Inductive Proofs and Definitions

Recursion's Mathematical Cousin

Wednesday, September 12, 2007

Reading: Stoughton 1.2 – 1.3

CS235 Languages and Automata

Department of Computer Science
Wellesley College

Goals for today

- Proof by induction
- Strong induction
- Trees and structural induction

The Induction Principle for Nat

Suppose P(n) is a property mentioning some $n \in$ Nat.

Suppose that

1. (basis step) P(0) holds.
2. (inductive step) For all $n \in$ Nat, P(n) $\Rightarrow$ P(n+1).

the inductive hypthesis (IH)

Then P(n) holds for all $n \in$ Nat.

Note: the inductive step is sometimes rephrased

for all naturals $n > 0$, P(n-1) $\Rightarrow$ P(n)

the inductive hypthesis (IH)

Why Induction Works

The **modus ponens** rule of logic:

$$\frac{p \,;\, p \Rightarrow q}{q}$$

(this means $(p \wedge (p \Rightarrow q)) \Rightarrow q$)

With the basis step and inductive step,
we can derive P(n) for any n:

$$\frac{\dfrac{\dfrac{\dfrac{P(0) \,;\, P(0) \Rightarrow P(1)}{P(1)} \quad ; \; P(1) \Rightarrow P(2)}{P(2)} \quad ; \; P(2) \Rightarrow P(3)}{\vdots}}{P(n)}$$

An Example: Summing 1 to n

Prove that $\Sigma_{i=1}^{n} i = n \cdot (n + 1)/2$

1. (basis step) Show that $\Sigma_{i=1}^{0} i = 0 \cdot (0 + 1)/2 = 0$

2. (inductive step)

Assume that $\Sigma_{i=1}^{n} i = n \cdot (n + 1)/2$ ← the inductive hypthesis (IH)

Show that $\Sigma_{i=1}^{n+1} i = (n + 1) \cdot (n + 2)/2$

A Slightly Different Approach

Prove that $\Sigma_{i=1}^{n} i = n \cdot (n + 1)/2$

Case 1: n = 0 (base case)

Show that $\Sigma_{i=1}^{0} i = 0 \cdot (0 + 1)/2 = 0$

Case 2: n > 0 (inductive case)

Assume that $\Sigma_{i=1}^{n-1} i = (n - 1) \cdot n /2$ ← the inductive hypthesis (IH)

Show that $\Sigma_{i=1}^{n} i = n \cdot (n + 1)/2$

This approach highlights the similarity of induction to recursive definitions

Other Ways to Skin A Cat

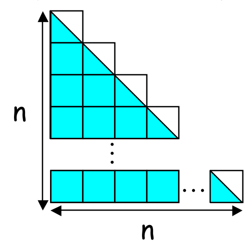
Proofs by induction are common, especially in CS235, but sometimes there are other ways to prove the same thing.

Another proof that $\sum_{i=1}^{n} i = n \cdot (n+1)/2$:

$$1 + 2 + 3 + \ldots + (n-2) + (n-1) + n$$

n/2 pairs, each of which sums to (n + 1)

And yet another proof – by picture:



blue area:	$n^2/2$
white area:	$n/2$
sum:	$n \cdot (n+1)/2$

Another Simple Example

Recursive definition of integer exponentiation:

- $x^0 = 1$
- $x^n = x \cdot x^{n-1}$

Using this defn., prove that $x^{a+b} = x^a \cdot x^b$, where $a,b \in$ Nat.

Proof: Assume b is fixed. Proof is by induction on a.

Base Case: a = 0

Inductive Case: a > 0

Assume $x^{(a-1)+b} = x^{a-1} \cdot x^b$ ← the inductive hypthesis (IH)

Strong Induction for Nat

Suppose P(n) is a property mentioning some $n \in Nat$.

Suppose that

(inductive step) For all $n \in Nat$, $(\bigwedge_{i=0}^{n-1} P(i)) \Rightarrow P(n)$.

Then P(n) holds for all $n \in Nat$.

the inductive hypthesis (IH)

The following derivation shows why this works:

$$\frac{\dfrac{\dfrac{\dfrac{() \Rightarrow P(0)}{P(0)} \quad ;\ (P(0)) \Rightarrow P(1)}{P(1)} \quad ;\ (P(0) \wedge P(1)) \Rightarrow P(2)}{P(2)} \quad ;\ (P(0) \wedge P(1) \wedge P(2)) \Rightarrow P(3)}{\dfrac{\vdots}{P(n)}}$$

( ) is the conjunction of zero premises, and so = T

Strong Induction Example

Consider the recursive definition of fast exponentiation:

fastexp(x,0) = 1
fastexp(x,n) = x ·fastexp(x, n-1), if n is odd
fastexp(x,n) = $(fastexp(x,n/2))^2$, if n is even

Prove that fastexp(x,n) = x^n

Can't use regular induction. Why?

Proof by strong induction:

Assume that forall i < n, fastexp(x,i) = x^i ← the inductive hypthesis (IH)

Case 1: n = 0

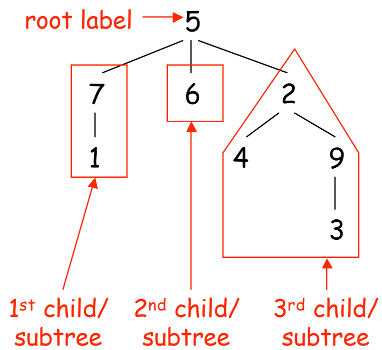
Case 2: n > 0 and is odd

Case 3: n > 0 and is even

Tree_X : Intuitions and Terminology

Tree_X is the set of all trees whose nodes are in the set X.

Here is a tree in Tree_{Nat}:

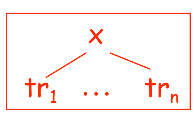


- Different nodes can have different numbers of children
- Nodes without children are **external nodes**, a.k.a. **leaves** (e.g. 1, 6, 4, 3)
- Nodes with children are **internal nodes** (e.g. 5, 7, 2, 9)
- Use $x(tr_1, \ldots, tr_n)$ to write a tree and abbreviate leaf x() as x. E.g.:

5(7(1), 6, 2(4, 9(3)))

Inductive Definition of Tree_X

Formally, Tree_X is the least set s.t.
for all $x \in X$, $n \in Nat$, and $tr_1, \ldots, tr_n \in \text{Tree}_X$,

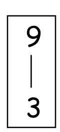


$\in \text{Tree}_X$

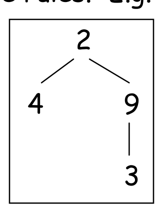
This is an **inductive definition**, which defines a set as the **least** set of elements satisfying a collection of rules — here, the one rule in red. (In contrast, a **co-inductive definition** uses the ***greatest*** set.) Each element can be constructed "from bottom up" in some finite number of steps using the rules. E.g. :

4 $\in \text{Tree}_{Nat}$

3 $\in \text{Tree}_{Nat}$



$\in \text{Tree}_{Nat}$



$\in \text{Tree}_{Nat}$

Structural Induction on Trees

Suppose P(tr) is a property mentioning some tr ∈ Tree .

Suppose that

(inductive step)

For all x ∈ X, n ∈ Nat, and $tr_1, \ldots, tr_n \in Tree_X$,

$(\bigwedge_{i=1}^{n} P(tr_i)) \Rightarrow P(\, x(tr_1, \ldots, tr_n)\,)$.

the inductive hypthesis (IH)

Then P(tr) holds for all $tr \in Tree_X$.

Note: in practice, typically need a leaf case and a nonleaf (internal node) case.

Structural Induction Example

In a **binary tree**, each internal node has two children.

Prove that, in every binary tree, the number of leaves (#L) is one more than the number of internal nodes (#I). I.e.

#L (tr) = #I(tr) + 1

Leaf case: tr is a leaf

Nonleaf case: tr is $x(tr_L, tr_R)$

Assume that for i in {L,R}, #L (tr_i) = #I(tr_i) + 1 ← the inductive hypthesis (IH)

Nat Induction is an Instance of Structural Induction

Every natural number can be viewed as a simple tree.

E.g.: