Solution to Homework 11

Task. In Homework 6, we considered the following pushdown automaton. This pushdown automaton has three states:

- the starting state s,
- the state b meaning that so far, we have earned at least as much as we spent, and
- the final state f.

The transitions are as follows:

- From s to b, the transition is $\varepsilon, \varepsilon \to \$$;
- From b to b, the transitions are: $e, \varepsilon \to d$ and $s, d \to \varepsilon$.
- From b to f, the transition is: $\varepsilon, \$ \to \varepsilon$.

Use the general algorithm to transform this pushdown automaton into a context-free grammar. Show, step-by-step, how the resulting grammar will generate the word *eess*.

Solution. Let us recall how the word *eess* is accepted by this automaton. We will list consequent states and the contents of the corresponding stacks, described from the top to bottom, and what symbols we see in the corresponding transitions:

- state s, stack is empty;

we jump from state s to state b and push \$ into the stack;

- state b, stack has \$;

we read e and use the rule $e, \varepsilon \to d$ to push d into the stack;

- state b, stack has d on top of \$;

we read s and use the rule $s, d \to \varepsilon$ to pop d from the stack:

- state b, stack has \$:
- state b, stack has d on top of \$;

we read s and use the rule $s, d \to \varepsilon$ to pop d from the stack:

- state b, stack has \$:

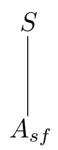
we use the rule $\varepsilon, \$ \to \varepsilon$ to go to the final state:

- state f, stack is empty.

These transitions can be described as follows:

read			e		s		e		s		
state	s	b	$\to$	b	$\to$	b	$\to$	b	$\to$	b	f
stack		\$	$\to$	d \$	$\to$	\$	$\to$	d \$	$\to$	\$	

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



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:

In general, we have the two transitions

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 and r, we place b;
- instead of s, we place f;
- instead of x and y, we place ε;
- instead of t, we place \$.

If we make these substitutions in the general rule:

$$A_{ps} \to x A_{qr} y,$$

we get the rule

$$A_{sf} \to \varepsilon A_{bb} \varepsilon.$$

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

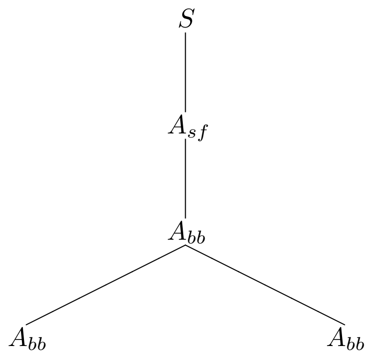
$$A_{sf} \to A_{bb}.$$

Thus, the derivation so far takes the following form:

We covered the transition from s to and from b to f. Let us underline what we have covered:

read			e		s		e		s		
state	$\underline{s}$	b	$\to$	b	$\to$	b	$\to$	b	$\to$	b	$\underline{f}$
stack		$\underline{\$}$	$\to$	d $\underline{\$}$	$\to$	$\underline{\$}$	$\to$	d $\underline{\$}$	$\to$	$\underline{\$}$	

If we ignore the dollar signs – since we already took care of them – then we see that we have one intermediate state with the empty stack. So, we need to use the transitivity rule, which in this case takes the form $A_{bb} \to A_{bb} A_{bb}$. Thus, the derivation tree takes the following form:



In each of the two transitions from b with an empty stack to b with an empty stack, we first push d, and then immediately pop d. Thus, we have the following combination of push-pop rules:

$$b \xrightarrow{e, \varepsilon \to d} b \qquad\qquad b \xrightarrow{s, d \to \varepsilon} b$$

This combination leads to the rule $A_{bb} \to eA_{bb}s$. Here, the remaining transition between b and b does not include any additional steps, so we can use the rule $A_{bb} \to \varepsilon$. So, we get the following final derivation: