Hacettepe University

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

Name and Surname: Fatih Pehlivan

Number: 21946529

Course: Data Structre Lab BBM203

Experiment: Assignment3

Subject: Stacks

Date Due: 17/12/2021

Advisor: Ahmet Alkılınç

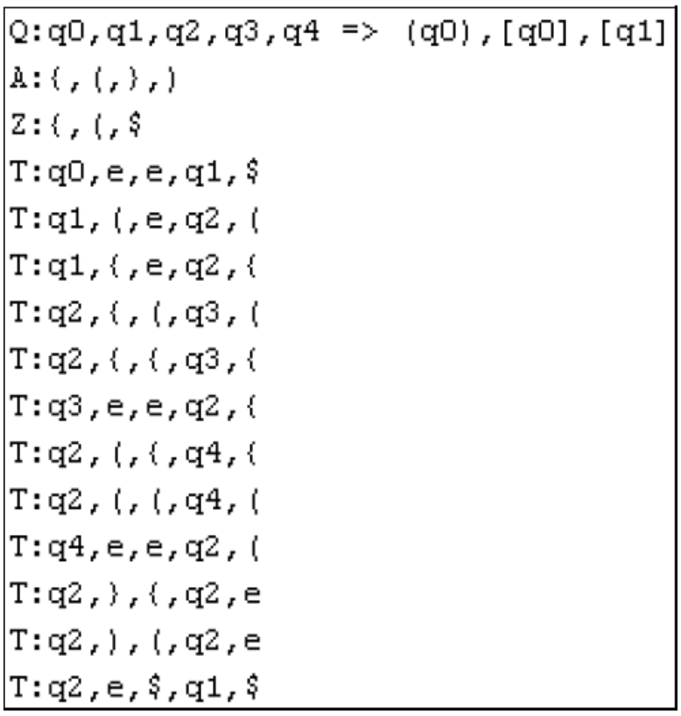
E-mail: b21946520@cs.hacettepe.edu.tr

Main Program: b21946529\src\Main.cpp

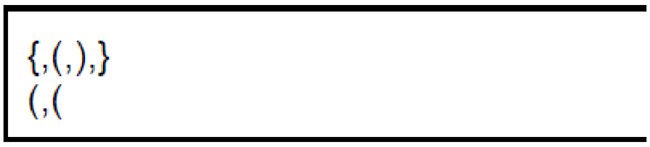
Software Using Documentation

Software Usage:

This program takes 3 arguments from the user. First is DPDA features, second is input strings and third is output file which will be printed our output there. An Example about inputs and the output.



```
Q:q0,q1,q2,q3,q4 =>  (q0),[q0],[q1]
A:{,(,},)
Z:{,(,$
T:q0,e,e,q1,$
T:q1,(,e,q2,(
T:q1,{,e,q2,{
T:q2,{,(,q3,(
T:q2,{,{,q3,{
T:q3,e,e,q2,{
T:q2,(,{,q4,{
T:q2,(,(,q4,(
T:q4,e,e,q2,(
T:q2,},{,q2,e
T:q2,),(,q2,e
T:q2,e,$,q1,$
```



```
{,(,),}
(,(
```

Input.txt

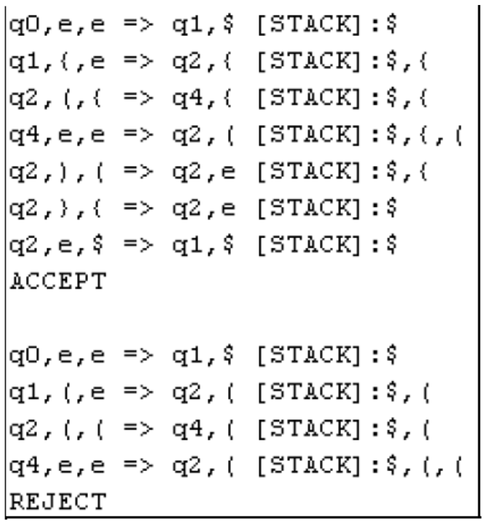
DPDA.txt

Q: Left side of “=>” is our states, right side of “=>” our initial and final states. Inıtial state must be written in brackets, final steps must be written square brackets.

A: is our alphabet

Z: is our stack Alphabet

T: is our steps.



```
q0,e,e => q1,$ [STACK]:$
q1,{,e => q2,{ [STACK]:$,{
q2,(,{ => q4,{ [STACK]:$,{
q4,e,e => q2,( [STACK]:$,{,(
q2,),( => q2,e [STACK]:$,{
q2,},{ => q2,e [STACK]:$
q2,e,$ => q1,$ [STACK]:$
ACCEPT

q0,e,e => q1,$ [STACK]:$
q1,(,e => q2,( [STACK]:$,(
q2,(,( => q4,( [STACK]:$,(
q4,e,e => q2,( [STACK]:$,(,(
REJECT
```

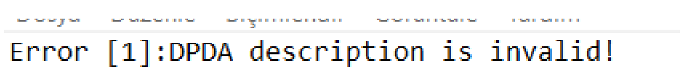
The output should be like that.

Error Message:

If there are states different than in Q’s left side. The program will print an error message. Initial stat, final states, and states which are in T cannot be different from our first definition.

The other error message if there is a word in T’s first element which is not included A, the program will print an error message. Or if there is a word in T’s second and fourth element which is not included Z, the program will print an error message.

The last is if you trying to pop an element from a stack that is empty. The program print before steps and after then prints an error message and then exits the program.



```
Error [1]:DPDA description is invalid!
```

Software Design Notes

Problem:

The problem is writing a program using DPDA. The program accepts a string belonging to a language or rejects a string not belonging to the language. The program firstly checks DPDA format is correct. Then take letters from the input file And apply rules according to the letter and DPDA rules. After that take the next letter. If there is more than one row in the input file, the program empties the stack and applies the above’s rules.

Solution:

I take inputs as arguments. I create some vectors. One includes states, the other includes A, other includes Z. I create some 2D vectors. One includes final states, the other includes T values which the first element is “e” the other includes other T values, the last one includes input files elements. I take the initial state as a string. First I make sure DPDA is true. I check it first. If it’s not true the program prints an error message and exits the program. I have a function which is a print stack. Apply the DPDA rules and every step I print the stack. If there is no element in the stack and the program is in the final state the program prints “ACCEPT” message. If not the program prints “REJECT” message.

Algorithm:

1-File Operations:

Read input and DPDA files.

Create output file. Everything I printed with “cout” goes to the file.

2-Applying DPDA Rules:

Create a stack from STL.

Check the DPDA file. If it is not true the program prints an error message.

Apply the rules.

Print stack in every step.

If there is an “e” in T’s 2nd element push T’s 4th element.

If there is an “e” in T’s 4th element, pop the stack’s last element.

If you trying to pop an element from an empty stack the program prints an error message.

If there are no more letters in the input file’s line and the stack is empty and the program is in the final state, the program prints “ACCEPT”

If there is no such that case the program prints “REJECT”.

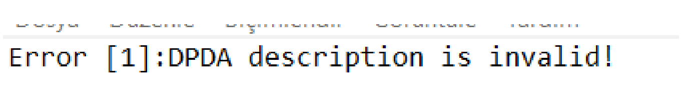
3-Closing Files:

Software Testing Notes

Bugs:

The input file is not checked if it’s true or false. It can be some bugs there.

If you trying to pop an element from an empty stack, the program prints an error message.



```
Error [1]:DPDA description is invalid!
```

References:

https://www.geeksforgeeks.org/print-stack-elements-from-bottom-to-top/

203 piazza page