CS 151

Structural Induction

Wednesday, October 17, 12

Announcements

Test #1 on Friday in class. If you need special considerations (extra time, quiet space, different time, etc.) talk to me today, either via email or in office hours.

Test topics:

- advanced Java:
  - abstract classes and interfaces, inheritance, exceptions, javadoc, jUnit

- algorithms analysis:
  - definition of big-Oh, big-Omega, little-Oh, little-omega, and big Theta
  - determine the running time of various algorithms (including log n ones)
  - linear and binary search, selection, insertion and merge sort
  - data structures: operations, implementations, running times
  - insertion and merge sort
  - definition of big-Oh, big-Omega, little-Oh, little-omega, and big Theta
  - abstract data and interfaces, inheritance, exceptions, polymorphism

Test Topics:

Structural Induction

We've seen some recursively defined data structures: LL, BT's, BST's, AVL trees. Suppose we want to prove some property about all instances of a structure. We'll use the recursive definition of our structures to help us prove these things.

To show that the property P(T) is always true we just need to show it is true for both cases:

1. An AVL tree T is either:
   - an empty tree
   - a root node r with left and right (sub) AVL trees whose heights differ by ≤ 1

2. A root node r with height h (and all AVL trees whose heights differ by ≥ 1)

Ex: An AVL tree T is either:

- A root node r with height h (and all AVL trees whose heights differ by ≥ 1)
- An AVL tree T is either:
  - an empty tree
  - a root node r with left and right (sub) AVL trees whose heights differ by ≤ 1

We've seen some recursively defined data structures: LL, BT's, BST's, AVL trees.
So the recursive case is true, and we are done with our proof!

So the recursive case is true, and we are done with our proof!

When $h(R) = h(L)$, we have

$$T(h) + f_1 + f_2 \leq 2 + \frac{1}{2}$$

because $h(R)$, $h(L)$ differ by 1, and $f_1$, $f_2$ are tree heights.

Our induction hypothesis (IH) assumes that $P(n')$ is true, i.e. that

$$f_1 + 2f_2 \leq 2 + \frac{1}{2}$$

for all $0 \leq n' \leq 2k$.

For a nat num $n$, let $P(n)$ be the boolean property that

$$f_1 + 2f_2 \leq 2 + \frac{1}{2}$$

for all $0 \leq n \leq 2k$.

Our induction hypothesis (IH) assumes that $P(n) = 0$.

So the base case is true because $P(n)$ is true when $n = 0$.

Then by definition of exponentiation

$$0^x = 0$$

by definition of $0$.

$$1^x = 1$$

by definition of $1$.

$$0 = 0$$

by definition of $0$.

Thus the base case is true.

Then plugging these in, we get

$$T(h) + f_1 \leq 2 + \frac{1}{2}$$

because $f_1$, $f_2$ are tree heights.

We want to show (WTS) that $P(T)$ is true when $T$ is an empty tree, i.e. that

$$f_1 + 2f_2 \leq 2 + \frac{1}{2}$$

for a AVL T, let P(T) be the boolean property that

An AVL tree T is either

structural induction:

Structural Induction Example 1

For a nat num n, let P(n) be the boolean property that

A natural number n is either

structural induction:

Structural Induction Example 2

Our induction hypothesis (IH) assumes that $P(n')$ is true, i.e. that

$$f_1 + 2f_2 \leq 2 + \frac{1}{2}$$

for all $0 \leq n' \leq 2k$.

For a nat num n, let P(n) be the boolean property that

A natural number n is either

structural induction:

Structural Induction Example 2
Also, for every natural number $n$, we know that

\[ \text{height}(T) = O(\log n) \]

Namely, height(T) is $O(\log n)$.

Therefore, by two inductive steps, we get

\[ \log(\text{nodes}(T)) \geq \frac{f(\text{height}(T)) + 2}{2} \]

Take the log (base-2) of both sides (to get rid of the exponent on the right):

\[ \log(\text{nodes}(T)) \geq \frac{f(\text{height}(T)) + 1}{2} \]

Multiply both sides by two then add one to both to get

\[ 2\log(\text{nodes}(T)) + 1 \geq f(\text{height}(T)) \]

For every AVL tree $T$, we know that

\[ \text{nodes}(T) \geq f(\text{height}(T)) + 1 \]

Also, for every natural number $n$, we know that

\[ \text{height}(T) = O(\log n) \]