Prefix Expression Evaluation Using Python

Venkatesh Subramaniam

April 22, 2017

1 Prefix Expression

Problem: You are given a prefix expression. Write a program to evaluate it. Your program should accept as its first argument a path to a filename. The file contains one prefix expression per line.

INPUT SAMPLE:

1 * + 2 3 4

Your program has to read this and insert it into any data structure you like. Traverse that data structure and evaluate the prefix expression. Each token is delimited by a whitespace. You may assume that the only valid operators appearing in test data are '+','*'and'/'(floating-point division). Please include unit tests that demonstrate how your code works.

Please zip the contents of your solution named: `prefix-{!lastname}.zip`

OUTPUT SAMPLE:

Print to stdout, the output of the prefix expression, one per line. E.g.

1 20

Constraints: The evaluation result will always be an integer >= 0. The number of the test cases is <= 40.

```
In [6]: #################################Code###########################


        ## Defining the function prefix
        def prefix(line):

            ## Split the elements passed into the function separated by whitespace
            l ="".join(line.rstrip())
            strArr = l.split(" ")

            ## Initializing symbols and numbers list
            symbols = []
            numbers = []

            ## Iterating the values of i from 0 to (length of String passed - 1)
            for i in range(0, len(strArr)):

                ## Check if the element is an integer both positive and negative
                ## Conditions strip - sign if is present and check if the element i
                if(strArr[i].lstrip("-").isdigit()):
```

```
        ## Add the number as the last and latest element in numbers lis
        numbers.append(strArr[i])

        ## Check if the previous element is an integer both positive an
        ## Conditions strip - sign if is present and check if the eleme
        if( strArr[i-1].lstrip("-").isdigit()):

            ## Check for the condition if the length of the number is n
            while( len(numbers) != 1 ):

                ## Take the last and latest value with the help of pop
                secondVal = float(numbers.pop())
                firstVal = float(numbers.pop())

                ## Intialize the calculateVal
                calculateVal = 0

                ## Take the latest and last value from the symbols list
                symbol = symbols.pop()

                ## Check if the symbol is plus and perform the addition
                if( symbol == '+' ):
                    calculateVal = firstVal + secondVal

                ## Check if the symbol is cross and perform the multipl
                elif( symbol == '*' ):
                    calculateVal = firstVal * secondVal

                ## Check if the symbol is divide and perform the divisi
                elif( symbol == '/' ):
                    calculateVal = float(secondVal / firstVal)

                ## Add the calculated value to numbers list(add it as l
                numbers.append(calculateVal)

    ## If the number is not numeric, it passes through else
    else:

        ## Add the symbol to symbols list as last and latest element
        symbols.append(strArr[i])

## Add the final result as the last and latest element in the numbers l
result = numbers.pop()

## Contraint is checked if the result is greater than or equal to 0 and
if (float(result) >= 0):
    print(result)
    ## If the result is less than zero, then "The result is less than zero'
    else:
        print("The result is less than zero")


## Intializing i = 1
i = 1

## Manually asking the user for input
filename = input('Enter The File Name (Give it in quotes if using python 2)

## Read the file
for line in open(filename,"r").readlines():

    ##If it is the first line, print the number of lines to follow
    if i == 1:
        print (line)
        i += 1

    ## Pass the lines from second to the function prefix one after other
    elif i <= 40:
        prefix(line)
        i += 1
```

```
Enter The File Name (Give it in quotes if using python 2) : pretest.txt
10

30.0
45.0
8.0
2.0
The result is less than zero
4.0
5.0
4.5
11.0
8.0
```

Input test case

```
In [2]: import pandas

        text = pandas.read_csv("pretest.txt")
        print (text)
```

```
            10
0    * + 2 4 5
```

```
1    * 3 * 3 5
2   + 3 + -2 7
3    / 2 * 2 2
4  + 7 * 4 -3
5   / / 2 8 16
6    * 3 / 3 5
7    * / 2 3 3
8    + 2 * 3 3
9  + 2 * -2 -3
```

```
In [ ]:
```