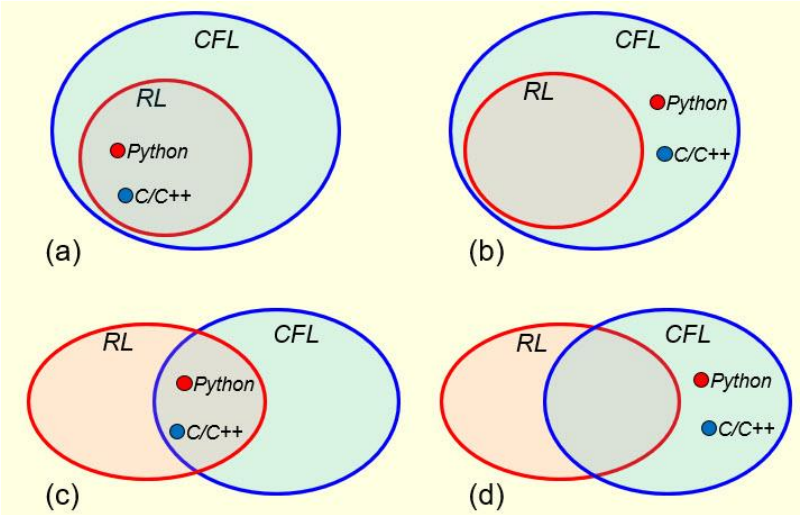
CS375 Homework Assignment 5 Solution Set (40 points)

Due date: 03/06/2025

1. (2 points)
   In the following (a), (b), (c) and (d) four cases, which one is the correct representation of the relationship between Regular Languages (RL) and Context-Free Languages (CFL) and the memberships of the Python and C/C++ programming languages?



Put your answer in the following text box.

(b)

2. (5 points)
   For the PDA's shown in Fig. 1 and Fig. 2, which one/ones or no one would accept the language $L = \{x \in \{a, b\}^* \mid N_b(x) = 1 + N_a(x)\}$ by empty stack? Use the execution of the string *abbba* to justify your answer. The execution should be carried out using 'Instantaneous Descriptions'.

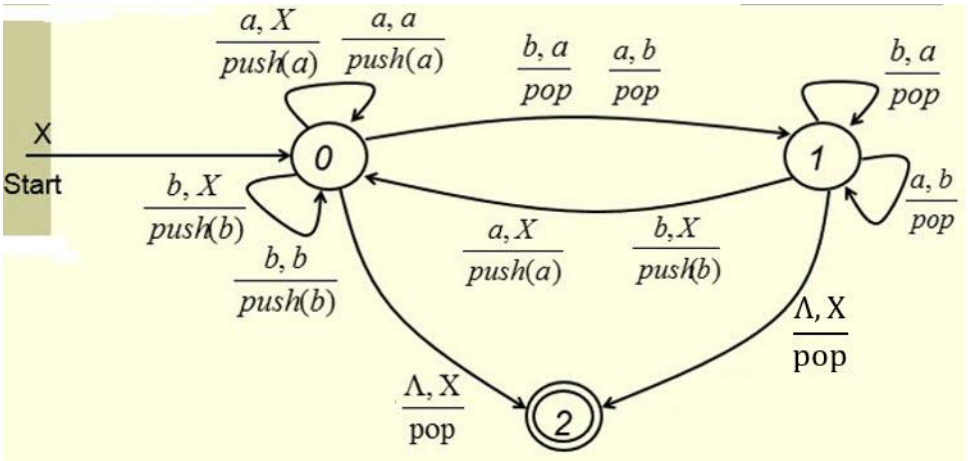


Fig. 1

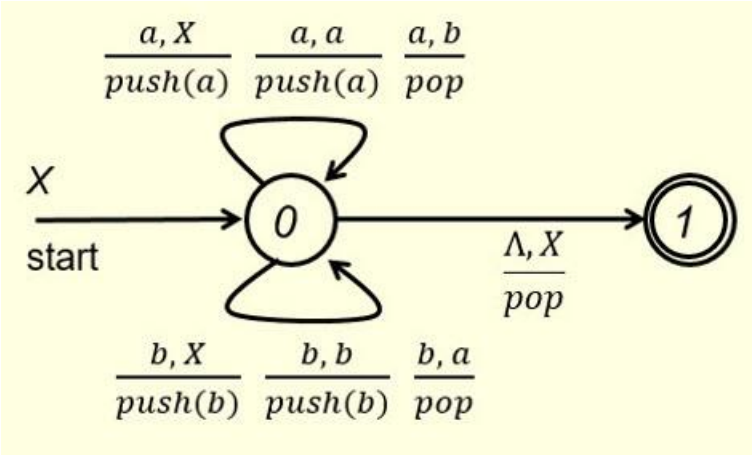


Fig. 2

Sol.

For the PDA in Fig. 1, we have

(0, abbba, X)
(0, bbba, aX)
(1, bba, X)
(0, ba, bX)
(0, a, bbX)
(1, Λ, bX)
We got stuck here. The stack is not empty. So this PDA does not accept abbba

For the PDA in Fig. 2, we have

(0, abbba, X)
(0, bbba, aX)
(0, bba, X)
(0, ba, bX)
(0, a, bbX)
(0, Λ, bX)
We got stuck here. The stack is not empty. So this PDA does not accept abbba

3. (4 points)

The language accepted by the final-state PDA shown in Fig. 1 below is:

$\{a^n b^{n+1} \mid n \in N\}$ (1 point)

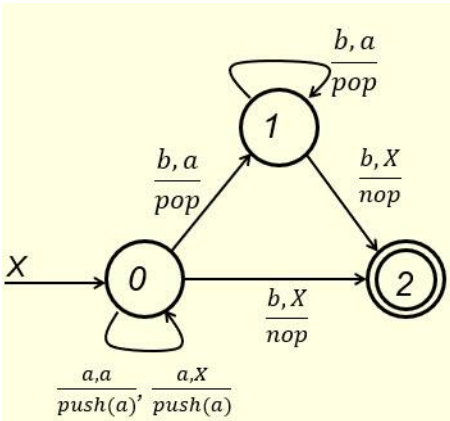


Fig. 1

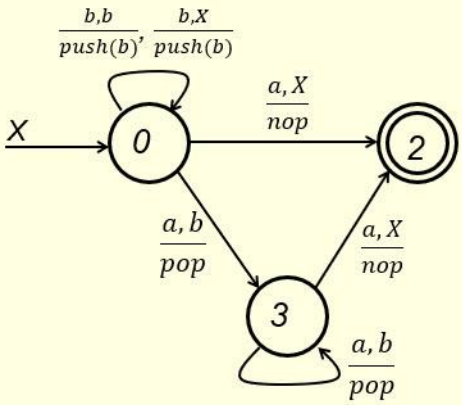


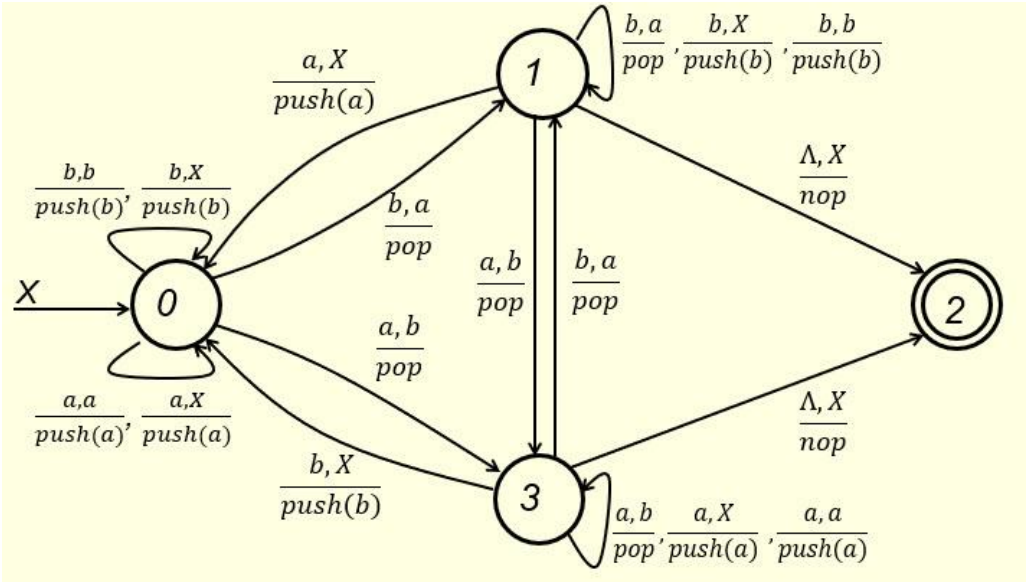
Fig. 2

The language accepted by the final-state PDA shown in Fig. 2 is:

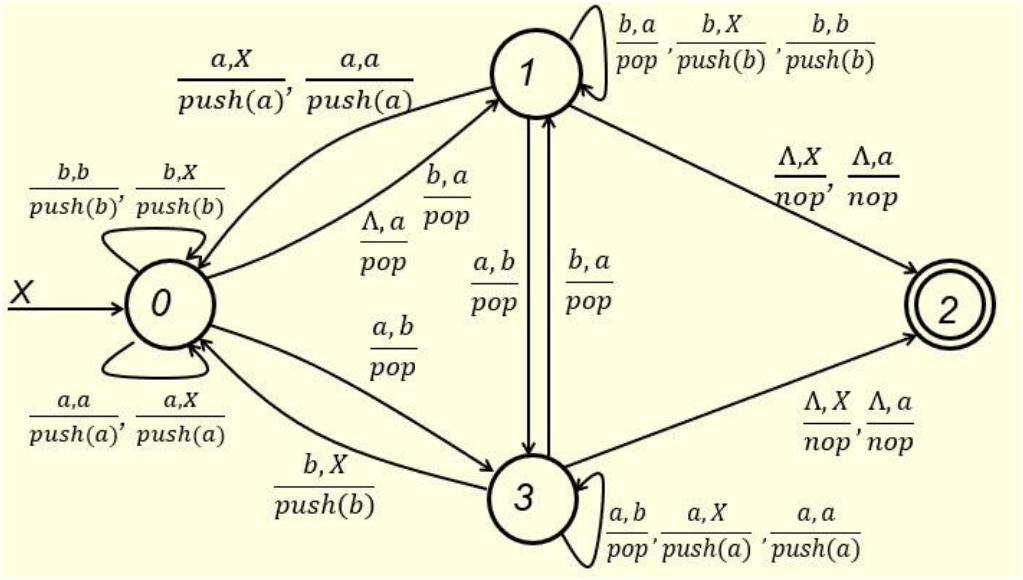
$\{b^n a^{n+1} \mid n \in N\}$ (1 point)

If the language accepted by the following final-state PDA is:

$\{\ w \in \{a,b\}^+ \mid N_a(w) = N_b(w)\ \}$



then what is the language of the following final-state PDA?



Put your answer in the following text box. (2 points)

$$\{\, w \in \{a,b\}^+ \mid N_a(w) = 1 + N_b(w) \,\}$$

4. (4 points)

The language generated by the following C-F grammar

$$S \rightarrow AB \mid c \mid d \qquad A \rightarrow aA \mid \Lambda \qquad B \rightarrow bB \mid \Lambda$$

is $\{a^m b^n + c + d \mid m, n \in N\}$. To transform the above C-F grammar to a one-state empty-stack PDA so that the accepted language of the PDA is the same as the language generated by this C-F grammar, we need to construct a set of specific instructions for this PDA. Four instructions have already been constructed below. Construct the remaining instructions for this PDA in the following blanks. You might not need all the provided blanks.

(0, a, a, pop, 0)
(0, b, b, pop, 0)
(0, c, c, pop, 0)
(0, d, d, pop, 0)

(0, Λ, S, <pop, push(B), push(A)>, 0)

(0, Λ, B, <pop, push(B), push(b)>, 0)

(0, Λ, B, pop, 0)

(0, Λ, A, <pop, push(A), push(a)>, 0)

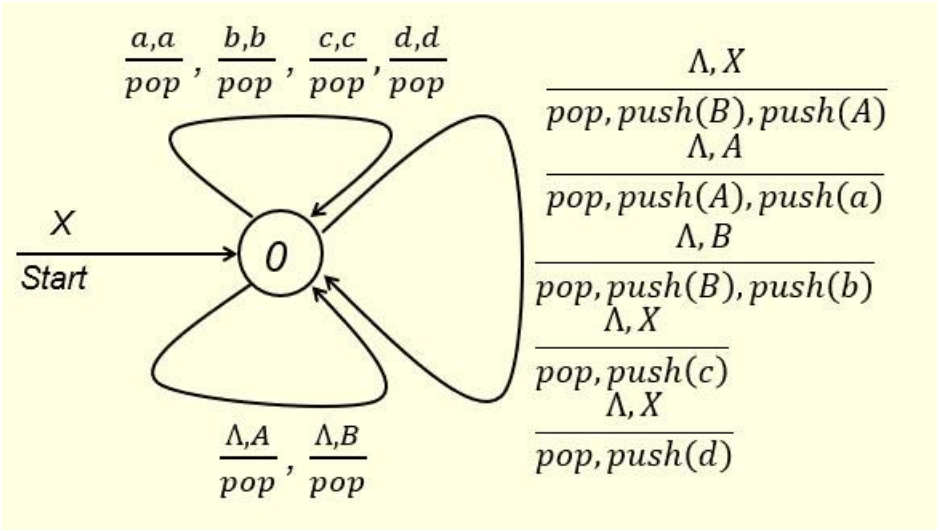
(0, Λ, A, pop, 0)

(0, Λ, S, <pop, push(c)>, 0)

(0, Λ, S, <pop, push(d)>, 0)

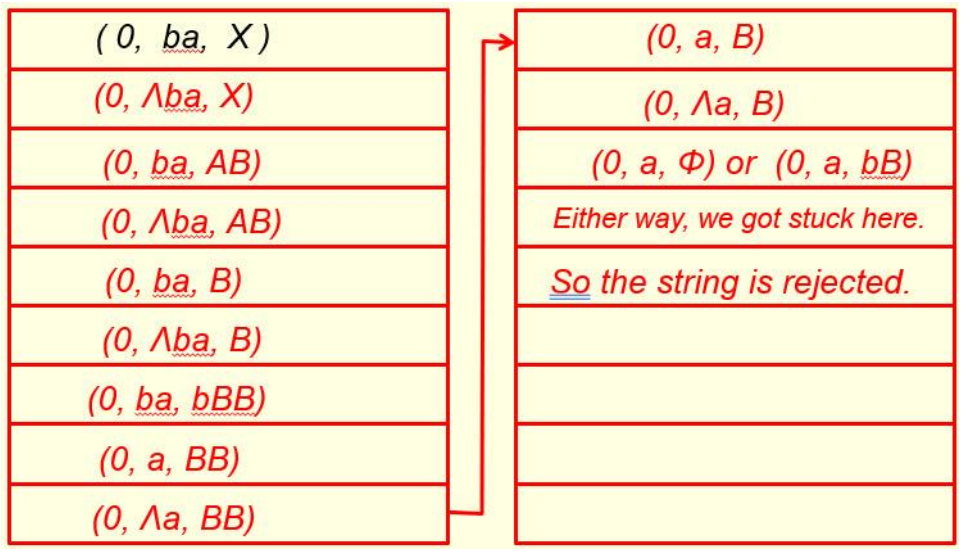
5. (10 points)

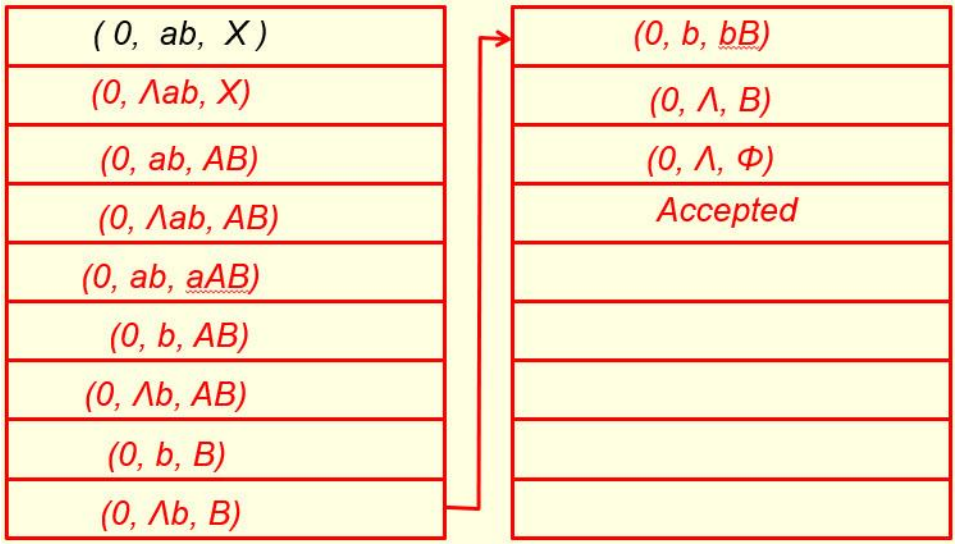
Given the following empty-stack PDA,



check if the strings *ba*, *ab, Λ and c* would be accepted by this PDA. Show your work in the following three tables, respectively. You need to create/use your own text boxes for your work in the tables.

Note that the key here is, whenever the top symbol of the stack is a nonterminal, if the next input symbol to be processed is not a Λ then you put a Λ in front of the un-consumed portion of athe input string so you can apply one of the seven instructions of the form "$\frac{\Lambda,?}{?}$" to replace the top symbol of the stack with a sequence of terminals and/or nonterminals.



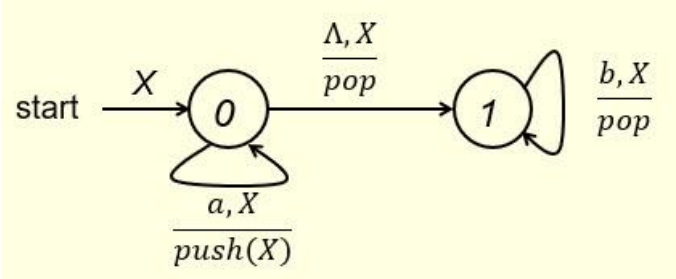


(0, Λ, X)
(0, Λ, AB)
(0, Λ, B)
(0, Λ, Φ)
Accepted

(0, c, X)
(0, Λc, X)
(0, c, c)
(0, Λ, Φ)
Accepted

6. (8 points)

Given the following empty-stack PDA, (1.2 points for each of the first five blanks)

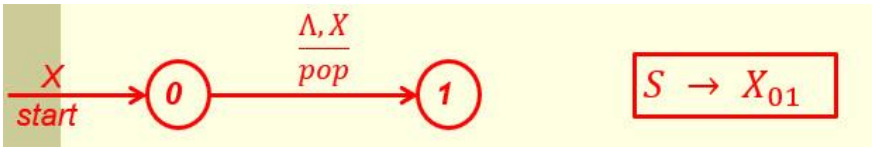


the language *L* accepted by the empty-stack PDA is $\{ a^n b^n \mid n \in N \}$. Note that Λ is an element of *L*.

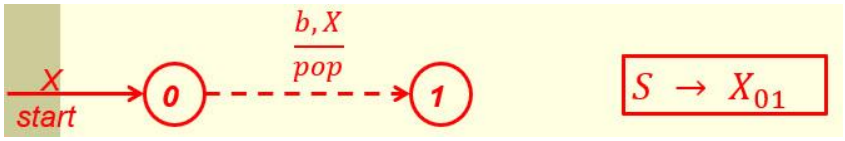
To transform the PDA to a C-F grammar, for each type of the PDA instructions, we need to construct the corresponding grammar productions. In the following, I list the PDA instructions on the left and you put the corresponding grammar productions in the blanks on the right. We start with Type 4.

Type 4:

The start state and the PDA instruction Λ,X/pop as shown below gives:



The start state and the PDA instruction b,X/pop as shown below gives the same production:



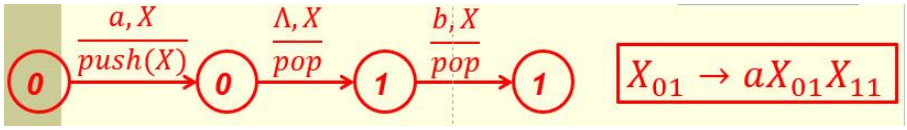
Type 1:

The PDA instruction Λ,X/pop by itself as shown below gives:

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

Type 3:

The PDA instruction a,X/push(X) together with Λ,X/pop and b,X/pop 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:

$S \rightarrow X_{01}$

$X_{01} \rightarrow a\, X_{01}\, X_{11}$

$X_{01} \rightarrow \Lambda$

$X_{11} \rightarrow b$

(0 points)

After simplification, we get

$S \rightarrow X$

$X \rightarrow aXb \mid \Lambda$

(2 points)

or

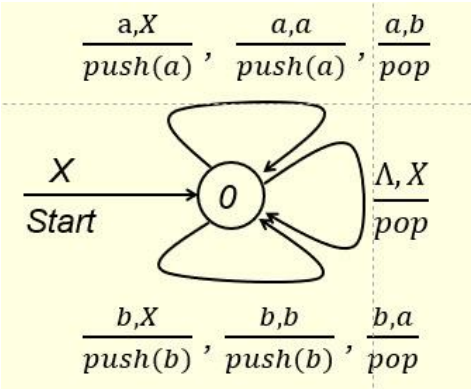
$S \rightarrow aSb \mid \Lambda$

(2 points)

(you only have to answer one case) and it is easy to see that this is a C-F grammar for *L*.

7. (7 points)

Given the following empty-stack PDA, (4 points for the first 8 blanks)

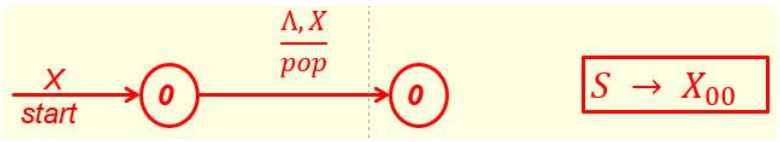


the language *L* accepted by the empty-stack PDA is the set of strings over the alphabet {a, b} such that each string has the same number of a's and b's.

To transform the PDA into a C-F grammar, for each type of the PDA instructions, we need to construct the corresponding grammar productions. In the following, put the corresponding grammar productions in the blanks on the right for the instructions shown on the left.

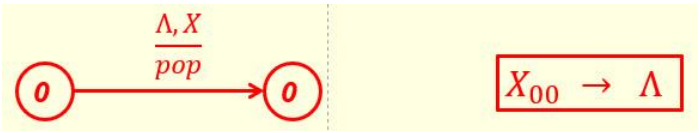
Type 4:

The start state and the PDA instruction Λ,X/pop as shown below gives:



Type 1:

The PDA instruction Λ,X/pop by itself as shown below gives:

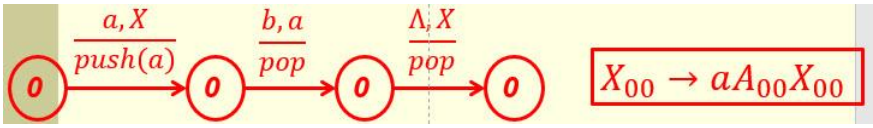


The PDA instruction a,b/pop by itself as shown below gives:

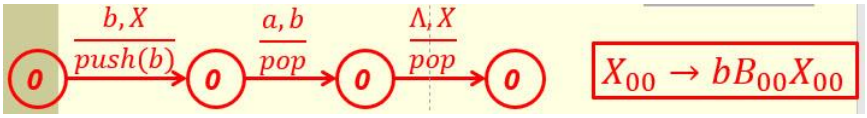
The PDA instruction b,a/pop by itself as shown below gives:

Type 3:

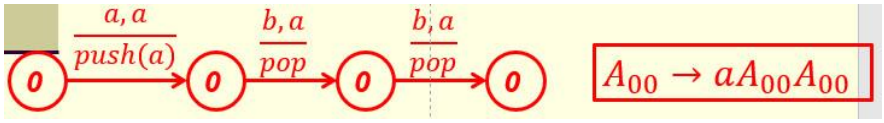
The PDA instruction a,X/push(a) together with the instructions b,a/pop and Λ,X/pop as shown below gives:



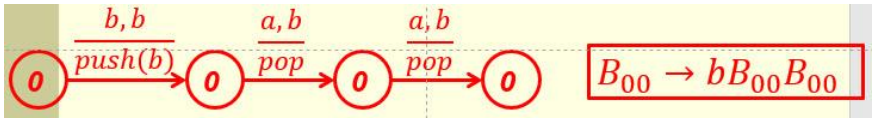
The PDA instruction b,X/push(b) together with the instructions a,b/pop and Λ,X/pop as shown below gives:



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



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



There are no Type 2 productions 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:

$S \rightarrow$	X_{00}		
$X_{00} \rightarrow$	Λ	$aA_{00}X_{00}$	$bB_{00}X_{00}$
$A_{00} \rightarrow$	b	$aA_{00}A_{00}$	
$B_{00} \rightarrow$	a	$bB_{00}B_{00}$	

(0 points)

After simplification, we get

$S \rightarrow$	Λ	aAS	bBS
$A \rightarrow$	b	aAA	
$B \rightarrow$	a	bBB	

(1 point for S; 1 point for A; 1 point for B)

and it is easy to see that this is a C-F grammar for *L*.