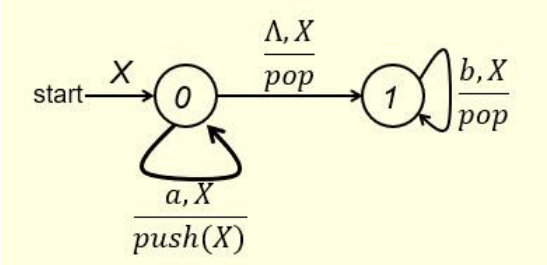
CS375 Homework Assignment 6 (40 points)

Due date: 11/06/2026

1. (9 points; one point for each of the first 5 blanks)
   Given the following empty-stack PDA,



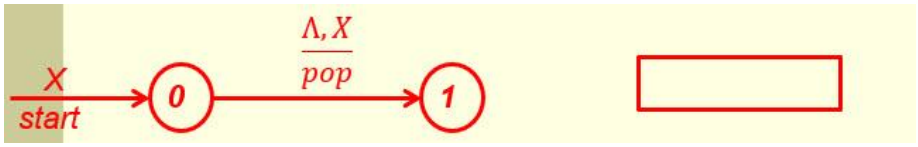
the language *L* accepted by the empty-stack PDA is $\{ a^n b^n \mid n \in N \}$. Note that Λ is an element of *L*.

To transform the PDA into 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:

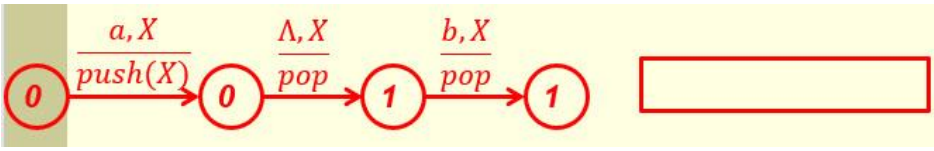
Type 1:

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:

$S \rightarrow$

$X_{01} \rightarrow$

$X_{01} \rightarrow$

$X_{11} \rightarrow$

(2 points)

After simplification, we get

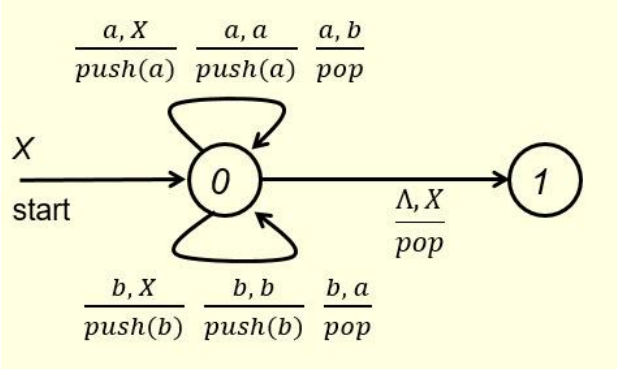
$S \rightarrow$

$X \rightarrow$

(2 points)

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

2. (9 points; 4 points for the first 8 blanks)
Given the following empty-stack PDA,

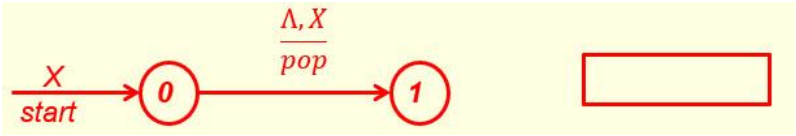


the language *L* accepted by the empty-stack PDA is the set of strings over the alphabet {a, b} such that each string has the same number of a's and b's.

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.

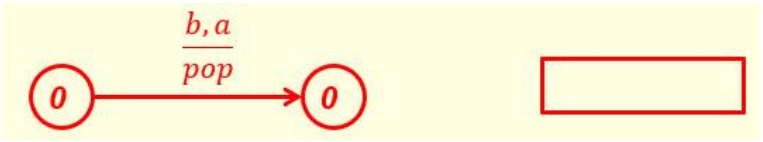
Type 4:
The start state and the PDA instruction Λ,X/pop as shown below gives:



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

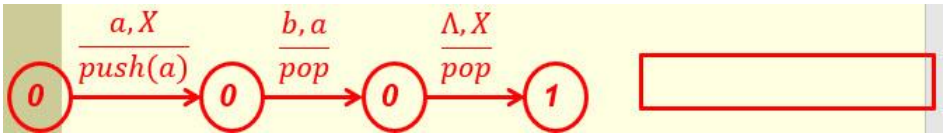
The PDA instruction a,b/pop by itself as shown below gives:

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

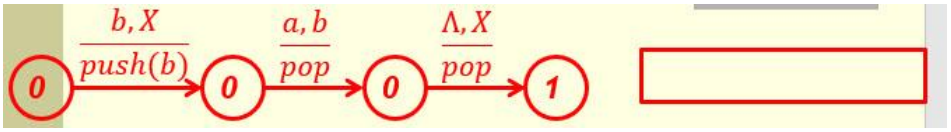


Type 3:

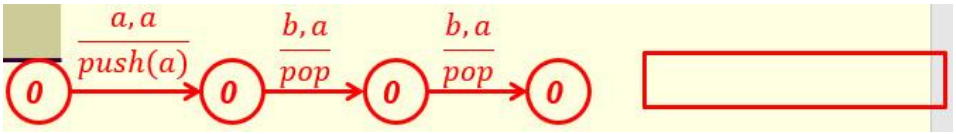
The PDA instruction a,X/push(a) together with the instructions b,a/pop and Λ,X/pop as shown below gives:



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

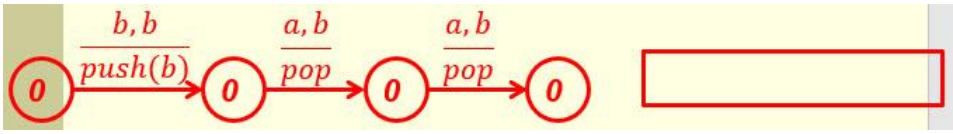


The PDA instruction a,a/push(a) together with the instruction b,a/pop (used twice) as shown below gives:



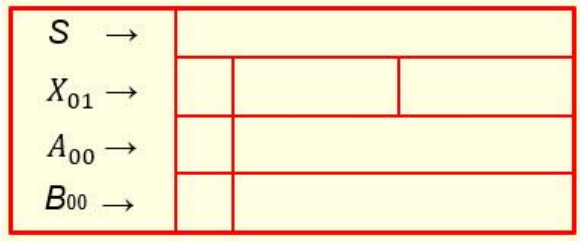
The PDA instruction b,b/push(b) together with the instruction a,b/pop (used twice) as

shown below gives:



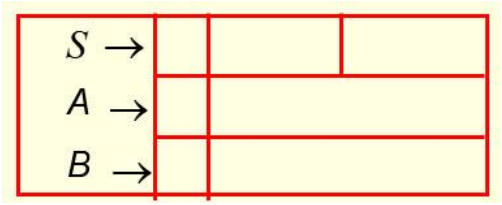
There are no Type 2 productions because there are no PDA instructions that perform nop stack operation.

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



(0 points)

After simplification, we get

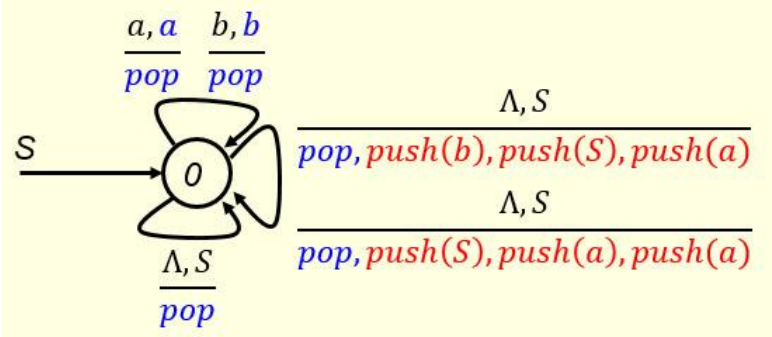


(3 points for S; 1 point for A; 1 point for B)

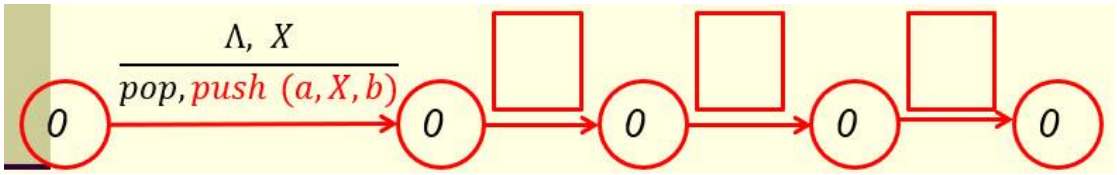
and it is easy to see that this is a C-F grammar for *L*.

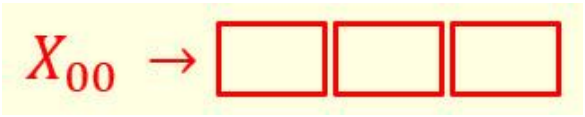
3. (6 points)

Given the context-free grammar {S → Λ ; S → aSb ; S → aaS }, we can convert it to an one-state empty-stack acceptance PDA as follows.

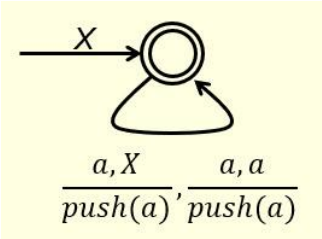


On the other hand, given such an one-state empty-stack acceptance PDA, we can also convert it to a CFG. In this case, we have one type-4 path, three type-1paths and two general type-3 paths. In the following, fill out the blanks in the general type-3 path for the case $\frac{\Lambda,S}{pop,push(b),push(S),push(a)}$ and the blanks in the corresponding CFG production. (6 points)



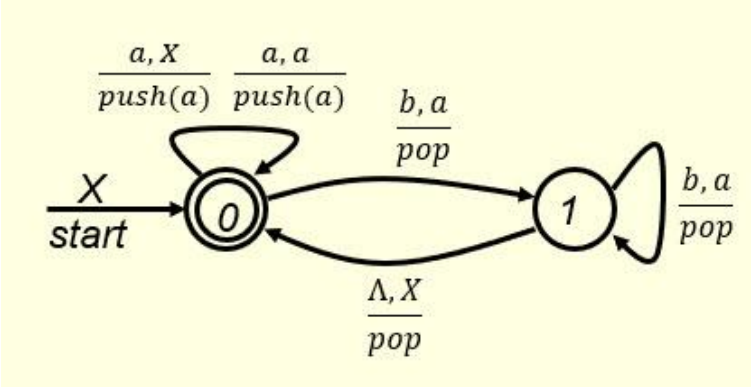


4. (1 point)
Final-state acceptance and empty-stack acceptance are equivalent only for NPDA's. They are not equivalent for DPDA's. For DPDA's the class of languages defined by final-state acceptance is bigger. In the following, use the given DPDA



to show that this is indeed the case by pointing out that the language L = { | n ϵ **N** } is accepted by the given (final-state) DPDA, but is not accepted by the DPDA when viewed as an empty-stack DPDA.

5. (7.5 points)
Given the following final-state DPDA,



and the following strings

Λ, aa, aaa, ab, aabb, aaabbb,

which of these strings are accepted by the given final-state DPDA? Put your answer in the following blank.

(3 points)

If the given final-state DPDA is considered as an empty-stack NPDA (state 0 is no longer a final state), then which of the given strings are accepted by the empty-stack DPDA? Put your answer in the following blank.

(1.5 points)

Now, consider the following two general questions. First, what is the language L_1 accepted by the given final-state DPDA? Put your answer in the following blank.

L_1 =

(2 points)

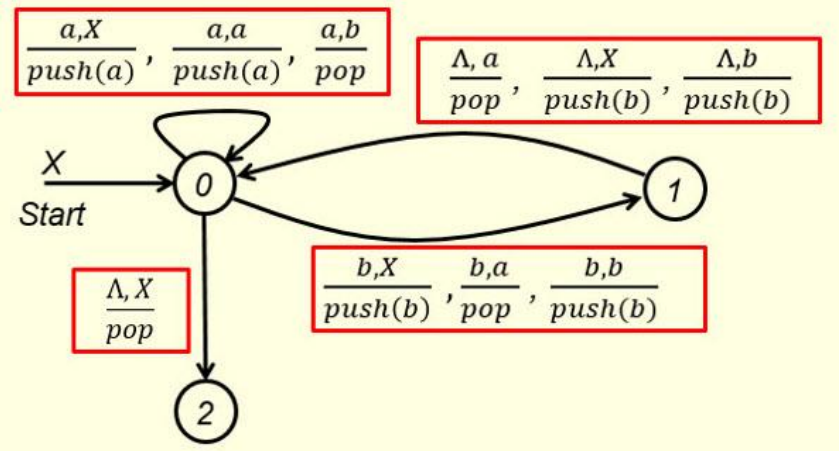
Second, what is the language L_2 accepted by this DPDA when viewed as an empty-stack DPDA? Put your answer in the following blank.

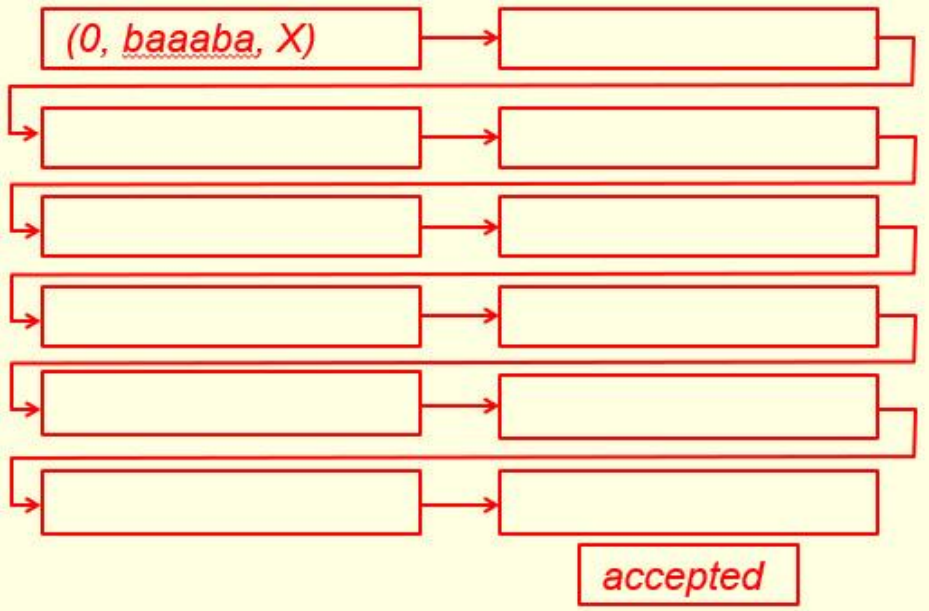
L_2 =

(1 point)

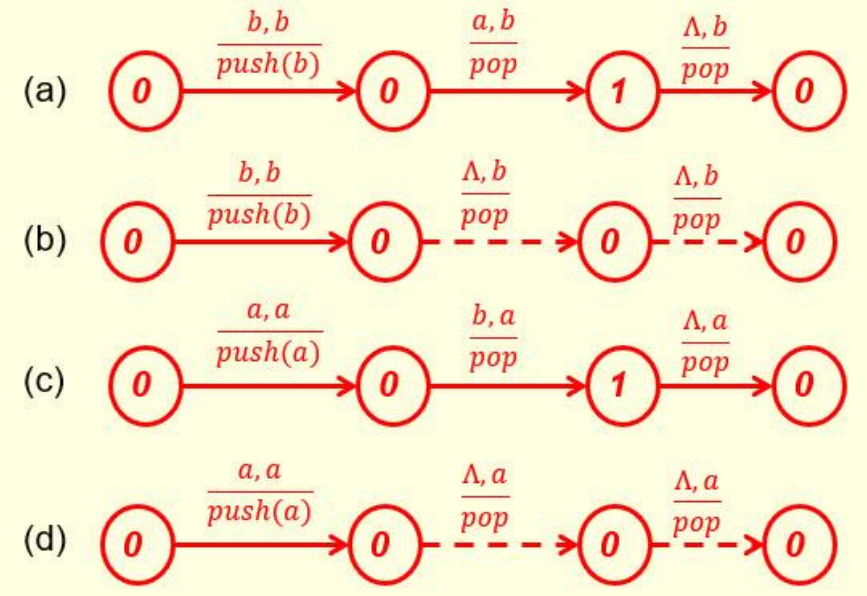
L_1 obviously is bigger than L_2.

6. (5.5 points)
The following empty-stack PDA accepts the language L = { w ϵ {a, b}* | $n_a(w) = 2n_b(w)$ } (assuming $\Lambda \in L$). In the following blanks show the execution of the string baaaba by this PDA.





7. (2 points)
The empty-stack PDA given in question #6 has one type 4, four type 1 and eight type 3 instructions. In the following four possible type 3 instructions, which one(s) are legitimate type 3?



Put your answer in the following blank.

- Solutions must be typed (word processed) and submitted both as a pdf file and a word file to Canvas before 23:59 on 11/06/2026.
- Don't forget to name your files as
  CS375_2026f_HW6_LastName.docx / CS375_2026f_HW6_LastName.pdf