ACD Lab ASSIGNMENT 2

Name : Aahan Singh Charak

Registration Number :189301024

Section : CSE A

Roll no : 5

1 . Write a program to design a PDA to check any string over a,b where (a^n b^n ; n>0), also show all transitions in output.

Sol :

Code : (python)

#question for cfg for a^nb^n

stack=['\0']

#the transition functions for the pda

#transitions are like this curstate:('input symbol','stack top','operation','transition to which state')

transitions={

    0:[('a','\0','push',0),('a','a','push',0),('b','\0','none',3),('b','a','pop',1)],

    1:[('a','\0','none',3),('a','a','none',3),('b','a','pop',1),('\0','\0','none',2),('b','\0','none',3)],

}

pattern=''.center(20,'\*')

curState=0

def alterStack(operation,char):

    global stack

    if operation=='push':

        stack.append(char)

    elif operation=='pop':

        stack.pop()

    else:

        pass

def makeTransition(char):

    global curState

    print(pattern)

    print('Current symbol is :')

    if char!='\0':

        print(char)

    else:

        print('String terminal')

    print(pattern)

    print('Cur state {}'.format(curState))

    print(pattern)

    stackTop=stack.pop()

    print('Stack top is :')

    if stackTop!='\0':

        print(stackTop)

    else:

        print('Null')

    print(pattern)

    stack.append(stackTop)

    if curState==3:

        return

    elif curState<2:

        for tup in transitions[curState]:

            if tup[0]==char and tup[1]==stackTop:

                alterStack(tup[2],char)

                curState=tup[3]

                break

    elif curState==2 and char !='\0':

        curState=3

    print('After transition state is {}'.format(curState))

    print('\n\n')

testString=input('Enter the string (must have either a or b) : ')

testString+='\0'

print(testString)

for char in testString:

    if char in ['a','b','\0']:

        makeTransition(char)

    else:

        print('Sorry unwanted symbol inside the input string')

        break

if curState==2:

    print('String accepted by the pda')

else:

    print('String not accepted')

Ouput:

Text

Description automatically generated

A screenshot of a computer

Description automatically generated

2. Write a program to design a PDA to check any string for even size palindrome. also show all transitions in output.

Sol:

Code in python:

import math

#question for cfg for a^nb^n

stack=['\0']

#the transition functions for the pda

#transitions are like this curstate:('input symbol','stack top','operation','transition to which state')

currentState=0

pattern=''.center(20,'\*')

def automatize(string):

    global currentState

    for index,char in enumerate(string):

        prevState=currentState

        print('Previous state is {}'.format(prevState))

        print(pattern)

        print('Input symbol is {}'.format(char))

        print(pattern)

        if char in ['a','b']:

            if index+1<=len(string)//2:

                print('Pushing into the stack')

                print(pattern)

                stack.append(char)

            else:

                lastEle=stack.pop()

                if char==lastEle:

                    print('Top of the stack equal to the sybmol {} . So performing pop operation'.format(char))

                    print(pattern)

                    currentState=2

                else:

                    print('Top of the stack not  equal to the sybmol {} . So quitting'.format(char))

                    print(pattern)

                    currentState=3

                    break

        else:

            print('Invalid symbols in the string')

            break

        print('Current state is {}\n\n'.format(currentState))

        print(pattern)

    if currentState==2:

        print('String is a palindrome')

    else:

        print('String is not a palindrome')

string=input('Enter the required string which is to be tested for a palindrome : ')

if(len(string)%2==0):

    print('Even Palindrome\n')

    automatize(string)

else:

    print('Odd palindrome\n')

    newstring=string[:math.floor(len(string)/2)]+string[math.floor(len(string)/2)+1:]

    automatize(newstring)

Ouput:

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence