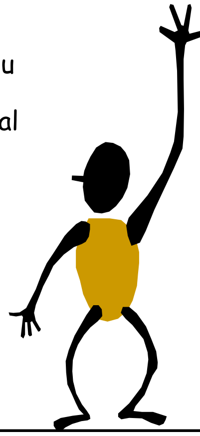
Great Theoretical Ideas In Computer Science

Victor Adamchik
CS 15-251
Lecture 2
Carnegie Mellon University

Inductive Reasoning

Raise your hand if you NEVER heard of mathematical induction

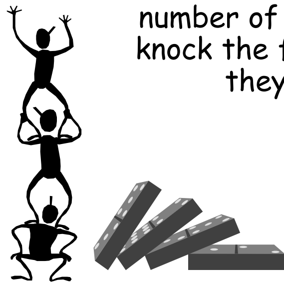


Mathematical Induction

- Standard Form
- Strong Form
- Least Element Principal
- Invariant Form
- Structural Induction

American Banks in 2008

Domino Effect: Line up any number of dominos in a row; knock the first one over and they will all fall



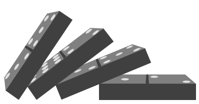
Dominoes Numbered 1 to n

D_k : "The k^{th} domino falls"

If we set them up in a row then each one is set up to knock over the next.
Here are the rules:

1. D_1
2. $D_k \Rightarrow D_{k+1}$ (if k^{th} falls, then $(k+1)^{st}$ falls)

$D_1 \Rightarrow D_2 \Rightarrow D_3 \Rightarrow$ ...
All Dominoes Fall



Proof by Mathematical Induction

In formal notation. Let P(k) be a statement

$$[P(0) \wedge \forall k\,(P(k) \Rightarrow P(k+1))] \longrightarrow \forall n\, P(n)$$

Instead of attacking a problem directly, we only explain how to get a proof for P(k+1) out of a proof for P(k)

Plain Induction

Suppose we have some statement P(n) that holds for some natural numbers n.

To demonstrate that P(n) is true for all n is a little problematic.

Inductive Proofs

Base step(s): Show that P(0) holds

Induction Hypothesis: Assume that P(k) holds

Induction Step: Show that P(k) implies P(k+1)

Example

Prove that $2^n < n!$ for $n \geq 4$.

Base step: n =4, $2^4=16 < 4! =24$

IH: assume $2^k < k!$

Prove it for k+1

$2^{k+1} = 2 \cdot 2^k < 2 \times k! < (k+1)!$

A Template for Induction Proofs

State that the proof uses induction.
If there are several variables, indicate which variable serves as k.

Define a statement P(k), aka IH

Prove initial case(s).

Prove that P(k) implies P(k+1), aka IS

State the induction principle allows you to conclude that P(n) is true for all nonnegative n.

Upper bound for P_n

$T(n) = 2 + T(n-1)$
$T(2) = 1$

Induction is helpful for <u>proving</u> the correctness of a statement, but not helpful for <u>discovering</u> it.

Prove $T(n) = 2 \cdot n-3$

Upper bound for P_n

$T(n) = 2 + T(n-1)$
$T(2) = 1$

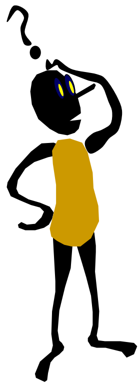
Prove $T(n) = 2 \cdot n-3$

Base step: n =2, $T(2) = 2 \cdot 2-3 = 1$

IH: assume $T(k) = 2 \cdot k-3$

Prove it for k+1

$T(k+1) = 2 + T(k) = 2 + 2 \cdot k-3 = 2 \cdot (k+1)-3$



Funny Example

Any group of people are all of the same gender

Prove this by induction

Any group of k people are all of the same gender

Base step: When k = 1, one person can certainly only be one gender.

IH: Suppose the claim holds for k people

Inductive step: given a set of k+1 people. Consider the subset formed by removing a person. WLOG, let it be (k+1)st.

Any group of k people are all of the same gender

The remaining 1…k people are of the same gender (let say girls), by IH

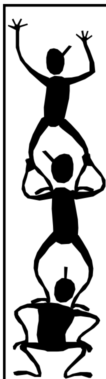
$$\underline{S_1, S_2, \ldots, S_k}, S_{k+1}$$

In particular, #2 is a girl. Also, by IH, from 2 to (k+1) are girls

$$S_1, \underline{S_2, S_3, \ldots, S_{k+1}}$$

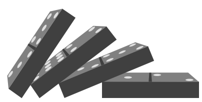
P(1) ⇏ P(2)

Therefore, all students are girls☺



Soundness of Induction

How do we know that INDUCTION really works?



Soundness of Induction

Proof by contradiction

Assume that for statement P(n), we can establish the base step P(0), and the induction step, but nonetheless it is not true that P(n) holds for all n.

So, for some values of n, P(n) is false.

$[P(0) \wedge \forall k\,(P(k) \Rightarrow P(k+1))] \longrightarrow \forall n\, P(n)$

Soundness of Induction

Let n_0 be the least such n that $P(n_0)$ is false.

Certainly, n_0 cannot be 0.

Thus, it must be $n_0 = n_1+1$, where $n_1 < n_0$.

Now, by our choice of n_0, this means that $P(n_1)$ holds.

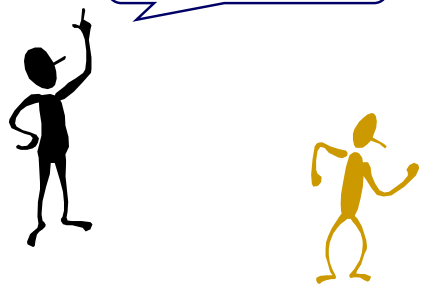
because of IH, since $n_1 < n_0$

Soundness of Induction

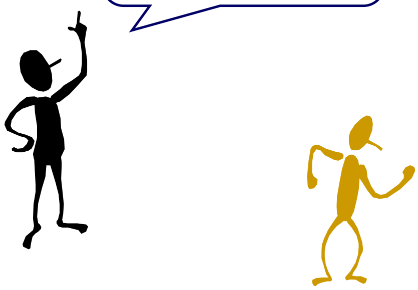
But then by Induction Step, $P(n_1+1)$ also holds.

Which is the same as $P(n_0)$, and we have a contradiction.

Review that proof



We can pick n_0 to be the least n where P(n) fails.



Least Element Principle

Every non-empty subset of the natural numbers must contain a least element.

Theorem

Every natural number > 1 can be factored into primes

Theorem. Every natural number > 1 can be factored into primes

Base case: 2 is prime

Inductive Hypothesis:
n can be factored into primes

How do we prove it for n+1 ?

A different approach:

Assume 2, 3,..., n all can be factored into primes

Then show that n+1 can be factored into primes

With respect to dominoes, we assume that (k+1)st falls because ALL previous dominoes fall.

Strong Induction

Establish Base Case: P(0)

Assume $\forall k \le n$, P(k) is true

Derive P(n+1)

$[P(0) \wedge (P(0),P(1),\dots,P(k) \Rightarrow P(k+1))] \longrightarrow \forall n\, P(n)$

Theorem. Every natural number > 1 can be factored into primes

Base case: 2 is prime

Inductive hypothesis:
$k \le n$ can be factored into primes

Case 1: n+1 is prime
Case 2: n+1 is composite, n+1 = p q
p, q > 1

Strong vs Weak

These two forms of induction are equivalent.
The conversion from weak to strong form is trivial
From strong in P to weak:

Let $Q(n): P(0) \wedge P(1) \wedge \dots \wedge P(n)$
Base Step: $Q(0) = P(0)$
Inductive Step: $Q(n) \Rightarrow P(n+1)$
$Q(n) \Rightarrow Q(n) \wedge P(n+1)$
$Q(n) \Rightarrow Q(n+1)$

Therefore, the strong induction in P can be written as a weak induction in Q

ATM Machine

Suppose an ATM machine has only seven dollar and ten dollar bills. You can type in the amount you want, and it will figure out how to divide things up into the proper number of 7's and 10's.

Claim: The ATM can generate any output amount $n \ge 54$.

ATM Machine: Proof

Base case: $54 = 2\cdot7 + 4\cdot10$

Induction step: assume $k = 7\cdot a + 10\cdot b$, for all k = 54, ..., n

How do we proceed for k=n+1 dollars?

ATM Machine: Proof

$n + 1 = n - 6 + 7, \ n \geq 54$

By IH: $n - 6 = 7 \cdot a + 10 \cdot b$

Hmm…, $n - 6$ could be less than 54…

Therefore, we have to extend the base cases to 55, 56, 57, 58 and 59

ATM Machine: Proof

Base cases:
$54 = 2 \cdot 7 + 4 \cdot 10$
$55 = 5 \cdot 7 + 2 \cdot 10$
$56 = 8 \cdot 7$
$57 = 1 \cdot 7 + 5 \cdot 10$
$58 = 4 \cdot 7 + 3 \cdot 10$
$59 = 7 \cdot 7 + 1 \cdot 10$

Induction step: assume $k = 7 \cdot a + 10 \cdot b$, for $k = 54, …, n$

ATM Machine: Proof

$n + 1 = (n - 6) + 7, \ n \geq 60$

By IH: $(n - 6) = 7 \cdot a + 10 \cdot b$

$n + 1 = 7 \cdot a + 10 \cdot b + 7 = 7 \cdot (a+1) + 10 \cdot b$

Therefore any number $\geq$ 54 can be formed using 7 and 10 bills

Faulty Induction

Claim. 251*n = 0 for all n>=0.

Base step: Clearly 251*0 = 0.

IH: Assume that 251*k=0 for all 0<=k<=n.

We need to show that 251*(n+1) is 0.
Write n+1=a+b., where a, b > 0.
251*(n+1) = 251*(a+b) = 251*a + 251*b =0+0 = 0

Chocolate Bar

Given a chocolate bar made up of n by k squares. At each step, you choose a piece of chocolate and snap it in two along any line, vertical or horizontal.

Show by induction that the number of snaps required to reduce it to single squares is
n k - 1.

Chocolate Bar

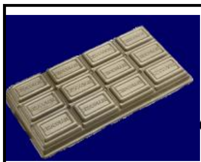
Claim. It requires nk-1 snaps

Note, there are two variables n and k.

We prove it by strong induction on the number of squares s = n k.

If s = 1, no breaks are required and
n k-1 = s -1 = 1-1 = 0

Chocolate Bar



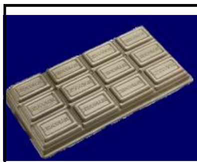
Claim. It requires s-1 snaps

IH: assume that this is true for $\forall s < n$

Consider a chocolate bar with s = n total pieces.

After the first snap, there will be two smaller bars with k_1 and $n-k_1$ pieces.

Chocolate Bar

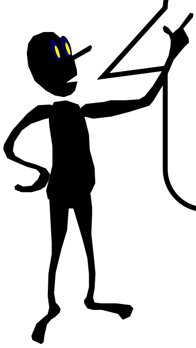


Claim. It requires s-1 snaps

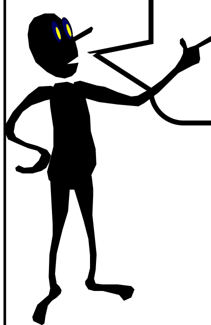
After the first snap, there will be two smaller bars with k_1 and $n-k_1$ pieces.

By IH, it requires $k_1 - 1$ and $n-k_1-1$ breaks.

Totally, $(k_1 - 1)+(n-k_1-1)+1= n-1$



And there are more ways to do inductive proofs



Yet another way of packaging inductive reasoning is to define "invariants"

In Programming. A rule that applies throughout the life of a data structure or procedure or loop. Each change to the data structure or loop maintains the correctness of the invariant

15-122 again …

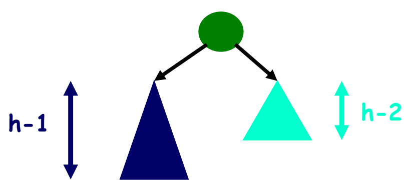
Loop invariants

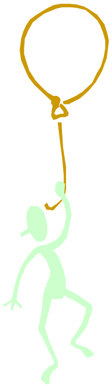
```
int fast_exp (int x, int y)
//@requires y > 0;
//@ensures \result == pow(x,y);
{
  int r = 1; int b = x; int e = y;
  while (e > 1)
  //@loop_invariant r * pow(b,e) == pow(x,y);
  {
     if (e % 2 == 1) r = b * r;
     b = b * b; e = e / 2;
  }
  return r * b;
}
```

15-122 again …

Data structure invariants:

BST, Heaps, AVL Trees





Inductive reasoning is the high level idea

"Standard" Induction
"Strong" Induction
"Least Element Principal"
"Invariants"
are all just
different packaging

Inductive Definition

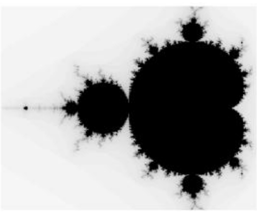
A linked list is either empty list or a node followed by a linked list

A binary tree is either empty tree or a node containing left and right binary trees.

F(0) = 0, F(1) = 1 *recursive function*
F(n) = F(n-1) + F(n-2)

Fractals

Fractals are geometric objects that are self-similar, i.e. composed of infinitely many pieces, all of which look the same.



The Koch Curve

Alphabet: { F, +, - }

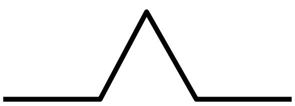
Rules: Rule(F) = F+F--F+F
Rule(+) = +
Rule(-) = -
Rule(w_1 ...w_n) = Rule(w_1) … Rule(w_n)

Rule (F): F+F--F+F

Rule(Rule (F)):

F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F

The Koch Curve

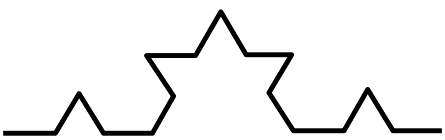


F+F--F+F

Visual representation:_

F draw forward one unit
\+ turn 60 degree left
\- turn 60 degrees right

The Koch Curve

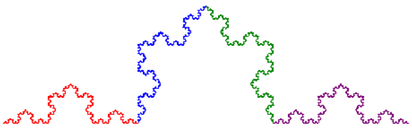


F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F

Visual representation:_

F draw forward one unit
\+ turn 60 degree left
\- turn 60 degrees right

The Koch Curve



a non-differentiable curve

Structural Induction

We can prove certain properties of inductively-defined sets and types (in ML, for example)

Structural Induction uses the same steps as regular induction.

1. Prove the base cases of the definition.
2. Prove the result of any recursive combination rule, assuming that it is true for all the parts

Recursively Defined Sets

Consider a set S defined by

Base Step: $3 \in S$

Recursive Step: if $x \in S$ and $y \in S$, then $x+y \in S$

What does S contain?

Structural Induction

1. Base Step: $3 \in S$
2. if $x \in S$ and $y \in S$, then $x+y \in S$

We will prove by induction that the set S contains number divisible by 3.

Base step: $3 \mid 3$.

Inductive step: Assume, $3|x$ and $3|y$, then $x = 3a$, $y = 3b$

$z=x+y = 3a+3b = 3(a+b) = 3c$, where $c=a+b$, thus $3|z$.

Structural Induction

Theorem . For any non-empty binary tree $T = (V; E)$, $|V| = |E| + 1$.

Base step: T is a single root node: $|V|=1$, $|E|=0$

Induction step: Assume T contains T_L and may contain T_R, for which the claim is true:

$|V_L|=|E_L|+1$, $|V_R|=|E_R|+1$,

If T_R is empty: $|V| = |V_L|+1 = |E_L|+1+1 = |E|+1$,

If T_R is nonempty: $|V| = |V_L|+|V_R|+1$
$= |V_L|+|V_R|+3 = |E|+1$,

Inductive Proofs:

- Standard Form
- Strong Form
- Least Element Principal
- Invariant Form
- Structural Induction

Study Bee