Trees

July 15, 2014

Definitions

Definition

A **tree** is a simple acyclic connected graph.

Definition

A **rooted tree** has a special node called the **root**. We usually draw the root on top, and all edges leading away from the root go downward (because we're just weird like that...).

What do you call a bunch of trees?

Nodes in trees

Definition

A **leaf** is a vertex of degree 1. An **internal node** is a non-leaf node.

Definition

In a rooted tree, a **parent** of a vertex is the vertex connected to it on the path to the root. A **child** of a vertex u is a vertex v such that u is the parent of v.

Definition

An **ancestor** of a vertex w is any vertex on the path from w to the root. A **descendant** of a vertex u is a vertex v such that u is an ancestor of v.

Types of trees

Definition

If G is a tree, then the **subtree rooted at** v is the subgraph of G consisting of v, all of the descendants of v, and the edges linking these nodes.

Definition

An m-**ary tree** is a rooted tree for which each vertex has at most m children.

Definition

An m-ary tree is **full** if every internal node has exactly 0 or n children.

Heights of trees

Definition

The **height** or **level** of a node is the length of the path between itself and the root (the root has height 0). The **height of a tree** is the maximum height of all of its leaves.

Definition

A tree is **complete** if all leaves have the same height.

Sizes of trees

- How many nodes are in a complete m-ary tree with height h?
- How many nodes are in a full m-ary tree with i internal nodes?

Structural induction

- I want to prove that a statement P is true for all trees.
- I let $P(h)$ say that P is true for trees of height h.
- (Base Case). I prove $P(0)$.
- (Inductive Step): I assume that $P(k)$ is true. Suppose that T is a tree with height $k+1$ and root r. Then P is true for each of the subtrees rooted at the children of r.
  I use this to prove that P is true for T, so $P(k+1)$ is true.

Example

Proposition

In a binary tree of height h, the number of nodes is at most $2^{h+1} - 1$.

Recursion trees

Suppose

$$f(n) = \begin{cases} 1 & \text{if } n = 1 \\ 2f(n/2) + n & \text{otherwise} \end{cases}$$

What is $f(n)$?

Recursion trees

Suppose

$$f(n) = \begin{cases} 1 & \text{if } n = 1 \\ f(n/2) + n & \text{otherwise} \end{cases}$$

What is $f(n)$?

Recursion trees

Suppose

$$f(n) = \begin{cases} 0 & \text{if } n = 0 \\ 3f(n/2) + n & \text{otherwise} \end{cases}$$

What is $f(n)$?

Context-free grammars

A CFG is a set of rules for rewriting strings. S is the starting string and is called the **start symbol**. Lowercase letters are **terminals** and cannot be rewritten. The symbol $|$ means "or."
Example:

1. $S \rightarrow x \mid y \mid z$
2. $S \rightarrow S + S$
3. $S \rightarrow S - S$
4. $S \rightarrow S * S$
5. $S \rightarrow S/S$
6. $S \rightarrow (S)$

CFG examples

(ϵ represents the empty string.)

Example:

1. $S \to aSb$
2. $S \to \epsilon$

Example:

1. $S \to aS \mid T$
2. $T \to aTb \mid \epsilon$.

Can all sets of strings be generated by CFGs? No.

Parse trees

A parse tree is a tree where each node is labeled with a string. The root is labeled with S and a node's children are labeled with the symbols that rewrite it.

Every string that can be generated from a grammar has its own parse tree (although it may have more than one).

Induction on CFGs

Consider the CFG

1. $S \to aSb$
2. $S \to \epsilon$

Prove that all strings generated by this CFG have the same number of a's and b's.

Another CFG induction example

Consider the CFG

1. $S \to aS \mid T$
2. $T \to aTb \mid \epsilon$.

Prove that all strings generated by this CFG have at least as many b's as a's.