COSC 290 Discrete Structures

Lecture 18: Proof by structural induction, part 2

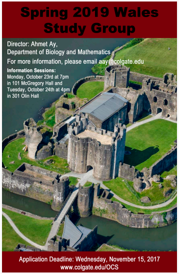
Prof. Michael Hay
Wednesday, Oct. 11, 2017
Colgate University

Plan for today

1. Properties related to tree height
2. Discuss mid-semester feedback

Announcement

Announcement



Recall: Binary Tree

A binary tree is either:

a) (base case) an empty tree, denoted *null*

b) (inductive case) a root node x, a left subtree T_ℓ, and a right subtree T_r where x is an arbitrary value and T_ℓ and T_r are both *binary trees*.

Terminology: nodes and edges

Can think of a tree T in terms of nodes and edges.

Node: root value x for each subtree in T.

Edge: a connection between a node and a *non-empty* subtree.

Proof of claim

- **Claim**: If T is non-empty, then $edges(T) = nodes(T) - 1$ where $edges(T)$ denotes the number of edges in T and $nodes(T)$ denotes the number of nodes.
- **Proof by structural induction**:
  - **Base cases:** T is empty, therefore the statement is true *(because the antecedent is False and $p \implies q$ is True whenever p is False).*
  - **Inductive case:** *(next slide)*

Faulty proof for inductive case

Claim: If T is non-empty, then $edges(T) = nodes(T) - 1$.

Inductive case: T is non-empty, consisting of node x and left and right subtrees T_ℓ and T_r.

Inductive hypothesis: $edges(T_\ell) = nodes(T_\ell) - 1$ (same for T_r).

$$nodes(T) = 1 + nodes(T_\ell) + nodes(T_r) \quad \text{(+1 for root)}$$

$$\begin{aligned} edges(T) &= (1 + edges(T_\ell)) + (1 + edges(T_r)) && \text{(+1 for each edge)} \\ &= (1 + (nodes(T_\ell) - 1)) + (1 + (nodes(T_r) - 1)) && \text{(inductive hypo.)} \\ &= nodes(T_\ell) + nodes(T_r) \\ &= nodes(T) - 1 \end{aligned}$$

- The inductive hypothesis only asserts a property for non-empty trees!
- What if T_ℓ is empty? What if T_r is empty?

Correct proof for inductive case

Claim: If T is non-empty, then $edges(T) = nodes(T) - 1$.

Inductive case: T is non-empty, consisting of node x and left and right subtrees T_ℓ and T_r. Cases:

- $T_\ell =$ null and $T_r =$ is null: $nodes(T) = 1$ and $edges(T) = 0$.
- $T_\ell \neq$ null and $T_r =$ null:

$$nodes(T) = 1 + nodes(T_\ell)$$
$$edges(T) = 1 + edges(T_\ell) = 1 + (nodes(T_\ell) - 1) = nodes(T) - 1$$

- $T_\ell =$ null and $T_r \neq$ null: same ideas as previous.
- $T_\ell \neq$ null and $T_r \neq$ null:

$$\begin{aligned} nodes(T) &= 1 + nodes(T_\ell) + nodes(T_r) \\ edges(T) &= 2 + edges(T_\ell) + edges(T_r) \\ &= 2 + (nodes(T_\ell) - 1) + (nodes(T_r) - 1) \\ &= nodes(T_\ell) + nodes(T_r) = nodes(T) - 1 \end{aligned}$$

Properties related to tree height

Height of a tree

The level of a node in T is the length of the path from it to the root of T. The height of a tree is the maximum level of any (leaf) node in T.

We can also define height recursively: let $h(T)$ denote the height of tree T.

- Base case: tree T is empty, $h(T) = -1$.
- Inductive case: T is non-empty, thus it consists a root node x, a left subtree T_ℓ, and a right subtree T_r. Then, $h(T) = 1 + \max\{\, h(T_\ell), h(T_r) \,\}$.

Exercise: prove claim

Claim: $nodes(T) \leq 2^{h(T)+1} - 1$

Please work in small groups at the white boards.

Lower bound?

- **False Claim**: $nodes(T) \geq 2^{h(T)+1} - 1$
- **Faulty proof by structural induction**:
  - **Base cases:** T is empty, height is -1 and $nodes(T) \geq 2^{-1+1} - 1 = 0$.
  - **Inductive case:** T is a non-empty tree of height h, consisting of node x and left and right subtrees T_ℓ and T_r.

$$\begin{aligned} nodes(T) &= 1 + nodes(T_\ell) + nodes(T_r) && \text{(a. +1 for root)} \\ &\geq 1 + (2^{h(T_\ell)+1} - 1) + (2^{h(T_r)+1} - 1) && \text{(b. ind. hypothesis)} \\ &\geq 1 + (2^{(h-1)+1} - 1) + (2^{(h-1)+1} - 1) && \text{(c. subtree heights)} \\ &= 2^{h+1} - 1 = 2^{h(T)+1} - 1 && \text{(d. algebra)} \end{aligned}$$

Where's the flaw? A) Inductive case, first sentence; B) Inductive case, line a; C) Inductive case, line b; D) Inductive case, line c; E) Inductive case, line d.

Poll: number of leaves

We just showed an upper bound on the number of nodes in T: $nodes(T) \leq 2^{h(T)+1} - 1$. What can we say about *leaves*(T), the number of leaves?

Give the *smallest* upper bound you can. (Hint: try some examples... then start sketching out a proof!)

Claim: $leaves(T) \leq$ what goes here?

A) 0
B) $2^{h(T)-1}$
C) $2^{h(T)}$
D) $2^{h(T)+1}$
E) $2^{h(T)+1} - 1$

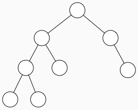
Height balanced

A binary tree is height balanced if for any node in the tree, the height of the left subtree and the height of the right subtree can differ by at most 1.

Poll: height balanced?

A binary tree is height balanced if for any node in the tree, the height of the left subtree and the height of the right subtree can differ by at most 1. (Empty subtree is -1)

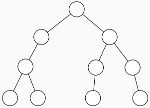
Is this tree height balanced?



A) Yes
B) No
C) I'm not sure

Poll: height balanced?

A binary tree is height balanced if for any node in the tree, the height of the left subtree and the height of the right subtree can differ by at most 1. (Empty subtree has height -1.)



Is this tree height balanced?

A) Yes
B) No
C) I'm not sure

Discuss mid-semester feedback