S;

- then, we use the fact that once we are in the state s and we see the symbol a, then we move to the state c; this transition corresponds to the rule $S \to aC$, so the generation so far is:

$$\underline{S} \to aC;$$

- then, we use the fact that once we are in the state c and we see the symbol 0, then we go back to the state c; this transition corresponds to the rule $C \to 0C$, so generation so far is

$$\underline{S} \to a\underline{C} \to a0C;$$

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

$$\underline{S} \to a\underline{C} \to a0\underline{C} \to a01C;$$

- we have read all the symbols of the word, and we are in the final state c; for the FA, this means that the word $a01$ is accepted; for CFG, we need to use the rule $C \to \varepsilon$ corresponding to the final state c; thus, we get the following derivation of the word $a01$:

$$\underline{S} \to a\underline{C} \to a0\underline{C} \to a01\underline{C} \to a01.$$

So, we have indeed derived the word $a01$ in the grammar.