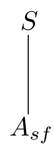
Solution to Homework 11

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 ABAA.

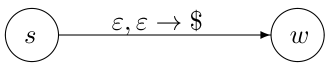
Solution. Let us recall how the word ABAA is accepted by this automaton:

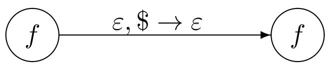
read			A		B	A		A				
state	s	w	a_1	w	w	a_1	w	a_2	w	f	f	f
stack		\$	\$	A \$	\$	\$	A \$	A \$	A A \$	A \$	\$	

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

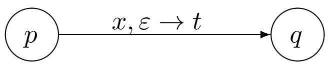


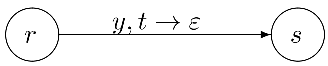
The first symbol we push is the dollar sign, this dollar sign is popped at the end. Thus, we have the following combination of pop-push rules:





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, we place w;
- instead of s and t, we place f;
- instead of x and y, we place ε.

If we make these substitutions in the general rule:

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

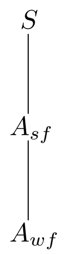
we get the rule

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

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

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

Thus, the derivation so far takes the following form:



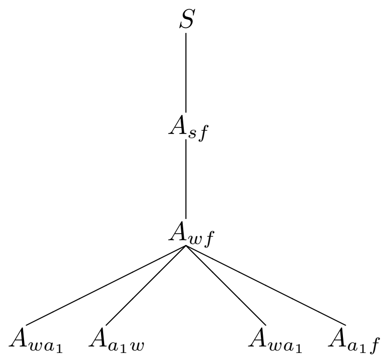
We covered how we push the dollar sign and how we pop it. Let us underline what we have covered:

read			A		B	A		A				
state	s	w	a_1	w	w	a_1	w	a_2	w	f	f	f
stack		$\underline{\$}$	$\underline{\$}$	A $\underline{\$}$	$\underline{\$}$	$\underline{\$}$	A $\underline{\$}$	A $\underline{\$}$	A A $\underline{\$}$	A $\underline{\$}$	$\underline{\$}$	

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

$$A_{wf} \to A_{wa_1}A_{a_1w}A_{wa_1}A_{a_1f}.$$

Thus, the derivation tree takes the following form:



Transition A_{wa_1}**.** In the first transition from w to a_1 we have only one rule: $A, \$ \to \$$. 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 $A, \varepsilon \to \varepsilon$. 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 $A_{wa_1} \to \mathrm{A}A_{a_1a_1}\varepsilon$, i.e., $A_{wa_1} \to \mathrm{A}A_{a_1a_1}$. Here, the remaining transition between a_1 and a_1 does not include any additional steps, so we can use the rule $A_{a_1a_1} \to \varepsilon$. So, we get the rule $A_{wa_1} \to \mathrm{A}$.

Transition A_{a_1w}**.** For the transition between a_1 and w, we push A and then pop A:

So, we have a transition $A_{a_1w} \to \varepsilon A_{ww}\mathrm{B}$, i.e., $A_{a_1w} \to A_{ww}\mathrm{B}$. Here, the remaining transition between w and w does not include any additional steps, so we can use the rule $A_{ww} \to \varepsilon$. So, we get the rule $A_{a_1w} \to \mathrm{B}$.

Transition A_{wa_1}**.** The next transition A_{wa_1} is the same as the first transition from w to a_1, so it corresponds to the rule $A_{wa_1} \to \mathrm{A}$.

Transition A_{a_1f}**.** In the last transition A_{a_1f}, we first push A, then pop A:

So, we get the transition $A_{a_1f} \to \varepsilon A_{wf}\varepsilon$, i.e., $A_{a_1f} \to A_{wf}$. This takes care of the second A, so we get:

read			A		B	A		A				
state	s	w	a_1	w	w	a_1	w	a_2	w	f	f	f
stack		\$	\$	A \$	\$	\$	A \$	A \$	A A \$	A \$	\$	

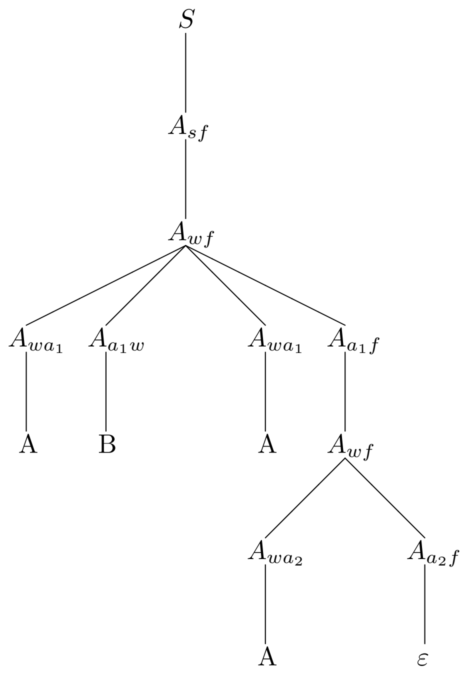
Now, if we ignore the second A, we get an intermediate state a_2 with an empty stack, so we need to use transitivity $A_{wf} \to A_{wa_2} A_{a_2 f}$.

For A_{wa_2}, there is only one rule – in which the stack does not change, so its application is equivalent to $A, \varepsilon \to \varepsilon$. Since there is only one rule, we pair it with a fictitious trivial rule:

So, we get $A_{wa_2} \to \mathrm{A} A_{a_2 a_2} \varepsilon$, i.e., $A_{wa_2} \to \mathrm{A} A_{a_2 a_2}$. There are no rules for the transition $A_{a_2 a_2}$, so we get $A_{a_2 a_2} \to \varepsilon$ and thus, $A_{wa_2} \to \mathrm{A}$.

Finally, for the transition $A_{a_2 f}$, we push and pop the last A by using the following rules:

Here, we get the rule $A_{a_2 f} \to \varepsilon A_{ww} \varepsilon$, i.e., $A_{a_2 f} \to A_{ww}$. There are no rules for the transition A_{ww}, so we get $A_{ww} \to \varepsilon$ and thus, $A_{wa_2} \to \varepsilon$. So, the generation of ABAA takes the following form:


S
A_sf
A_wf
A_wa1
A_a1w
A_wa1
A_a1f
A
B
A
A_wf
A_wa2
A_a2f
A
ε