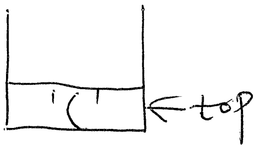
eXample of converting an infix to postfix with parens.

input: (A - B) * C

initially, stack is empty and output is "", empty string.
i=0, pointing to the first item of input infix.

while 1st iteration:

Current item is '(', do else if. push it onto the stack.

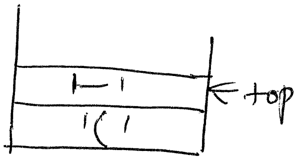


while 2nd iteration:

Current item is 'A', an operand. write it to the postfix
So output = "A".

while 3rd iteration:

current item is '-', an operator, do else,
while condition top item on stack has precedence 0, } False
current item has precedence 1,
skip while body. push ('-'),



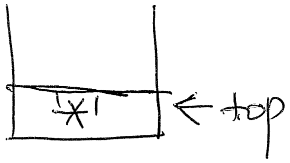
while 4th iter:
current is 'B', operand, write into output postfix:
output = "AB"

while 5th iter:
Current item is ')', do else if
- while top item is not '('. pop() and send out to output; = "AB-"
- pop() to discard '('

empty stack

while 6th iter:
current item is '*' operator, do else, empty stack (skip while)
push ('*')

After push() : Stack:



While 7th iteration:

Current item 'C' is operand, send to output postfix:

so output = "AB-C"

While loop drops out,
Then pop and write all operators on stack into postfix:

output = "AB-C*"

example of convert infix into postfix expression.

input : A - B * C + D / E

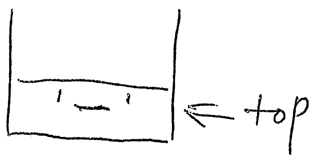
initially stack is empty and output is "", empty string.
i = 0 point to the first item of input Infix expression.

while 1st iteration :

A is an operand ; → write to Postfix , so → output = "A"

while 2nd iteration :

'-' is the current item, do else branch ;
push ('-')

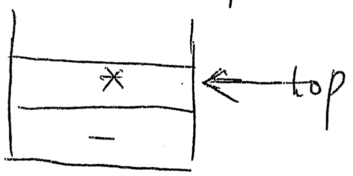


while 3rd iteration :

B → write to postfix so → output = "AB"

while 4th iteration :

* → operator → do else branch,
but top item in stack '-' has smaller precedence
So ; skip while in else ; push ('*')

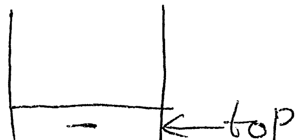


while 5th iteration :

C is operand → write to postfix → output = "ABC"

while 6th iteration :

"+" → Operator → do else branch ;
item on top of stack '*', has greater precedence
pop( ) → '*', write '*' to postfix → output = "ABC*"

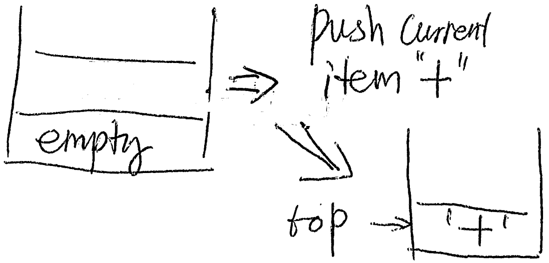


continue with iteration 6; In else branch

now, item on top of stack '–' has precedence greater than current item "+"; → by refering to precedence Lookup table

pop again; → '–', write into postfix →

output = "ABC*–"



while 7th iteration

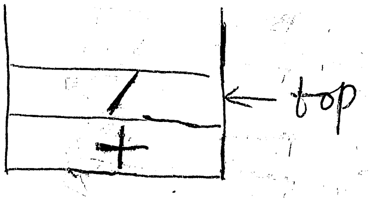
D → operand → output to postfix → output = "ABC*–D"

while 8th iteration:

'/' → operator → go else branch ⇒ 'item on stack top is '+', has smaller precedence;

So skip while body

push ('/');



while 9th iteration:

'E' → operand → write it to postfix → output = "ABC*–DE"

After outer while Loop drops out:

pop & write all operators in stack to postfix:

output = "ABC*–DE/+"