CSE 2215:
Data Structures and Algorithms-I

Stacks

Queues

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU

Application of Stacks:
Convert Infix to Postfix Expression

- Infix expression: The expression of the form a op b. When an operator is in-between every pair of operands.
  - Example: 5 + 8

- Postfix expression: The expression of the form a b op. When an operator is followed for every pair of operands.
  - Example: 5 8 +

- Why postfix representation of the expression?
  - Infix expressions are readable and solvable by humans. We can easily distinguish the precedence of operators, associativity of operators, and also know the uses of parenthesis.
  - The computer cannot differentiate the operators and parenthesis easily, that's why postfix conversion is needed.

Dr. Md. Abul Kashem Mia, Professor, CSE Dept and Pro-Vice Chancellor, UIU
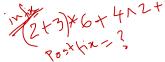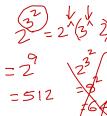
Mathematical Calculations

- The way we are use to writing expressions is known as infix notation.

- What does $3 + 2 * 6$ equal?   Ans: 15
  What does $3 * 2 + 6$ equal?   Ans: 12

*[handwritten: infix (2+3)*6 + 4^2+8, Postfix = ?]*

- The precedence of operators affects the order of operations.

- A mathematical expression cannot simply be evaluated left to right.

- Postfix expression does not require any precedence rules.

Operator	Precedence	Associativity
( )	1	L to R
^	2	R to L
*, /	3	L to R
+, -	4	L to R

[handwritten: $2^{3^2} = 2^{(3^2)} = 2^9 = 512$; $2^{3^2} = 2^9 = 64$]

Algorithm: Convert Infix to Postfix Expression

1. Scan the infix expression from left to right.
2. If the scanned character is an operand, output it.
3. Else [scanned character is an operator]
   a) if stack is empty or contains a left parenthesis '(' on top, push the incoming operator onto the stack.
   b) if the incoming symbol is '(', push it onto the stack.
   c) if the incoming symbol is ')', pop the stack and print the operators until left parenthesis is found.
   d) if the incoming symbol has higher precedence than the top of the stack, push it onto the stack.
   e) if the incoming symbol has lower precedence than the top of the stack, pop and print the top. Then test the incoming operator against the new top of the stack.
   f) if the incoming symbol has equal precedence with the top of the stack, use associativity rule.
      i) if associativity is L to R, then pop and print the top. Then push the incoming operator.
      ii) if associativity is R to L, then push the incoming operator.
4. At the end of the expression, pop and print all operators of stack.

*[handwritten: 4 + (8+2*5); Postfix: 482+5*+; stack: +()*]*

Simple Example 1

- Convert the following expressions from infix to postfix:

  3 + 2 * 4

  Infix Expression: 3 + 2 * 4
  PostFix Expression: 3 2 4 * +
  Operator Stack: + *

Infix to Postfix: Practice Question 1

- Convert the following expressions from infix to postfix:
  a) 2 ^ 3 ^ 4 + 5 * 7

  Postfix : 234∧∧57 * +

  Stack
  ∧
  ∧ ∧
  ∧
  +
  + *
  +

Infix to Postfix: Practice Question 1

- Convert the following expressions from infix to postfix:
  a) 2 ^ 3 ^ 4 + 5 * 7
      Ans: 2 3 4 ^ ^ 5 7 * +

  b) 9 + 2 - 5 * 6 / 3 + 2 ^ 3 / 4
      Ans: 9 2 + 5 6 * 3 / - 2 3 ^ 4 / +

  c) 1 + 3 ^ 2 + (5 * 6 - 4) * 7
      Ans: 1 3 2 ^ + 5 6 * 4 – 7 * +

Application of Stacks:
Evaluation of Postfix Expression

Stack = operand

2 + 3 × 4 = 14

post: 2 3 4 × + = ?

for each character in postfix expression do
- if it is an operand then push it onto the stack
- else if it is an operator (say, (op)) then
  - pop the stack for the right hand operand (say, b)
  - pop the stack for the left hand operand (say, a)
  - apply the operator to the two operands (result = a op b)
  - push the result onto the stack
- when the expression has been exhausted the result is the top (and only element) of the stack

Stack: 2 ~~3~~ ~~4~~ 12

12
3 × 4

result = 14

14
2 + 12

(14)

Practice Question 2

- What does the following postfix expression evaluate to?

  6 3 2 + *

  Postfix Expression: 6 3 2 + *

  Stack: 6 3 2

- What does the following postfix expression evaluate to?

  2 3 4 ^ ^ 5 7 * +

  $\frac{4}{3}$ 2 3 4 + − 5 7
                                                  * +
  stack = 2 3 4

  2^{81} 3 5

  = 2 81

  = 2^{81} 5 7 $\left(2^{81} + 3 5 \right)$