Induction and Recursion

Clay Thomas

February 24, 2015

Let's just take a casual look at some pseudo-code that implements induction.

```
--A tree is either a Nil (zero node tree, aka null)
--or it holds an a along with a left and right subtree
data Tree a = Nil
            | Tree a (Tree a) (Tree a)
--Simply count the nodes
numNodes :: Tree a → Int
numNodes Nil = 0
numNodes (Tree a left right)
  = 1 + numNodes left + numNodes right

--Count the nodes without any children
numLeaves :: Tree a → Int
numLeaves Nil = 0
numLeaves (Tree a Nil Nil) = 1
numLeaves (Tree a left right)
  = numLeaves left + numLeaves right

--Find the length of the longest path from the root
height :: Tree a → Int
height Nil = (-1)
height (Tree a left right)
  = 1 + max (height left) (height right)
```

Each of the above functions uses recursion. Recursion and induction work hand in had. Induction builds up complicated theorems from small foundational steps. Recursion reduces complicated problems to simple base cases.

Recursion is not just **a** way to solve problems in computer science, it is **the** way to solve problems in computer science. Your first thoughts when approaching a task should always be "how can I make this task smaller." If you can phrase a big problem in terms of a smaller instance of itself, you have an opportunity to use recursion.

1 Structural Induction

First, try so called "structural induction" which is when you induct on a data structure instead of on a natural number.

1.1 Quicksort

Quicksort is a sorting algorithm that uses the following rough algorithm:

1. Pick a "pivot value" and remove it from the list.
2. Filter all the other values into two lists, "less" and "greater", according to whether they are less than or equal to the pivot, or greater than the pivot.
3. Quicksort less and greater.
4. Your result is the list that contains less, then the pivot, then greater, in that order.

Argue for the correctness of quicksort and specify the necessary base cases.

1.2 Tree Functions

For each of the function provided in the introduction, prove or provide an argument that each function works correctly.

Furthermore, show that $numLeaves(T) \leq 2^{height(T)}$.

2 Mathematical Induction

Induction usually works on a sequence of natural numbers, as in the following proofs:

- $\Sigma_{k=1}^{n} k^3 = \left(\Sigma_{k=1}^{n} k\right)^2$
- $1 \cdot 2 \cdot 3 + 2 \cdot 3 \cdot 4 + ... + n(n+1)(n+2) = \frac{n(n+1)(n+2)(n+3)}{4}$

It is often difficult to come up with the answer to a recursion-type problem and prove its correctness at the same time. Try to derive a "closed form" expression for the following sequence, then prove your answer.

- $a_n = 3a_{n-1} + 4n$ in terms of a_0