Solution to Homework 11

**Task.** Use the general algorithm to transform the pushdown automaton from Problem 6 into a context-free grammar. Show, step-by-step, how the resulting grammar will generate the word CLC.

**Solution.** Let us recall how the word CLC is accepted by this automaton:

<table>
<thead>
<tr>
<th>read</th>
<th>C</th>
<th>L</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>s</td>
<td>w</td>
<td>a₃</td>
</tr>
<tr>
<td>stack</td>
<td>$</td>
<td>$</td>
<td>C</td>
</tr>
</tbody>
</table>

We start with the state $s$, we end up in the final state $f$. Thus, the first rule we apply is the rule $S \rightarrow A_{sf}$:

$$ S \quad \rightarrow \quad A_{sf} $$

The first symbol we push is the dollar sign, this dollar sign is popped at the end. Thus, we have the following combination of pop-push rules:

$$ s \quad \varepsilon, \varepsilon \rightarrow $ \quad w \quad f \quad \varepsilon, \$ \rightarrow \varepsilon \quad f $$

In general, we have the two transitions

$$ p \quad x, \varepsilon \rightarrow t \quad q \quad r \quad y, t \rightarrow \varepsilon \quad s $$

What do we need to plug in instead of $p, q, etc.$ in the general 2-rule picture to come up with this particular picture:

- instead of $p$, we place $s$;
• instead of $q$, we place $w$;
• instead of $s$ and $t$, we place $f$;
• instead of $x$ and $y$, we place $\varepsilon$.

If we make these substitutions in the general rule:

$$A_{ps} \rightarrow xA_{qr}y,$$

we get the rule

$$A_{sf} \rightarrow \varepsilon A_{wf}\varepsilon.$$

Since concatenation with the empty string does not change anything, this means

$$A_{sf} \rightarrow A_{wf}.$$

Thus, the derivation so far takes the following form:

$$S \rightarrow A_{sf} \rightarrow A_{wf}.$$

We covered how we push the dollar sign and how we pop it. Let us underline what we have covered:

<table>
<thead>
<tr>
<th>read</th>
<th>C</th>
<th>L</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>s</td>
<td>w</td>
<td>$\varepsilon$</td>
</tr>
<tr>
<td>stack</td>
<td>$$</td>
<td>$$</td>
<td>C</td>
</tr>
</tbody>
</table>

If we ignore the dollar signs – since we already took care of them – then we see that we have three intermediate states with the empty stack. So, we need to use the transitivity rule, which in this case takes the form

$$A_{wf} \rightarrow A_{wa_3}A_{a_3w}A_{wa_3}A_{a_3f}.$$

Thus, the derivation tree takes the following form:
Transition $A_{wa3}$. In the first transition from $w$ to $a_1$ we have only one rule: $C, \$, \rightarrow \$. In this rule, we pop \$ and then push it, so, in effect, nothing changes in the stack. Thus, in this case, the use of this rule is equivalent to $C, \varepsilon \rightarrow \varepsilon$. Since there is only one rule, according to the general algorithm, we need to add a fictitious rule to form a pair. We ignore the dollar signs that do not change:

This combination leads to the rule $A_{wa3} \rightarrow CA_{a3a3} \varepsilon$, i.e., $A_{wa3} \rightarrow CA_{a3a3}$.

Here, the remaining transition between $a_3$ and $a_3$ does not include any additional steps, so we can use the rule $A_{a3a3} \rightarrow \varepsilon$. So, we get the rule $A_{wa3} \rightarrow C$.

Transition $A_{a3w}$. For the transition between $a_3$ and $w$, we push $C$ and then pop $C$:

So, we have a transition $A_{a3w} \rightarrow \varepsilon A_{wa} L$, i.e., $A_{a3w} \rightarrow A_{wa} L$. Here, the remaining transition between $w$ and $w$ does not include any additional steps, so we can use the rule $A_{ww} \rightarrow \varepsilon$. So, we get the rule $A_{a3w} \rightarrow L$.

Transition $A_{wa3}$. The next transition $A_{wa3}$ is the same as the first transition from $w$ to $a_3$, so it corresponds to the rule $A_{wa3} \rightarrow C$.

Transition $A_{a3f}$. In the last transition $A_{a3f}$, we first push $C$, then pop $C$:

So, we get the transition $A_{a3f} \rightarrow \varepsilon A_{wf} \varepsilon$, i.e., $A_{a3f} \rightarrow A_{wf}$.

This takes care of the second $C$, so we get:
For the remaining transition from $w$ to $f$, there is only one rule – in which the stack does not change. Since there is only one rule, we pair it with a fictitious trivial rule:

$$A_{wf} \rightarrow \varepsilon A_{ff} \varepsilon$$

So, we get $A_{wf} \rightarrow \varepsilon A_{ff} \varepsilon$, i.e., $A_{wf} \rightarrow A_{ff}$. There are no rules for the transition $A_{ff}$, so we get $A_{ff} \rightarrow \varepsilon$ and thus, $A_{wf} \rightarrow \varepsilon$. So, the generation of CLC takes the following form: