HACETTEPE UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF ARTIFICIAL INTELLIGENCE ENGINEERING

BBM203 – DATA STRUCTURE

PROGRAMMING ASSIGNMENT II

Hacettepe University

Computer Science and Engineering Department

Name and Surname : Mehmet Berat Ersarı

Identity Number : 21992943

Course : BBM 203- Data Structure Lab.

Experiment : Assignment 3

Subject : Stack

Data Due : 17.12.2021

2. Software Using Documentation

2.1. Software Usage

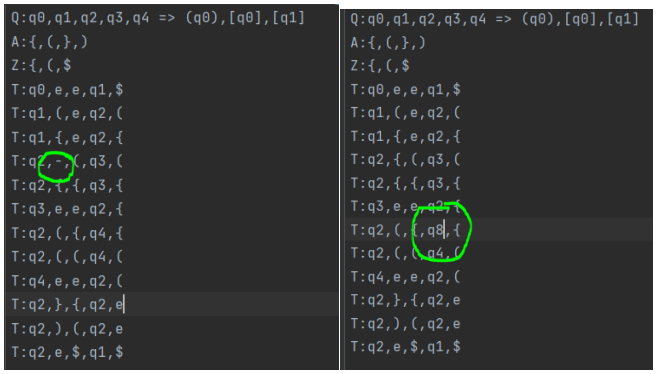
This software's algortihm is like dpda systems. It takes dpda file and input file of the dpda. Then, program makes the all operations, adding an element to stack. Also, prints every step while doing these operations. After all operations, checkes whether the stack is empty or not. Lastly, if the stack is empty, it prints "ACCEPTED". Else, it prints "REJECTED".

2.2 Error Messages

Error messages are feedback to users. A detailed table of errors is essentiaL A row of this table is as follows.

Error [1]:DPDA description is invalid!

If there is an statement which is not defined before in the dpda file. This will give error. Check the picture to understand better.



There is an "-" in transtisions. However, there is no element with matching this in A A. Also, there is an statement which is q8. But, it is not defined before. These conditions will give you error. If you dont get any error, write the input file correctly.

3.Software Design Notes

3.1. Desctription of the program

3.1.1. Problem

There are some transtions rules and operations. We should do all operations while using transtions rules. Then, prints the every step and result.

3.1.2.Solution

I solved this problem with the stack and dpda data. I created an stack in the main program and with the vectors I stored dpda and input data. For each given input line, I made all operations while printing every steps. Maybe my solution is a bit slow for given very big input. My solution can be more quick. If I had enough time, I can do it more efficient.

3.4. Algorithm

1. Make initialisation.

1.1. Open the input: 'dpda.txt' and 'dpda_input.txt'

Output: 'dpda_output.txt'(It is taken with arguments)

2. For every input in 'dpda.txt'

2.1. Read the dpda and dpda-input data and split it

2.1.1 Push the input which you splitted in vector separately.

2.2 Then check whether the dpda file is valid or not.

2.2.1 If it is not valid, print an error.

2.3 Create a stack.

2.4. Create dpda_output.txt

2.5. Apply all transtion rules in a while loop

2.5.1 Start with initial state

2.5.2 Apply transition rule while reading the inputs sequantially. And write it on output.txt Push and pop the elements.

2.5.3 After all these, check the last state is final or not and empty or not. If it is final and empty, check there is empty transtion rule. Apply it and check it is final state. If it is final state, write "ACCEPTED" on output.txt, else write "REJECTED" on output.txt

2.3.2 Print ACCEPT if stack is empty and the last state is final state else REJECT

3. Close files

3.5. Special Design Properties

I could do my program more efficient. For example, I made one more searching while applying transtion rules. All search can be in one step.

Software Testing Notes *:*

My program is completely sucessful. I tested my program in many aspects.

4. SOFTWARE TESTING NOTES :

4.1. Bugs and Software Reliability

I think there is no bug If you give correct dpda file.

REFERENCES

https://www.tutorialspoint.com/cplusplus/cpp_files_streams.html