HACETTEPE UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

BBM 203 SOFTWARE LABORATORY
ASSIGNMENT 3 – STACKS
December 17, 2021

Advisor: Ahmet ALKILINÇ
Email: b21946198@cs.hacettepe.edu.tr

Umut GÜNGÖR 21946198

This software runs when it takes two input txt files with command line arguments. Both input files contain data that will be process in software. That processed data is going to create and write and output txt file unless there is an error.

There can be errors when software runs. Such as wrong inputs are given. Example for an error can be like that:

<invalid input error>: <Error [1]:DPDA description is invalid!>

In this assignment, we are asked to implement a Deterministic Pushdown Automato with using stacks. Also, since we read inputs from file another goal of this assignment is taught file input/output operations.

First of all, we should read inputs from txt files correctly. After read them all, I prefer store that data in the vectors. Because, this data's size not fixed so, I have used dynamic arrays. I write my own <Stack> class in order to use my functions easily. For instance, when I want to access my stack content I can use one function and I can access whole content of the stack as well. Moreover, my <Stack> class contains typical stack methods. Such as <pop>, <top>, <push> etc. Secondly, I write the output to the output file unless there is not an error. If there is an error like not invalid input according to the stack alphabet or states program writes an error message to the output file and terminates itself. Otherwise, program reads inputs and process them according to the Deterministic Pushdown Automata's rules. And write this data into the output file.

INPUTS	PROGRAMS	OUTPUT
dpda1.txt	Main.cpp	output1.txt
input1.txt	AutomatonStack.hpp	
	AutomatonStack.cpp	

I used stack data structure and vectors mainly. Used stack for check Deterministic Pushdown Automata's input alphabet correctly. Used vectors to store inputs dynamically.

My algorithm to solve this problem has couple of parts. One of them is, store inputs in dynamic arrays and when I need to access them iterate over these arrays and process them. I used stacks because Deterministic Pushdown Automata requires it and also the stack operations complexity are O(1). So, that reduces time and space complexity of the software.

This software may contain some possible bugs. One of them is of course invalid character in input files. The input files should be appropriate. Another error is the transition function's members are not included the given alphabets. For example, a given state may not be in the given states. In this situation, program can catch this and write an error message to the output file and finish itself.