**NAME –** DEVANGI PURKAYASTHA

**REGISTRATION NUMBER –** RA1811003010689

**SECTION-** D2

LAB EXP:7 CONSTRUCTION OF SHIFT-REDUCE PARSER DATE: 12 March 2021

**AIM : To construct a Shift-Reduce Parser**

**CODE:**

gram = {

    "E":["2E2","3E3","4"]

}

starting\_terminal = "E"

inp = "2324232$"

"""

# example 2

gram = {

    "S":["S+S","S\*S","i"]

}

starting\_terminal = "S"

inp = "i+i\*i"

"""

stack = "$"

print(f'{"Stack": <15}'+"|"+f'{"Input Buffer": <15}'+"|"+f'Parsing Action')

print(f'{"-":-<50}')

while True:

    action = True

    i = 0

    while i<len(gram[starting\_terminal]):

        if gram[starting\_terminal][i] in stack:

            stack = stack.replace(gram[starting\_terminal][i],starting\_terminal)

            print(f'{stack: <15}'+"|"+f'{inp: <15}'+"|"+f'Reduce S->{gram[starting\_terminal][i]}')

            i=-1

            action = False

        i+=1

    if len(inp)>1:

        stack+=inp[0]

        inp=inp[1:]

        print(f'{stack: <15}'+"|"+f'{inp: <15}'+"|"+f'Shift')

        action = False

    if inp == "$" and stack == ("$"+starting\_terminal):

        print(f'{stack: <15}'+"|"+f'{inp: <15}'+"|"+f'Accepted')

        break

    if action:

        print(f'{stack: <15}'+"|"+f'{inp: <15}'+"|"+f'Rejected')

        break

**ALGORITHM-**

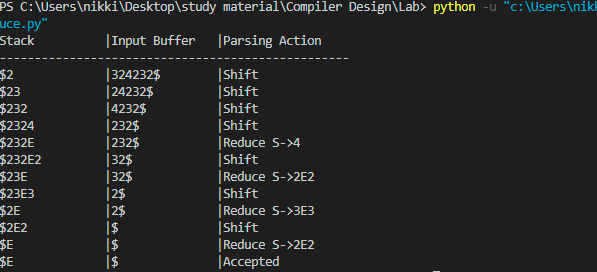
**Shift Reduce parser** attempts for the construction of parse in a similar manner as done in bottom up parsing i.e. the parse tree is constructed from leaves(bottom) to the root(up). A more general form of shift reduce parser is LR parser.   
This parser requires some data structures i.e. 

* A input buffer for storing the input string.
* A stack for storing and accessing the production rules.

**Basic Operations –** 

* **Shift:**This involves moving of symbols from input buffer onto the stack.
* **Reduce:**If the handle appears on top of the stack then, its reduction by using appropriate production rule is done i.e. RHS of production rule is popped out of stack and LHS of production rule is pushed onto the stack.
* **Accept:**If only start symbol is present in the stack and the input buffer is empty then, the parsing action is called accept. When accept action is obtained, it is means successful parsing is done.
* **Error:**This is the situation in which the parser can neither perform shift action nor reduce action and not even accept action.

**OUTPUT** –



**RESULT –**

The given program has been successfully executed.