Computer Science 2710, Spring 2016
Programming Assignment
Due: May 5, 2016 at 11:59 p.m. on ilearn
NO LATE ASSIGNMENTS ACCEPTED!

Write a program which reads in files containing descriptions of two-stack PDA's and allows the user to test them with input strings. Your program must allow the capability to process multiple strings and multiple PDAs without restarting the program. When you turn in your program, you must turn in a zip file on ilearn. This zip file will contain all your source code and instructions for compiling and running your program. In addition, you must include three test input files that are different from the test files shown below. The test two-stack PDA's must utilize both stacks in a nontrivial fashion. This means pushing and popping from both stacks, not necessarily on the same transitions.

Input file format:
line 1: input alphabet (elements of the alphabet are lower-case letters), where the letters are separated by spaces. Example: a b c d e

line 2: number of states in the two-stack PDA. States are assumed to be numbered 0,…,n-1, where n is the number of states.

line 3: list of final states in the two-stack PDA, separated by spaces. Example: 0 2

lines 4 to 4 + (n – 1): transitions for each state and input symbol, indicating the resulting state, what is to be popped from stack 1, what is to be pushed on stack 1, what is to be popped from stack 2, and what is to be pushed on stack 2. ! denotes the empty string. Line 4 contains state 0's transitions, line 5 contains state 1's transitions, etc.

Example: Suppose the following line 4.
2,a,abc,!,d   3,!,!,b,!   4,a,b,b,a   5,!,!,!,!   2,e,e,ef,!

This means that from state 0, if a is the input character, and a is popped from stack 1, and nothing is popped from stack 2, then abc is pushed onto stack 1 (first an a, then a b, then a c), and d is pushed onto stack 2, and the machine goes to state 2. If b is the input character, and nothing is popped from stack 1, and b is popped from stack 2, then nothing is pushed onto stack 1 and nothing is pushed onto stack 2, and the machine goes to state 3. The others can be explained similarly. When popping, if there are multiple characters, they must be popped in the order given. Remember that a stack is a LIFO (last in, first out) data structure. The general format:

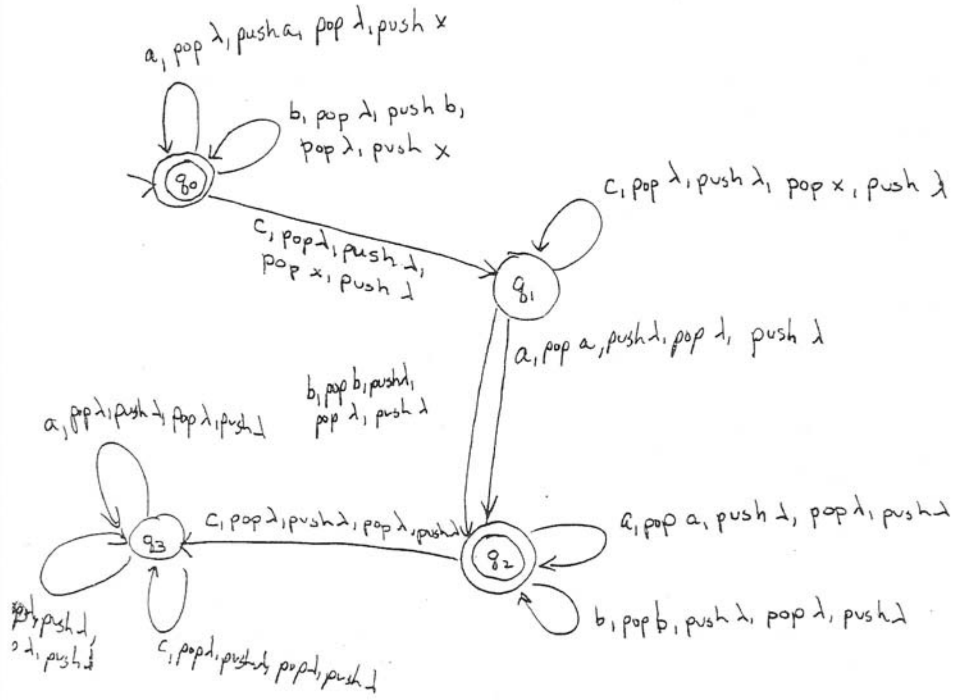
new state, what to pop from stack 1, what to push on stack 1, what to pop from stack 2, what to push on stack 2

Spaces are used to separate transitions, and commas are used to delimit the parts of a transition

After an input file has been read and its information stored in an appropriate data structure that you develop, you will then allow the user to run inputs through the two-stack PDA. Allow the user to do a fast run or a step-by-step run. Let the user know whether the string was accepted or rejected. Assume acceptance by final state and empty stacks. In the step-by-step run, show the user the current state and the contents of both stacks (i.e., a configuration of the two-stack PDA).

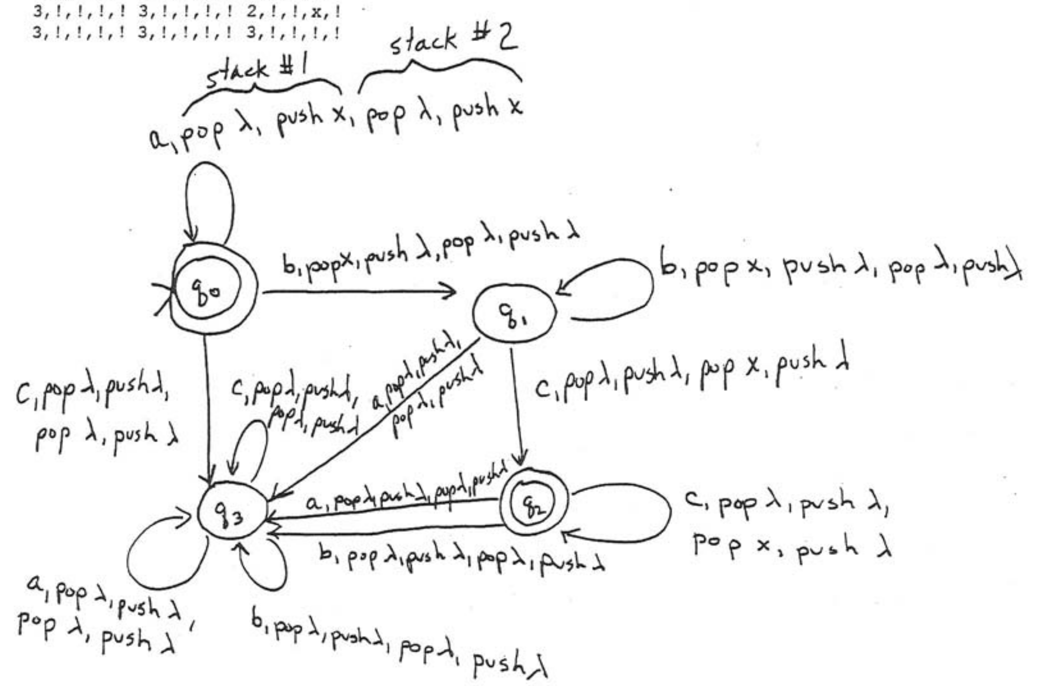
```
a b c
4
0 2
0,!,a,!,x 0,!,b,!,x 1,!,!,x,!
2,a,!,!,! 2,b,!,!,! 1,!,!,x,!
2,a,!,!,! 2,b,!,!,! 3,!,!,!,!
3,!,!,!,! 3,!,!,!,! 3,!,!,!,!
```

$$\{ wxw^R \mid |w| = |x|,\ w \in \{a,b\}^*,\ x \in \{c\}^* \}$$



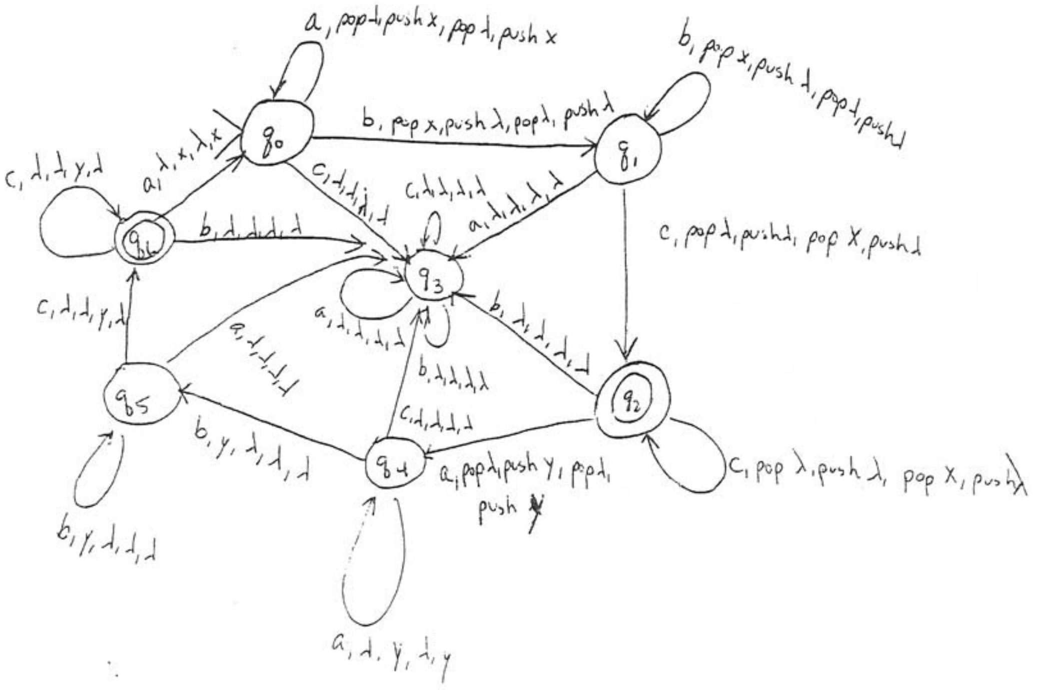
a, pop λ, push a, pop λ, push x
b, pop λ, push b, pop λ, push x
q0
c, pop λ, push λ, pop x, push λ
c, pop λ, push λ, pop x, push λ
q1
a, pop a, push λ, pop λ, push λ
b, pop b, push λ, pop λ, push λ
a, pop λ, push λ, pop λ, push λ
c, pop λ, push λ, pop λ, push λ
q3
q2
a, pop a, push λ, pop λ, push λ
b, pop b, push λ, pop λ, push λ
pop λ, push λ, pop λ, push λ
c, pop λ, push λ, pop λ, push λ

```
a b c
4
0 2
0,!,x,!,x 1,x,!,!,! 3,!,!,!,!
3,!,!,!,! 1,x,!,!,! 2,!,!,x,!
3,!,!,!,! 3,!,!,!,! 2,!,!,x,!
3,!,!,!,! 3,!,!,!,! 3,!,!,!,!
```


stack #1
stack #2
a, pop λ, push x, pop λ, push x
b, pop x, push λ, pop λ, push λ
b, pop x, push λ, pop λ, push λ
c, pop λ, push λ, pop λ, push λ
c, pop λ, push λ, pop λ, push λ
a, pop λ, push λ, pop λ, push λ
c, pop λ, push λ, pop x, push λ
a, pop λ, push λ, pop λ, push λ
c, pop λ, push λ, pop x, push λ
b, pop λ, push λ, pop λ, push λ
a, pop λ, push λ, pop λ, push λ
b, pop λ, push λ, pop λ, push λ
q0
q1
q2
q3

$\{ a^n b^n c^n \mid n \geq 0 \}$

```
a b c
7
2 6
0,!,x,!,x 1,x,!,!,! 3,!,!,!,!
3,!,!,!,! 1,x,!,!,! 2,!,!,x,!
4,!,y,!,y 3,!,!,!,! 2,!,!,x,!
3,!,!,!,! 3,!,!,!,! 3,!,!,!,!
4,!,y,!,y 5,y,!,!,! 3,!,!,!,!
3,!,!,!,! 5,y,!,!,! 6,!,!,y,!
0,!,x,!,x 3,!,!,!,! 6,!,!,y,!
```



$$\left(\{a^n b^n c^n \mid n \geq 1\}\right)^+$$