Solution to Homework 9

**Background.** In Problem 7, we considered a grammar with rules $I \rightarrow +U$, $I \rightarrow -U$, $I \rightarrow U$, $U \rightarrow DU$, $U \rightarrow D$, $D \rightarrow 0$, and $D \rightarrow 1$.

**Tasks:**

1. Use a general algorithm to construct a (non-deterministic) pushdown automaton that corresponds to context-free grammar described in Problem 7.

2. Show, step by step, how the word +101 will be accepted by this automaton.

**Solution to Task 1.** By using the general algorithm, we get the following pushdown automaton:
Solution to Task 2. Let us show how this is done on the example of the word +110 generated by the above automaton:

\[ I \rightarrow +U \rightarrow +DU \rightarrow +1U \rightarrow +1DU \rightarrow +11U \rightarrow +11D \rightarrow +110. \]

To make this derivation clearer, let us mark the variables \( U \) and \( D \) corresponding to different transitions by subscripts:

\[ I \rightarrow +U_1 \rightarrow +D_1U_2 \rightarrow +1U_2 \rightarrow +1D_2U_3 \rightarrow +11U_3 \rightarrow +11D_3 \rightarrow +110. \]

Let us now trace what our pushdown automaton will do. We start in the state \( s \) with an empty stack:
The only thing we can do when in the state $s$ is push the dollar sign into the stack and get to the intermediate state $i$: 
The contents of the stack is as follows:

When we are in the state $i$, the only thing we can do is push the starting variable $I$ into the stack and go into the working state $w$. 

4
Now, the stack contains the starting variable on top of the dollar sign:

\[
\begin{array}{c}
I \\
\$ \\
\end{array}
\]

Now that we are in the working state, we can start following the rules that were used to derive the word +110. The first rule was \( I \rightarrow +U \), or, to be precise, \( I \rightarrow +U_1 \). As we have mentioned, this rule is implemented in two steps:

1. first, we pop \( I \) and push the last symbol of the right-hand side – in this cases, the symbol \( U \) (that corresponds to the first occurrence \( U_1 \)) – into the stack, getting into the auxiliary state \( a_1 \);

2. then, we push \( + \) into the stack, and go back to the working state \( w \).

Let us illustrate this step by step.

First, we pop \( I \), push \( U \), and go into the state \( a_1 \):
The stack will now have $U$ instead of the original $I$:

```
U
$  
```

Then, we push $+$ into the stack and go back to working state $w$: 

6
The stack will now have + on top of its previous contents:

```
+  
U  
$  
```

Now, the symbol + is top of the stack. The only thing we can do if a terminal symbol is on top of the stack is use one of the rules of the type $x, x \rightarrow \varepsilon$ where $x$ stands for the corresponding terminal symbol.

In our case:

- since the terminal symbol on top of the stack is the symbol +,
- we need to use the rule $+, + \rightarrow \varepsilon$,

i.e., we read the symbol + from the original word +110 and pop the top symbol + from the stack. We remain in the same state $w$, but the stack changes. The stack now has the following form:
In the derivation of our word, next, we use the rule $U \rightarrow DU$. As before, this rule is implemented in two steps:

- first, we pop $U$ and push the last symbol of the right-hand side – in this cases, the symbol $U$ into the stack, getting into the auxiliary state $a_3$;
- finally, we push $D$ into the stack, and go back to the working state $w$.

Let us illustrate this step by step.

First, we pop $U$, push $U$, and go into the state $a_3$:

The stack remains unchanged. Next, we push $D$ into the stack and get back to the working state:
The stack will now have $D$ on top:

\[
\begin{array}{c}
D \\
U \\
\$
\end{array}
\]

Then, we use the rule $D \to 1$, which corresponds to the rule $\varepsilon, D \to 1$ of the pushdown automaton. Namely, we replace $D$ on top of the stack with 1.
Now, the stack has the form:

```
1
U
$
```

On top of the stack is a terminal symbol 1. The only way to delete it from the stack is to use the rule $1, 1 \rightarrow \varepsilon$, i.e., to read symbol 1 and pop 1 from the top of the stack. We still remain in the working state, but the stack changes to

```
U
$
```

Next, we again use the rule $U \rightarrow DU$ of the grammar. So, first, we replace $U$ on top of the stack with $U$ and go to state $a_3$: 
The stack takes the following form:

\[
\begin{array}{c}
U \\
S
\end{array}
\]

Next, we push $D$ into the stack and go back to the working state:
The stack now has the form:

\[
\begin{array}{c}
D \\
U \\
$ \\
\end{array}
\]

Next, we use the rule \( D \to 1 \) from the grammar, which corresponds to the rule \( \varepsilon, D \to 1 \) of the pushdown automaton. We replace \( D \) on top of the stack with 1, and stay in the same state \( w \). The stack now takes the following form:

\[
\begin{array}{c}
1 \\
U \\
$ \\
\end{array}
\]

On top of the stack is a terminal symbol 1. So eliminate this symbol from the stack, we use the rule \( 1, 1 \to \varepsilon \), i.e., we read the symbol 1 and delete 1 from the top of the stack. Then, the stack takes the following form:
After this, we use the rule $U \rightarrow D$ of the grammar, that corresponds to the rule $\varepsilon, U \rightarrow D$ of the pushdown automaton. Namely, we replace $U$ on top of the stack with $D$. Thus, the stack takes the following form:

\[
\begin{array}{c}
D \\
\$ \\
\end{array}
\]

Then, we use the rule $D \rightarrow 0$ of the grammar, that corresponds to the rule $\varepsilon, D \rightarrow 0$ of the pushdown automaton. Namely, we replace $D$ on top of the stack with $0$. Thus, the stack takes the following form:

\[
\begin{array}{c}
0 \\
\$ \\
\end{array}
\]

On top of the stack is a terminal symbol $0$. So eliminate this symbol from the stack, we use the rule $0, 0 \rightarrow \varepsilon$, i.e., we read the last symbol $0$ of the word and delete $0$ from the top of the stack. Then, the stack takes the following form:

\[
\begin{array}{c}
\$ \\
\end{array}
\]

We have read all the symbols of the original word, and we only have the dollar sign remaining in the stack. Thus, we can use the rule $\varepsilon, \$, $\rightarrow \varepsilon$ to delete the dollar sign from the stack and to move to the final state $f$:  

13
The stack is now empty. We have read all the symbols of the given word and we end up in the final state with the empty stack. Thus, the word +110 is accepted.

A graphical description of the transitions.

<table>
<thead>
<tr>
<th>read</th>
<th>state</th>
<th>stack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>s</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>i</td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>a_1</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>f</td>
<td>$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>read</th>
<th>state</th>
<th>stack</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>$</td>
</tr>
<tr>
<td></td>
<td>w</td>
<td>$</td>
</tr>
</tbody>
</table>
|      | w     | $     | 14