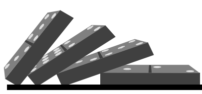
15-251

Great Theoretical Ideas in Computer Science

15-251

Proof Techniques for Computer Scientists



Inductive Reasoning

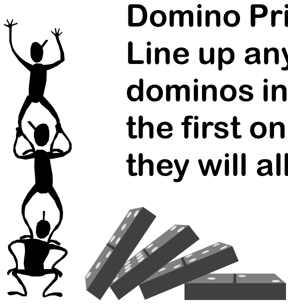
Lecture 2 (August 28, 2008)

Induction

This is the primary way we'll

1. prove theorems
2. construct and define objects

Dominoes



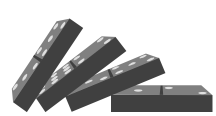
Domino Principle:
Line up any number of dominos in a row; knock the first one over and they will all fall

n dominoes numbered 0 to n-1

$F_k \equiv$ The k^{th} domino falls

If we set them all up in a row then we know that each one is set up to knock over the next one:

For all $0 \leq k < n$:
$F_k \Rightarrow F_{k+1}$



n dominoes numbered 0 to n-1

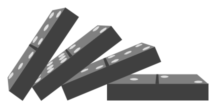
$F_k \equiv$ The k^{th} domino falls

For all $0 \leq k < n-1$:

$F_k \Rightarrow F_{k+1}$

$F_0 \Rightarrow F_1 \Rightarrow F_2 \Rightarrow \ldots F_{n-1}$

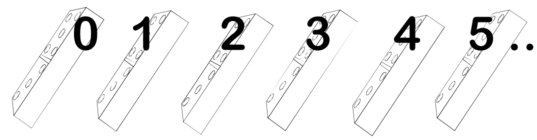
$F_0 \Rightarrow$ All Dominoes Fall



The Natural Numbers

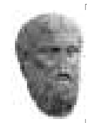
$\mathbb{N} = \{ 0, 1, 2, 3, \ldots \}$

One domino for each natural number:



Plato: The Domino Principle works for an infinite row of dominoes

Aristotle: Never seen an infinite number of anything, much less dominoes.



Plato's Dominoes
One for each natural number

Theorem: An infinite row of dominoes, one domino for each natural number. Knock over the first domino and they all will fall

Proof:

Suppose they don't all fall.
Let $k > 0$ be the lowest numbered domino that remains standing.
Domino $k-1 \geq 0$ did fall, but $k-1$ will knock over domino k.
Thus, domino k must fall and remain standing.
Contradiction.

Mathematical Induction
statements proved instead of dominoes fallen

Infinite sequence of dominoes	Infinite sequence of statements: $S_0, S_1, \ldots$
F_k = "domino k fell"	F_k = "S_k proved"

Establish: 1. F_0

2. For all k, $F_k \Rightarrow F_{k+1}$

Conclude that F_k is true for all k

Inductive Proofs

To Prove $\forall k \in \mathbb{N}, S_k$

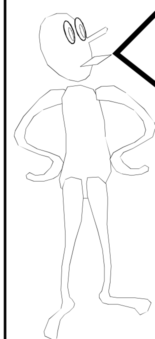
1. Establish "Base Case": S_0
2. Establish that $\forall k, S_k \Rightarrow S_{k+1}$

To prove $\forall k, S_k \Rightarrow S_{k+1}$:
- Assume hypothetically that S_k for any particular k;
- Conclude that S_{k+1}

Theorem?

The sum of the first n odd numbers is n^2

Check on small values:



Theorem?

The sum of the first n odd numbers is n^2

Check on small values:

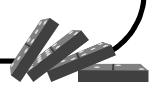
1	= 1
1+3	= 4
1+3+5	= 9
1+3+5+7	= 16

Theorem?

The sum of the first n odd numbers is n^2

The k^{th} odd number is $(2k – 1)$, when $k > 0$

S_n is the statement that:
"$1+3+5+(2k-1)+\ldots+(2n-1) = n^2$"



Establishing that $\forall n \geq 1\ S_n$

S_n = "$1 + 3 + 5 + (2k-1) + . . +(2n-1) = n^2$"

Base Case: S_1 = "$1 = 1^2$" ✓

Induction Hypothesis (I.H.)

for some k, Assume S_k is true ~~for some k~~

Inductive Step: $1+3+5+ \ldots + (2k-1) = k^2$

Add 2k+1 to both sides

$\Rightarrow 1+3+5+ \ldots + 2k-1 + 2k+1 = k^2+2k+1$

$= (k+1)^2$ ☺

$\Rightarrow S_{k+1}$ is true

Establishing that $\forall n \geq 1\ S_n$

S_n = "$1 + 3 + 5 + (2k-1) + . . +(2n-1) = n^2$"

Base Case: S_1

Domino Property:

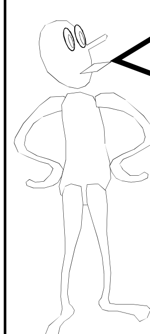
Assume "Induction Hypothesis": S_k

That means:

$1+3+5+\ldots+ (2k-1) \quad = k^2$

$1+3+5+\ldots+ (2k-1)+(2k+1) \quad = k^2 +(2k+1)$

Sum of first k+1 odd numbers = $(k+1)^2$



Theorem

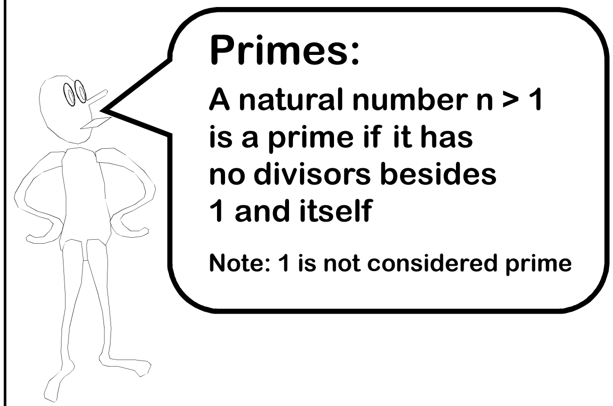
The sum of the first n odd numbers is n^2

Inductive Proofs

To Prove $\forall k \in \mathbb{N}, S_k$

1. Establish "Base Case": S_0
2. Establish that $\forall k, S_k \Rightarrow S_{k+1}$

To prove $\forall k, S_k \Rightarrow S_{k+1}$: Assume hypothetically that S_k for any particular k; Conclude that S_{k+1}



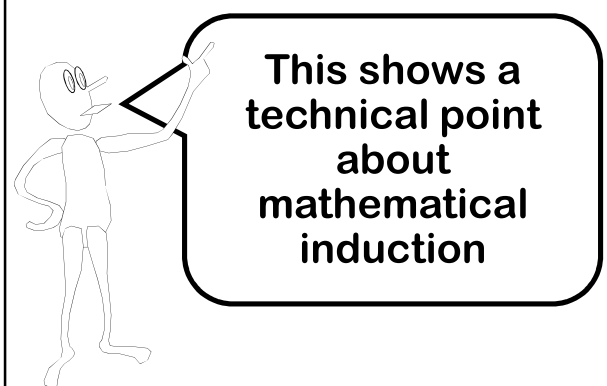
Theorem?

Every natural number n > 1 can be factored into primes

S_n = "n can be factored into primes"

Base case:
2 is prime $\Rightarrow S_2$ is true

How do we use the fact:
S_{k-1} = "k-1 can be factored into primes"
to prove that:
S_k = "k can be factored into primes"



Theorem?

Every natural number > 1 can be factored into primes

A different approach:

Assume 2,3,…,k-1 all can be factored into primes
Then show that k can be factored into primes

Theorem?

Every natural number > 1 can be factored into primes

I.H. 2,3....k-1 can be factored into primes
I.Step: Either k is a prime
or k is composite $k = a \times b$, $a, b < k$
but a can be written as product of primes (by I.H.)
b ______
$K = a \cdot b$ = product of primes

All Previous Induction

To Prove $\forall k, S_k$

Establish Base Case: S_0

Establish Domino Effect:

Assume $\forall j<k, S_j$
use that to derive S_k

All Previous Induction

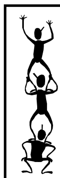
To Prove $\forall k$

Establish Base Case

Sometimes called "Strong Induction"

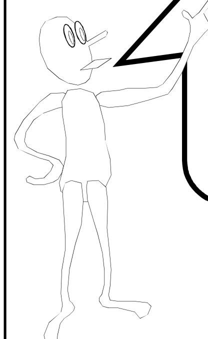
It's really a repackaging of regular induction

"All Previous" Induction



Repackaged As Standard Induction

	Define $T_i = \forall j \leq i, S_j$
Establish Base Case: S_0	Establish Base Case T_0
Establish Domino Effect:	Establish that $\forall k, T_k \Rightarrow T_{k+1}$
Let k be any number Assume $\forall j<k, S_j$	Let k be any number Assume T_{k-1}
Prove S_k	Prove T_k



Method of Infinite Descent

Pierre de Fermat

Show that for any counter-example you can find a smaller one

Now if you choose the "least" counter-example, you'd find a smaller counter-example

This contradicts that you had the "least" counterexample to start with

Technical point: requires that any set of statements (in particular, the counter-examples) has a "least" statement. This is true since we identify statements with the naturals.

Theorem:

Every natural number > 1 can be factored into primes

Let n be a counter-example

Hence n is not prime, so n = ab

If both a and b had prime factorizations, then n would too

Thus a or b is a smaller counter-example

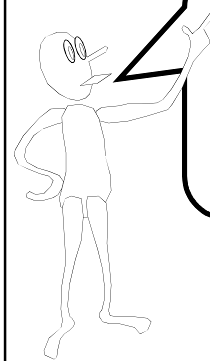
Method of Infinite Descent

Pierre de Fermat

Show that for any counter-example you can find a smaller one

Now if you choose the “least” counter-example, you’d find a smaller counter-example

This contradicts that you had the “least” counterexample to start with

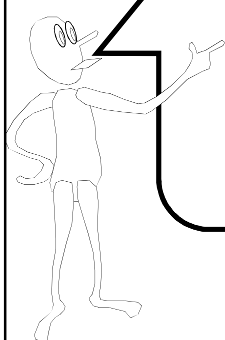


Regular Induction

All-previous Induction

Infinite Descent

And one more way of packaging induction…



Invariants

Invariant (n):
1. Not varying; constant.
2. *Mathematics.* Unaffected by a designated operation, as a transformation of coordinates.

Invariant (n):
3. *Programming.*
   A rule, such as the ordering of an ordered list, that applies throughout the life of a data structure or procedure.
   Each change to the data structure maintains the correctness of the invariant

Invariant Induction

Suppose we have a time varying world state: $W_0, W_1, W_2, \ldots$

Each state change is assumed to come from a list of permissible operations. We seek to prove that statement S is true of all future worlds

Argue that S is true of the initial world

Show that if S is true of some world – then S remains true after one permissible operation is performed

Odd/Even Handshaking Theorem

At any party at any point in time define a person’s parity as ODD/EVEN according to the number of hands they have shaken

Statement:
The number of people of odd parity must be even

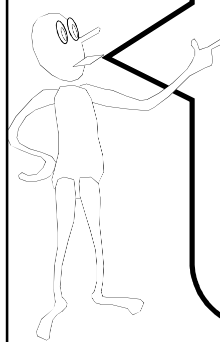
Statement: The number of people of odd parity must be even

Initial case: Zero hands have been shaken at the start of a party, so zero people have odd parity

Invariant Argument:

If 2 people of the same parity shake, they both change and hence the odd parity count changes by 2 – and remains even

If 2 people of different parities shake, then they both swap parities and the odd parity count is unchanged



Inductive reasoning is the high level idea

“Standard” Induction
“All Previous” Induction
“Least Counter-example”
“Invariants”
all just
different packaging

Induction Problem

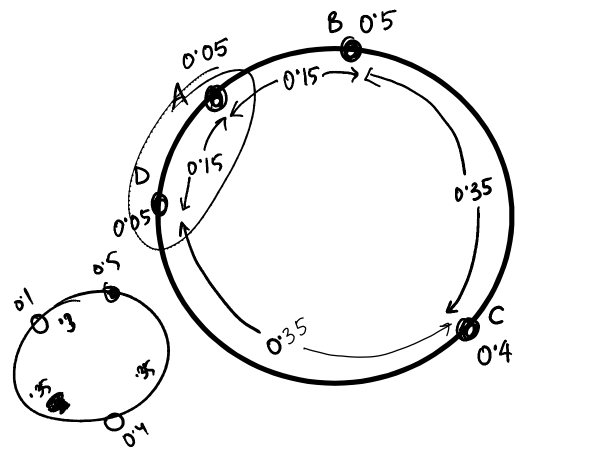
A circular track that is one mile long

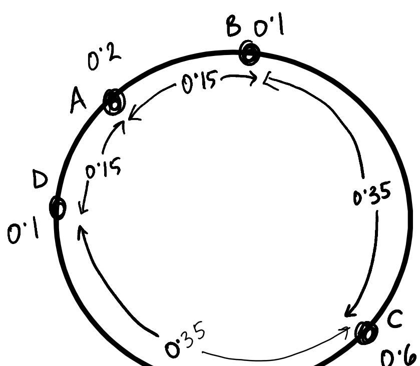
There are $n > 0$ gas stations scattered throughout the track

The combined amount of gas in all gas stations allows a car to travel exactly one mile

The car has a very large tank of gas that starts out empty

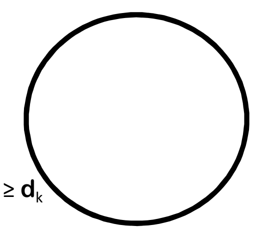
Show that no matter how the gas stations are placed, there is a starting point for the car such that it can go around the track once (clockwise).





$g_1 + g_2 + \ldots + g_n = 1$

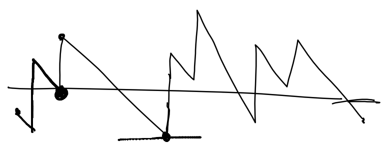
$d_1 + d_2 + \ldots + d_n = 1$



So there is a k such that $g_k \geq d_k$

Remove the gas station (k+1)
and set the gas $g'_k = g_k + g_{k+1}$

By the I.H. there is a good starting point for this new set of (n-1) gas stations and amounts.



One more useful tip…

Here's another problem

Let $A_m = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^m$

Prove that all entries of A_m are at most m. ≥ 1

B.C.: A_1 all entries ≤ 1

I.H. A_m has all entries $\leq m$ $\quad A_m = \begin{pmatrix} p & q \\ r & s \end{pmatrix}$

$A_{m+1} = \begin{pmatrix} p & q \\ r & s \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} p & p+q \\ r & r+s \end{pmatrix}$

So, is it false?

$A_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$

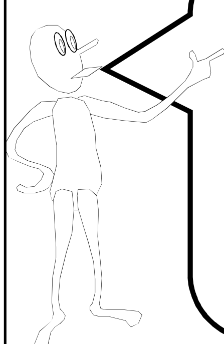
$A_3 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}$

$A_4 = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 4 \\ 0 & 1 \end{pmatrix}$

Prove a stronger statement!

Claim: $A_m = \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix}$

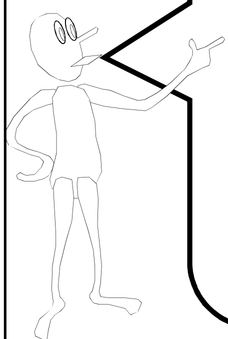
Corollary: All entries of A_m are at most m.



Often, to prove a statement inductively

you may have to prove a stronger statement first!

Using induction to define mathematical objects



Induction is also how we can define and construct our world

So many things, from buildings to computers, are built up stage by stage, module by module, each depending on the previous stages

Inductive Definition

Example

Initial Condition, or Base Case:
F(0) = 1

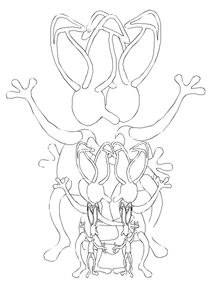
Inductive Rule:
For n > 0, F(n) = F(n-1) + F(n-1)

Inductive definition of the powers of 2!

n	0	1	2	3	4	5	6	7
F(n)	1	2	4	8	16	32	64	128

Leonardo Fibonacci

In 1202, Fibonacci proposed a problem about the growth of rabbit populations



Rabbit Reproduction

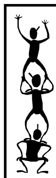
A rabbit lives forever

The population starts as single newborn pair

Every month, each productive pair begets a new pair which will become productive after 2 months old

F_n= # of rabbit pairs at the beginning of the n^{th} month

month	1	2	3	4	5	6	7
rabbits	1	1	2	3	5	8	13

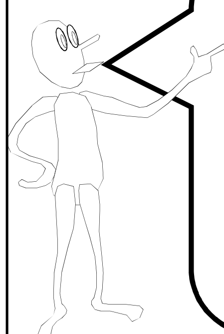


Fibonacci Numbers

month	1	2	3	4	5	6	7
rabbits	1	1	2	3	5	8	13

Stage 0, Initial Condition, or Base Case:
Fib(1) = 1; Fib (2) = 1

Inductive Rule:
For n>3, Fib(n) = Fib(n-1) + Fib(n-2)



If you define a function inductively, it is usually easy to prove it's properties using induction!

Example

Theorem$^?$: $F_1 + F_2 + \ldots + F_n = F_{n+2} - 1$

Example

Theorem$^?$: $F_1 + F_2 + \ldots + F_n = F_{n+2} - 1$

Example

Theorem$^?$: $F_1 + F_2 + \ldots + F_n = F_{n+2} - 1$

Base cases: $n=1$, $F_1 = F_3 - 1$

$n=2$, $F_1 + F_2 = F_4 - 1$

I.H.: True for all $n < k$.

Induction Step: $F_1 + F_2 + \ldots + F_k$

$= (F_1 + F_2 + \ldots + F_{k-1}) + F_k$

$= (F_{k+1} - 1) + F_k$ (by I.H.)

$= F_{k+2} - 1$ (by defn.)

Another Example

$T(1) = 1$
$T(n) = 4T(n/2) + n$

Notice that T(n) is inductively defined only for positive powers of 2, and undefined on other values

$T(1) = 1$ $T(2) = 6$ $T(4) = 28$ $T(8) = 120$

Guess a closed-form formula for T(n)

Guess: $G(n) = 2n^2 - n$

Inductive Proof of Equivalence

Base Case: $G(1) = 1$ and $T(1) = 1$

Induction Hypothesis:
$T(x) = G(x)$ for $x < n$

Hence: $T(n/2) = G(n/2) = 2(n/2)^2 - n/2$

$T(n) = 4\,T(n/2) + n$

$= 4\,G(n/2) + n$

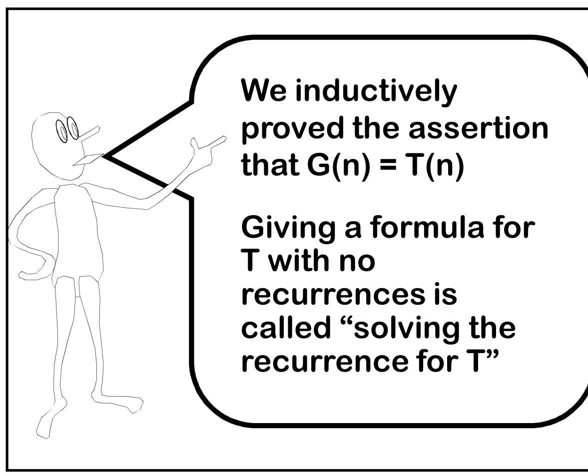
$= 4\,[2(n/2)^2 - n/2] + n$

$= 2n^2 - 2n + n$

$= 2n^2 - n$

$= G(n)$

$\boxed{G(n) = 2n^2 - n \qquad T(1) = 1 \qquad T(n) = 4T(n/2) + n}$



Technique 2

Guess Form, Calculate Coefficients

$\boxed{T(1) = 1, T(n) = 4\,T(n/2) + n}$

Guess: $T(n) = an^2 + bn + c$ ⟵ $(b+1)\,4 + 3c = 0$
for some a,b,c $(b+1)\,8 + 3c = 0$

Calculate: $T(1) = 1$, so $a + b + c = 1$ $(b+1)\,16 + 3c = 0$

$T(n) = 4\,T(n/2) + n$

$\cancel{an^2} + bn + c = 4\,[a(n/2)^2 + b(n/2) + c] + n$

$= \cancel{an^2} + 2bn + 4c + n$

$(b+1)n + 3c = 0$

Therefore: $b = -1$ $c = 0$ $a = 2$

Induction can arise in unexpected places

The Lindenmayer Game

Alphabet: {a,b}

Start word: a

Productions Rules:

Sub(a) = ab Sub(b) = a

$\text{NEXT}(w_1\, w_2 \ldots w_n) = \text{Sub}(w_1)\,\text{Sub}(w_2) \ldots \text{Sub}(w_n)$

Time 1: a
Time 2: ab
Time 3: aba
Time 4: abaab
Time 5: abaababa

How long are the strings at time n?

FIBONACCI(n)

The Koch Game

Alphabet: { F, +, - }

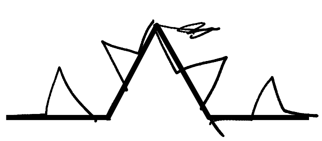
Start word: F

Productions Rules: Sub(F) = F+F--F+F
Sub(+) = +
Sub(-) = -

$\text{NEXT}(w_1\, w_2 \ldots w_n) = \text{Sub}(w_1)\,\text{Sub}(w_2) \ldots \text{Sub}(w_n)$

Time 0: F
Time 1: F+F--F+F
Time 2: F+F--F+F+F+F--F+F--F+F--F+F+F+F--F+F

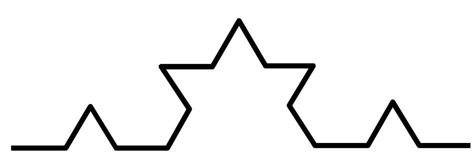
The Koch Game



F+F--F+F

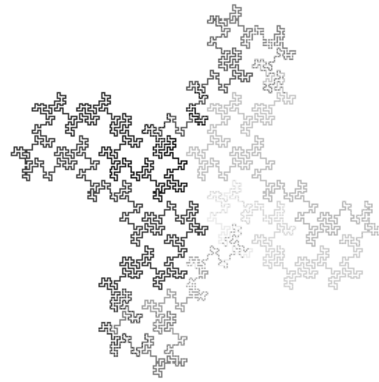
Visual representation:
F draw forward one unit
\+ turn 60 degree left
\- turn 60 degrees right

The Koch Game



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



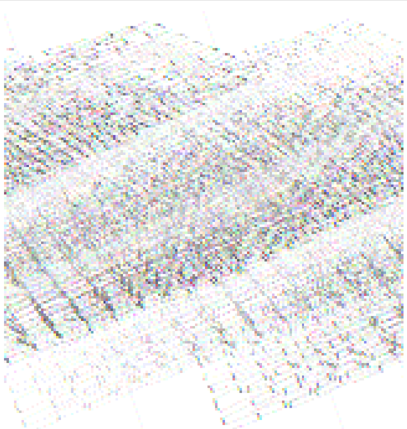
Dragon Game

Sub(X) = X+YF+ **Sub(Y) = -FX-Y**

Hilbert Game

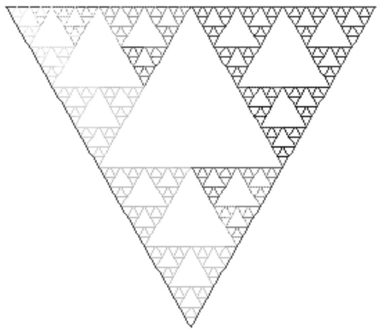
Sub(L) = +RF-LFL-FR+
Sub(R) = -LF+RFR+FL-

Note: Make 90 degree turns instead of 60 degrees



Peano-Gossamer Curve

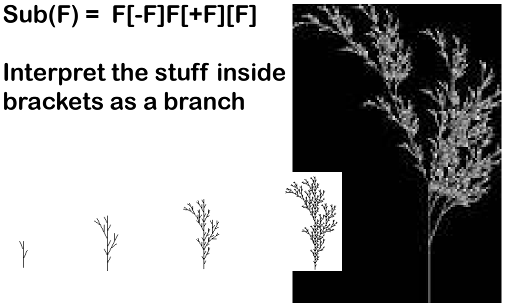
Sierpinski Triangle

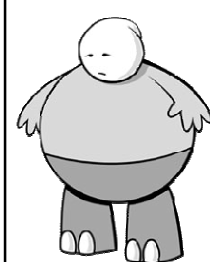


Lindenmayer (1968)

Sub(F) = F[-F]F[+F][F]

Interpret the stuff inside brackets as a branch





Here's What You Need to Know…

Inductive Proof
- Standard Form
- All Previous Form
- Least-Counter Example Form
- Invariant Form

Strengthening the Inductive Claim

Inductive Definition
- Recurrence Relations
- Fibonacci Numbers
- Guess and Verify