Solution Set to Homework Assignment 7 (40 points)

Due date: 4/8/2016

1. The following empty-stack PDA accepts strings with the same number of a’s and b’s. Transform this PDA into a C-F grammar. Your solution should be clearer than the work given in the notes such as which type is used for each step. (20 points)

Solution:
In the following, we list the PDA instructions on the left and the corresponding grammar production on the right. We start with Type 4 first.

Type 4:
The start state and the PDA instruction $\Lambda, X/pop$ as shown below gives:

$$S \rightarrow X_{00}$$

Type 1:
The PDA instruction $\Lambda, X/pop$ by itself as shown below gives:

$$X_{00} \rightarrow \Lambda$$

The PDA instruction $b, a/pop$ by itself as shown below gives:

$$A_{00} \rightarrow b$$
The PDA instruction $a, b/pop$ by itself as shown below gives:

Type 3:
The PDA instruction $a, X/push(A)$ together with $b, a/pop$ and $\Lambda, X/pop$ as shown below gives:

The PDA instruction $b, X/push(B)$ together with $a, B/pop$ and $\Lambda, X/pop$ as shown below gives:

The PDA instruction $a, A/push(A)$ and $b, A/pop$ (used twice) as shown below gives:

The PDA instruction $b, B/push(B)$ and $a, B/pop$ (used twice) as shown below gives:

There are no productions in the Type 2 category because there are no PDA instructions that perform nop stack operation.

Collecting all the productions, we get a context-free grammar whose production set is as follows:
After simplification, we get

\[
\begin{align*}
S & \rightarrow \Lambda \mid aAS \mid bBS \\
A & \rightarrow b \mid aAA \\
B & \rightarrow a \mid bBB
\end{align*}
\]

2. Given the following PDA, transform it into a C-F grammar using the algorithm shown in class. (20 points)

Or, instead, you can use the following PDA to do the transformation.

\[
\begin{align*}
(0, a, a, \text{pop}, 0) \\
(0, b, b, \text{pop}, 0) \\
(0, \Lambda, S, \text{pop}, 0) \\
(0, \Lambda, S, \langle \text{pop, push}(a), \text{push}(S), \text{push}(a) \rangle, 0) \\
(0, \Lambda, S, \langle \text{pop, push}(b), \text{push}(S), \text{push}(b) \rangle, 0)
\end{align*}
\]

Or, instead, you can use the following PDA to do the transformation.

\[
\begin{align*}
\frac{a, X}{\text{push}(a)} & \quad \frac{a, a}{\text{push}(a)} & \quad \frac{a, b}{\text{push}(a)} & \quad \frac{a, a}{\text{pop}} & \quad \frac{b, b}{\text{pop}} \\
X & \quad \frac{\Lambda, X}{\text{nop}} & \quad \frac{\Lambda, a}{\text{nop}} & \quad \frac{\Lambda, b}{\text{nop}} & \quad \frac{\Lambda, X}{\text{pop}} \\
0 & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}} \\
1 & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}} \\
2 & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}} & \quad \frac{\text{nop}}{\text{nop}}
\end{align*}
\]

**Solution:**

We shall call the given PDA P1 and the one below it P2. Note that P1 can be expressed
In graphical form as follows:

```
<table>
<thead>
<tr>
<th></th>
<th>a, a</th>
<th>b, b</th>
<th>(\Lambda, X)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pop</td>
<td>pop</td>
<td>pop</td>
<td>pop</td>
</tr>
</tbody>
</table>
```

In the following, we transform P1 into a C-F grammar first and then transform P2 into a C-F grammar.

**Transform P1 into a C-F grammar:**
As in the first problem, we list the PDA instructions on the left and the corresponding grammar production on the right. We start with Type 4 first.

**Type 4:**
The start state and the PDA instruction \(\Lambda, X/pop\) as shown below gives:

```
\(X\)  \(\Lambda, X/pop\)  \(S \rightarrow X_{00}\)
```

The start state and the PDA instruction \(\Lambda, X/pop, push(a,X,a)\), and the start state and the PDA instruction \(\Lambda, X/pop, push(b,X,b)\) would each generate the same production. So we ignore these two Type 4 cases.

**Type 1:**
The PDA instruction \(\Lambda, X/pop\) by itself as shown below gives:

```
\(X_{00} \rightarrow \Lambda\)
```

The PDA instruction \(a,a/pop\) by itself as shown below gives:
The PDA instruction $b,b/pop$ by itself as shown below gives:

$$A_{00} \rightarrow a$$

The PDA instruction $\lambda,X/pop,push(a,X,a)$, and the PDA instruction $\lambda,X/pop,push(b,X,b)$ would each generate the same production as $\lambda,X/pop$. So we ignore these two Type 1 cases as well.

**Type 2:**
None

**Type 3:**
The PDA instruction $\lambda,X/pop,push(a,X,a)$ together with $a,a/pop$, $\lambda,X/pop$ and $a,a/pop$ as shown below gives:

$$X_{00} \rightarrow A_{00}X_{00}A_{00}$$

The PDA instruction $\lambda,X/pop,push(b,X,b)$ together with $b,b/pop$, $\lambda,X/pop$ and $b,b/pop$ as shown below gives:

$$X_{00} \rightarrow B_{00}X_{00}B_{00}$$
Collecting all the productions, we get the production set of a context-free grammar as follows:

\[
\begin{align*}
S & \rightarrow X_{00} \\
X_{00} & \rightarrow \Lambda \\
A_{00} & \rightarrow a \\
B_{00} & \rightarrow b \\
X_{00} & \rightarrow A_{00}X_{00}A_{00} \\
X_{00} & \rightarrow B_{00}X_{00}B_{00}
\end{align*}
\]

After direct substitution and simplification, we get

\[
S \rightarrow \Lambda | aSa | bSb
\]

**Transform P2 into a C-F grammar:**

Here we use the same convention as above, we list the PDA instructions on the left and the corresponding grammar production on the right. We start with Type 4 first.

**Type 4:**

The start state and the PDA instruction \(\Lambda, X/pop\) between state 1 and state 2 as shown below gives:

**Type 1:**

The PDA instruction \(\Lambda, X/pop\) by itself as shown below gives:

The PDA instruction \(a, a/pop\) by itself as shown below gives:
The PDA instruction $b,b/\text{pop}$ by itself as shown below gives:

Type 2:
The PDA instructions $\Lambda,X/\text{nop}$ and $\Lambda,X/\text{pop}$ as shown below gives:

The PDA instructions $\Lambda,a/\text{nop}$ and $a,a/\text{pop}$ as shown below gives:

The PDA instructions $\Lambda,b/\text{nop}$ and $b,b/\text{pop}$ as shown below gives:

Type 3:
The PDA instruction $a,X/\text{push}(a)$ together with $a,a/\text{pop}$ and $\Lambda,X/\text{pop}$ as shown below gives:

The PDA instruction $a,a/\text{push}(a)$ together with $a,a/\text{pop}$ (used twice) as shown below gives:
The PDA instruction \( b,X/push(b) \) together with \( b,b/pop \) and \( \Lambda,X/pop \) as shown below gives:

![Diagram 1](image1)

The PDA instruction \( b,b/push(b) \) together with \( b,B/pop \) (used twice) as shown below gives:

![Diagram 2](image2)

The PDA instruction \( a,b/push(a) \) together with \( a,a/pop \) and \( b,b/pop \) as shown below gives:

![Diagram 3](image3)

The PDA instruction \( b,a/push(b) \) together with \( b,b/pop \) and \( a,a/pop \) as shown below gives:

![Diagram 4](image4)

Collecting all the productions, we get the production set of a context-free grammar as follows:

\[
\begin{align*}
S & \rightarrow X_{02} \\
X_{02} & \rightarrow X_{12} | a A_{01} X_{12} | b B_{01} X_{12} \\
A_{01} & \rightarrow A_{11} | a A_{01} A_{11} | b B_{01} A_{11} \\
B_{01} & \rightarrow B_{11} | b B_{01} B_{11} | a A_{01} B_{11} \\
A_{11} & \rightarrow a \\
B_{11} & \rightarrow b \\
X_{12} & \rightarrow \Lambda
\end{align*}
\]

After a direct substitution, we have:

\[
\begin{align*}
S & \rightarrow \Lambda | a A | b B \\
A & \rightarrow a | a A | a | b B | a \\
B & \rightarrow b | b B b | a A b
\end{align*}
\]

( * )
In the following we prove that this production set is equivalent to the following one:

\[ S \rightarrow \Lambda \mid aSa \mid bSb \]  

( ** )

Note that from ( * ) we have

\[ S \Rightarrow \Lambda \]
\[ S \Rightarrow aA \Rightarrow aa = a\Lambda a \]
\[ S \Rightarrow aA \Rightarrow aaA = aaaa = aa\Lambda aa \]
\[ S \Rightarrow aA \Rightarrow aaAa = aaaa = aaaa = aaaa = aaaa\Lambda aaa \]

By induction, we can prove that

\[ S \Rightarrow a^n a^n = a^n \Lambda a^n \]  

for any n

Hence, ( * ) implies

\[ S \rightarrow \Lambda \mid aSa \]  

($$ )

Similarly, we can prove that ( * ) implies

\[ S \rightarrow \Lambda \mid bSb \]  

($$$)

To show that ( * ) implies ( ** ), note that

\[ S \Rightarrow aA \Rightarrow abBa = abba = ab\Lambda ba \]
\[ S \Rightarrow bB \Rightarrow baAb = baab = ba\Lambda ab \]

Hence, together with ($$) and ($$$), we have

\[ S \rightarrow \Lambda \mid aSa \mid bSb \]  

( ** )

QED

- Solutions must be typed (word processed) and submitted by email both as a pdf file and a word file to your grader at: sriharitha.ambati@uky.edu before 23:59 on 4/8/2016.