THEORY OF COMPUTATION
ASSIGNMENT

Q2.1 Given a CFG convert it to PDA

BASIC STRUCTURE

1. **CFG_PDA.h** is the header file that contains all the variables and the function prototypes
2. **CFG_PDA.cpp** is where the implementation of the conversion algorithm takes place.
3. **cfg_pda_main.cpp** is the driver function where the file containing the grammar is read and the corresponding grammar is converted to PDA using the functions in CFG_PDA.cpp

VARIABLES

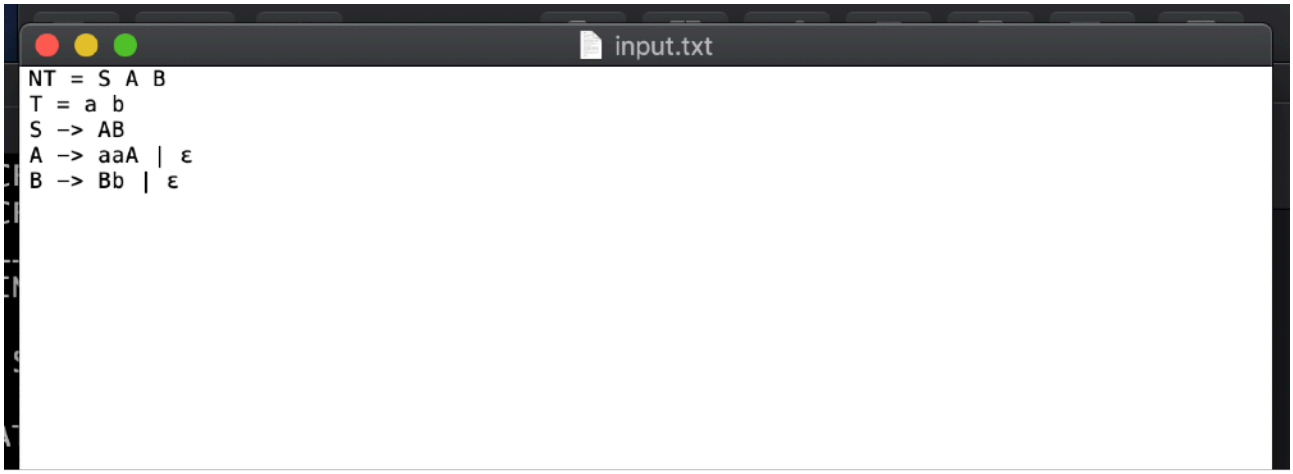
TYPE	NAME	DESCRIPTION
vector<string>	Non_Term_States	Vector to store the Non Terminal States
vector<string>	Term_states	Vector to store the Terminal States
vector<pair<string,vector<string>>>	Grammer	Grammer is a vector in which each entry is a pair (LHS and RHS) where LHS is symbol (string) and RHS is productions (vector) in one of the production rules
ifstream	input	Input stream object to operate on files

FUNCTIONS

1. **void FileRead() -** Reads a file input.txt which contains the grammar
   - Checks error in opening file
   - Takes each line of the input file
     - Line 1 contains Non Terminals (eg NT = S A B)
     - Line 2 contains Terminals (eg T = a b)
   - Consecutive lines contain the production rules, having a non terminal symbol on the left (LHS) and other symbols on the right (RHS).
   - Use make_pair function to put the LHS of production rule in string and RHS to vector<string> of '**vector<pair<string,vector<string>>>'**

2. **CFG_to_PDA()** - Displays the all the symbols and production rules using the vectors and corresponding PDA template of the above CFG

INPUT



OUTPUT