COSC 290 Discrete Structures

Lecture 17: Proof by structural induction

Prof. Michael Hay
Friday, Oct. 6, 2017
Colgate University

Plan for today

1. Wrap up Strong Induction
2. Structural Induction
3. Mid-semester feedback

Wrap up Strong Induction

Tilings

Jacobsthal numbers: $J_0 = 0$, $J_1 = 1$ and $J_n = J_{n-1} + 2J_{n-2}$ for $n \geq 2$.

1. Claim: for any $n \geq 0$, given $n \times 2$ grid, the number of tilings using either 1×2 dominoes or 2×2 squares is J_{n+1}.
2. Claim: $J_n = \frac{2^n - (-1)^n}{3}$

Proof shown on board

Structural Induction

Recursively defined structures

A recursively defined structure is a structure defined in terms of one or more *base* cases and one or more *inductive* cases.

Applications in computer science

Many fundamental computer science structures are recursively defined structures:

- lists
- trees
- propositional logic
- circuits
- syntax of all programming languages

Many practical systems/applications are built using recursively defined structures. Example: Apache Spark Resilient Distributed Datasets (RDDs).

Having the ability to reason about such structures is important!

Example: 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.

Examples shown on board.

Proof by Structural Induction Template

Let S be a (well-ordered) set of structures generated by a recursive definition.

- **Claim**: $\forall x \in S : P(x)$
- **Proof by structural induction**:
  - **Base cases:** for every x defined as a base case in the definition of S, prove $P(x)$.
  - **Inductive case:** for every x defined in terms of $y_1, y_2, \ldots, y_k \in S$ by an inductive case in the definition of S, prove that $P(y_1) \wedge P(y_2) \wedge \cdots \wedge P(y_k) \implies P(x)$.
    - **Assume:** $P(y_1) \wedge P(y_2) \wedge \cdots \wedge P(y_k)$ is true.
    - **Want to show:** $P(x)$ is true.
    - "Suppose that [$P(y_1) \wedge P(y_2) \wedge \cdots \wedge P(y_k)$ is true]... "
    - ... *body of proof for inductive case...*
    - "... therefore the inductive step holds."
  - **Conclusion:** "By structural induction, the claim has been shown."

Property of trees

Claim: For any binary tree T, 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 of claim

- **Claim**: Let T be a binary tree. If T is non-empty, then $edges(T) = nodes(T) - 1$.
- **Proof by structural induction**:
  - **Base cases:** T is empty, therefore...
  - **Inductive case:** T is non-empty, consisting of node x and left and right subtrees T_ℓ and T_r.
    Therefore...

Poll: base case

- **Claim**: If T is non-empty, then $edges(T) = nodes(T) - 1$.
- **Proof by structural induction**:
  - **Base cases:** T is empty, therefore… what goes here?

A) $nodes(T) = 1$ and $edges(T) = 0$ because T is empty.
B) The claim does not apply because T is empty.
C) The claim does is false because T is empty.
D) The claim is true because T is empty.
E) None of the above.

Poll: base case

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

Poll: 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).

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

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

A) The proof is correct.
B) The inductive assumption is incorrect.
C) There is an error in the proof logic/math.
D) None / More than one

- The inductive hypothesis only asserts a property for non-empty trees!

Proof of claim

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:

$$nodes(T) = 1 + nodes(T_\ell) + nodes(T_r)$$

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

Mid-semester feedback

Your feedback

Please complete the feedback form. When you are finished, please place your form on the front desk.

(If the last person could bring the forms to my office, I would appreciate it!)