Data Structure and Algorithms

Session-8

Dr. Subhra Rani Patra
SCOPE, VIT Chennai

Evaluation of prefix using stack

Scan the prefix expression from R to L

for each char in prefix expression

do

if operand is there, push it onto the stack

else if operator is there, pop 2 elements.

OP1 = top element

OP2 = next to top of the stack.

result = <OP1><Operator><OP2>

push the result into the stack

return stack[top]

Evaluation of prefix using stack

a+b*c-d/e^f

After prefix conversion

-+a*bc/d^ef

-+2*34/16^23

Lets suppose a=2,b=3,c=4,d=16,e=2,f=3

<op1><operator><op2>

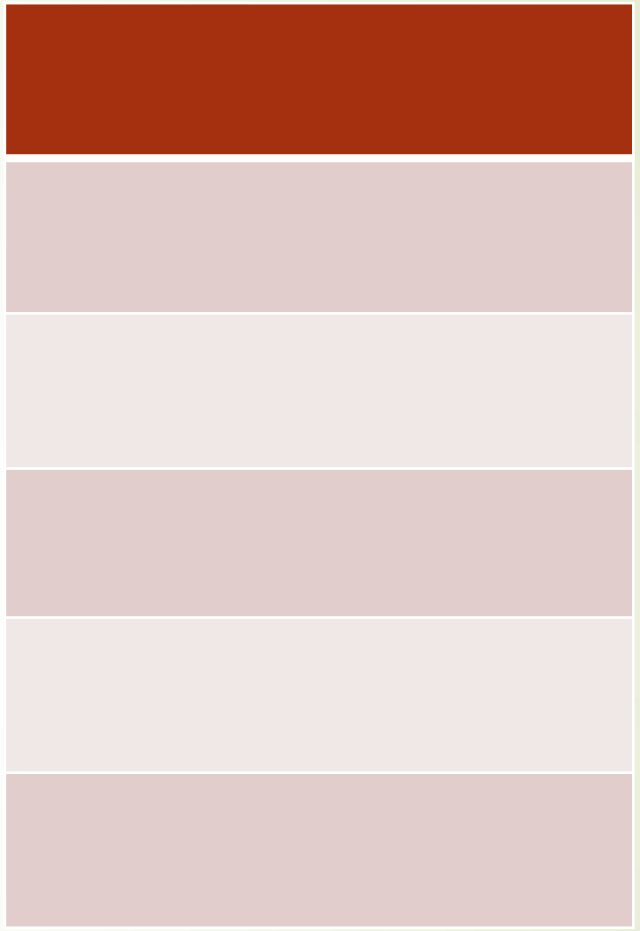
Op1=2

Op2=3

2^3=8

Push 8 into the stack

<op1><operator><op2>



Stack

Evaluation of postfix using stack

Begin
for each character in postfix expression
do
if operand is there, push it onto the stack
else if operator is there, pop 2 element

OP1 = top element

OP2 = next to top of the stack

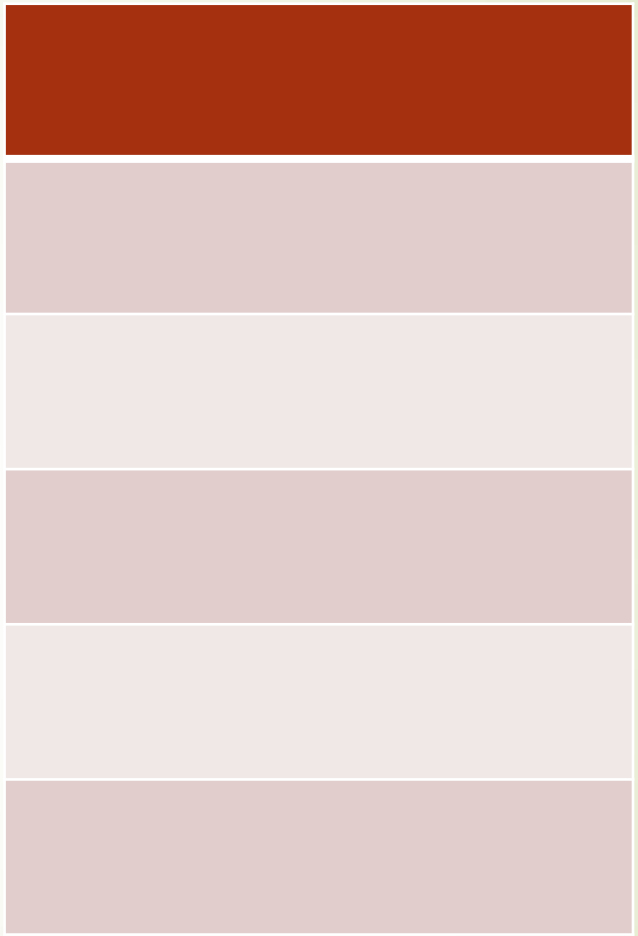
result = (OP2)(operator)(OP1)

push result on to the stack

return element from Stack[top]

Evaluation of postfix using stack

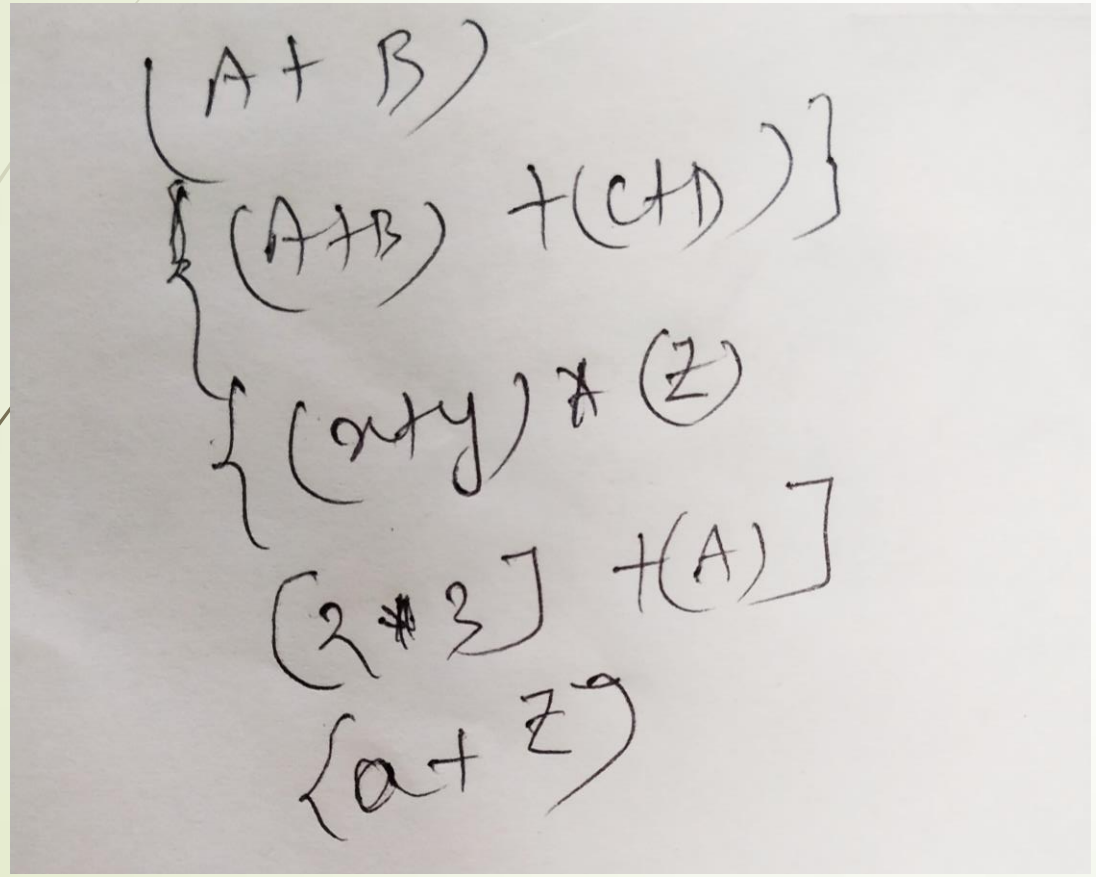
a+b*c-d/e^f
After postprefix conversion
abc*+dcf^/-
234*+1623^/-
Lets suppose a=2,b=3,c=4,d=16,e=2,f=3

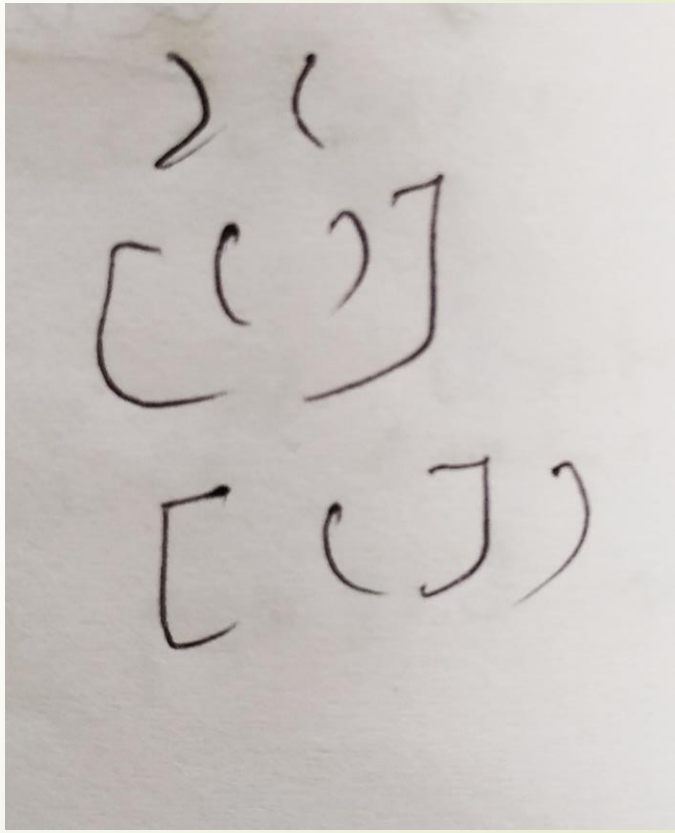


<op2><operator><op1>
Op1=4
Op2=3
Result=3*4=12

Push 12 into the stack
<op2><operator><op1>

Balancing parenthesis





Balancing parenthesis

Algorithm

1) Declare a character stack S.
2) Now traverse the expression string expression.

a) If the current character is a starting bracket (‘(‘ or ‘{‘ or ‘[‘) then push it to stack.
b) If the current character is a closing bracket (‘)’ or ‘}’ or ‘]’) then pop from stack and if the popped character is the matching starting bracket then fine else parenthesis are not balanced.

3) After complete traversal, if there is some starting bracket left in stack then “not balanced”

Example

Let's take an example

String = "{ ( ) } [ ]"

So, we declare a stack S and traverse the String.

At i = 0, String[0] = '{' : Push '{'

At i = 1, String[1] = '(' : Push '('

String = “{ ( ) } [ ]”

Stack

(← Top
{

At i = 2, String[2] = ‘)’, now, we pop one element from the stack. This closing bracket matches the opening bracket we popped.

Stack

{ ← Top

At i = 3, Stack[3] = ‘}’, now, we pop one element from the stack. This closing bracket matches the opening bracket we popped.

String = “{ ( ) } [ ]”

At i = 4, String[4] = ‘[‘ : Push ‘[‘

Stack

[← Top

At i = 5, String[5] = ‘]’, Now, we pop one element from the stack. This closing bracket matches the opening bracket we popped.

Now, we have traversed through the entire string and the Stack S is empty. Therefore, the string is balanced.

Another Example

String = “{ (}) [ ]”

At i = 0, String[0] = ‘{‘ : Push ‘{‘

At i = 1, String[1] = ‘(‘ : Push ‘(‘

Stack

(← Top

{

At i = 2, String[2] = ‘}’, this closing bracket does not match the opening bracket at the top. This means, the string is unbalanced.


Thank
you