

CSI 35 LECTURE NOTES (Ojakian)

Topic 17: Tree Traversals

OUTLINE

(References: Rosen: 11.3)

1. Prefix, postfix, infix
 2. Preorder, postorder, inorder
-

1. Representing Expressions - 3 Notations

Arithmetic, logical, etc.

- (a) Infix: Operator inbetween operands (usual way)
- (b) Prefix: Operator before operands
- (c) Postfix: Operator after operands.
- (d) Calculation
 - i. Prefix: calculate **right to left**. Postfix: Calculate **left to right**.
 - ii. When an operator is encountered, the operands appear immediately before it.
 - iii. The result becomes a new operand
- (e) Comparison
 - i. Infix: NOT left to right, in that follow order of operations.
 - ii. Postfix and Prefix: need no parentheses and are just read right to left (or left to right).
 - iii. Postfix and Prefix: operation order is just order in expression, left to right or right to left.
 - iv. Thus, typically, easier for a computer to work with postfix or prefix
- (f) Exercises.

PROBLEM 1. Calculate the following postfix expressions:

i. $5\ 2\ +\ 8\ 3\ -\ *$

ii. $6\ 13\ +\ 3\ 5\ -\ /\$

PROBLEM 2. Calculate the following prefix expressions

i. $*\ 9\ +\ 2\ 6$

ii. $+\ 7\ *\ 45\ +\ 2\ 0$

2. Tree Traversal

Do by quick method of Figure 9 (page 814).

- (a) Preorder: Add an element to your list the first time you pass it.
- (b) Postorder: Go **in reverse**. Add an element to **the end** of your list the first time you pass it
(i.e. the last time you pass it going the ordinary way)
- (c) Inorder: Add a leaf the first you pass it, and add an internal vertex the second time you pass it.

3. Expression Tree

- (a) Example: Put up tree for infix $3 * (4 + 2)$
- (b) Expression tree has no implicit order.
- (c) Can calculate from leaves up.

4. Expression Tree to Expression

- (a) Inorder Traversal - corresponds infix expression
- (b) Postorder Traversal - corresponds postfix expression
- (c) Preorder Traversal - corresponds prefix expression

5. Expression to Expression Tree

- (a) From expression, pick one of the 3 expression notations (infix, prefix, or postfix).
- (b) Carry out the calculation process, BUT instead of doing calculation, build the tree as you go:
 - i. When an operator applies, make it a new vertex, with its operands as its children
 - ii. Can step by step convert the expression to a tree, placing the subtree instead of the calculated value.

6. Translation

To translate one notation to another notation:

- (a) Start with an expression in one notation.
- (b) Build the expression tree.
- (c) Do the appropriate tree traversal of the other notation.

7. Exercises

Note: Re-ordered from the book.

- (a) Evaluation. 23 , 24
- (b) Expression trees. 16, 17, 18, 19, 22
- (c) Traversals. 7 - 15, 28, 29
- (d) Parentheses. 20, 21