Trees (I)

Hwansoo Han
Trees

- Hierarchical structure on a collection of items
  - Examples
    - Genealogy
    - Organization chart

- Academic areas
  - Computer science
    - Information organization in database systems
    - Syntactic structure of source code in compilers
  - Electrical circuit analysis
  - Structure of mathematical formulas
Terminology

- **Tree**
  - A collection of nodes
  - One of nodes is distinguished as a *root*
  - Hierarchical structure on nodes – *parenthood* relation
Recursive definition

A single node is a tree

- The node itself is the root of the tree

Construct a new tree by make a node \( n \) be the parent of \( n_1, n_2, \ldots, n_k \)

- Suppose \( n \) is a node and \( T_1, T_2, \ldots, T_k \) are trees
- \( n_1, n_2, \ldots, n_k \) are roots \( T_1, T_2, \ldots, T_k \) of respectively

Subtrees

Parent / children
Tree Example

Table of contents

Tree representation of TOC
Terminology

- **Path (from node \( n_1 \) to node \( n_k \))**
  - \( n_1, n_2, \ldots, n_k \) is a sequence in a tree, where \( n_i \) is the parent of \( n_{i+1} \) (\( 1 \leq i < k \))
  - **Unique path** from the root to any node in a tree
Terminology

- **Ancestor / descendant**
  - If there is a path from node \( a \) to node \( b \),
  - \( a \) is an ancestor of \( b \)
  - \( b \) is a descendant of \( a \)
  - Proper ancestor/descendant excludes the node itself

- **Sibling**
  - If two nodes have the common parent, they are siblings

- **Leaf**
  - Nodes in a tree that do not have children
Example

- Ancestors of s2.1?

- Descendants of s2.1?

- Siblings of s2.1?

- Leaves?
Terminology

- **Length (of a path)**
  - One less than the number of nodes in a path

- **Height (of a node, of a tree)**
  - Length of the longest path from the node to a leaf
  - Length of the longest path from the root to a leaf

- **Depth (of a node)**
  - Length of the unique path from the root to that node

- **Level**
  - Assume the root at level 0, level = depth
    - In some textbooks, assume the root at level 1, level -1 = depth
Example

- Length of a path from C2 to s2.1.1?
- Height of C1, height of the tree?
- Depth of s2.3?
- Level of s2.3?
Order of Nodes

- Left-to-right order for children
  - Unordered tree ignores the order of children

- Extended meaning
  - If a and b are siblings, and a is to the left of b, then all the descendants of a are to the left of all the descendants of b
Node Order Example

- Two trees are equal?

- List all nodes to the right of node 8
Preorder, Postorder, Inorder

- Systematical orderings all nodes of a tree
- Definitions of preorder, inorder, postorder listings
  - If a tree $T$ is null, the empty list is the 3 listings of $T$
  - If a tree $T$ has a single node, the node by itself is the 3 listings of $T$
  - $n$ is the root, $T_1, T_2, \ldots, T_k$ are subtrees
Definitions (continued)

- **Preorder**: 
  - \( n \), nodes in \( T_1 \) in preorder, nodes in \( T_2 \) in preorder, ..., nodes in \( T_k \) in preorder

- **Inorder**: 
  - nodes in \( T_1 \) in inorder, \( n \), nodes in \( T_2 \) in inorder, ..., nodes in \( T_k \) in inorder

- **Postorder**: 
  - nodes in \( T_1 \) in postorder, nodes in \( T_2 \) in postorder, ..., nodes in \( T_k \) in postorder, \( n \)
Preorder, Postorder, Inorder Example

- Preorder listing (traversal)
- Inorder listing (traversal)
- Postorder listing (traversal)
Labeled Trees

- Associate a *label*, or value with each node in a tree
  - Name of node is the position in a tree
  - Label of node is the value of that node, so it can be changed
Expression Tree – example labeled tree

- Represent an expression with labels
  - e.g. \((a+b)\*(a+c)\)
Expression Tree

- Prefix form of an expression
  - \((E_1) \, \theta \, (E_2)\), where \(\theta\) is a binary operator
  - \(\Rightarrow \, \theta \, P_1 \, P_2\), where \(P_1, P_2\) are prefix expressions of \(E_1, E_2\)
  - \((a+b)*(a+c) \Rightarrow \ast + ab + ac\)
  - No parentheses are necessary

- Postfix form of an expression
  - Reverse polish representation,
  - \(\Rightarrow \, P_1 \, P_2 \, \theta\), where \(P_1, P_2\) are postfix expressions of \(E_1, E_2\)
  - \((a+b)*(a+c) \Rightarrow ab + ac + \ast\)
  - No parentheses are necessary
Expression Tree

- Infix expression
  - Inorder traversal of expression tree
  - e.g. $a + b * a + c$
  - Requires parentheses for correct meaning

- Convert Infix to Reverse Polish Notation (RPN)
  - RPN = postfix expression
  - “Shunting yard algorithm” by Dijkstra