

# Lecture 34

## Expression Tree

## Multiway Trees

*December 07, 2021*  
*Tuesday*

# POLISH NOTATION

- One of the applications of binary trees is an unambiguous representation of arithmetical, relational, or logical expressions.
- In the early 1920s, a Polish logician, Jan Lukasiewicz (pronounced: wook-a-sie-vich)
- A special notation for propositional logic that allows us to eliminate all parentheses from formulas, called **Polish Notation**.

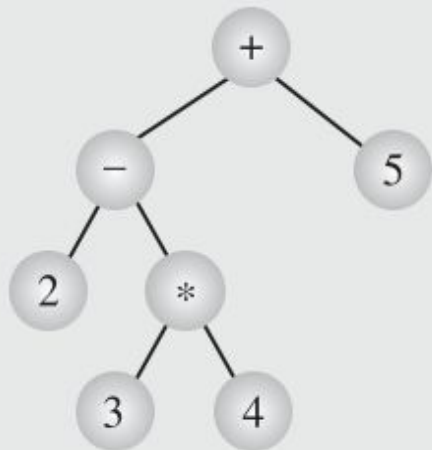
# POLISH NOTATION

- However, the polish notation resulted in **less readable formulas than the parenthesized originals** and were not used widely.
- **If avoiding ambiguity is the only goal**, then these symbols can be omitted at the cost of **changing the order of symbols** used in the formulas.

# POLISH NOTATION | EXAMPLE

- $2 - 3 \times 4 + 5$ 
  - Figure (A)
- $(2 - 3) \times (4 + 5)$ 
  - Figure (B)
- $2 - (3 \times 4 + 5)$ 
  - Figure (C)

# EXPRESSION TREE



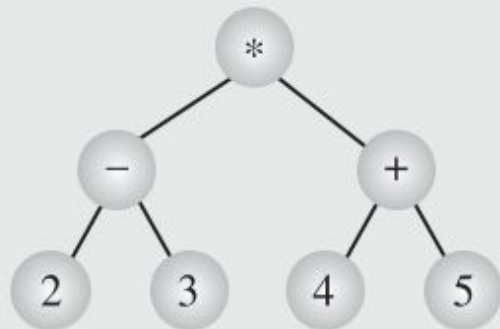
$2 - 3 * 4 + 5$

Preorder  $+ - 2 * 3 4 5$

Inorder  $2 - 3 * 4 + 5$

Postorder  $2 3 4 * - 5 +$

(a)



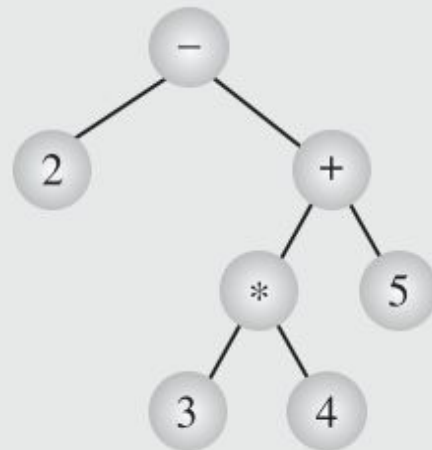
$(2 - 3) * (4 + 5)$

Preorder  $* - 2 3 + 4 5$

Inorder  $2 - 3 * 4 + 5$

Postorder  $2 3 - 4 5 + *$

(b)



$2 - (3 * 4 + 5)$

Preorder  $- 2 + * 3 4 5$

Inorder  $2 - 3 * 4 + 5$

Postorder  $2 3 4 * 5 + -$

(c)

# PREFIX TO EXPRESSION TREE

1. Read one symbol at a time from the prefix expression.
2. Check if the symbol is an operand or operator.
3. If the symbol is an operand, create a single node tree and return it.
4. If the symbol is an operator, create a new node
  - a. Create left Subtree
  - b. Create right subtree
5. Return the pointer to the root of the tree.

# INFIX TO EXPRESSION TREE

1. Create two Stacks, **CharStack** & **NodeStack**. Read one symbol at a time from the infix expression.
2. Check if the symbol is an operand or operator.
  - a. If the symbol is an operand, push it onto NodeStack.
  - b. If the symbol is an operator or opening brace, and the top of CharStack has lower precedence push it onto CharStack.
    - i. If ')' closing brace comes pop the CharStack and make it root.
      1. Pop two the NodeStack twice and make these children of the root.
      2. Until you visit the '(' in CharStack.
    - ii. Push the root onto NodeStack.

# POSTFIX TO EXPRESSION TREE

1. Read one symbol at a time from the postfix expression.
2. Check if the symbol is an operand or operator.
3. If the symbol is an operand, create a one node tree and push a pointer onto a stack
4. If the symbol is an operator, pop two pointers from the stack namely T1 & T2 and form a new tree with root as the operator, T1 & T2 as a left and right child
5. A pointer to this new tree is pushed onto the stack