Great Theoretical Ideas In Computer Science

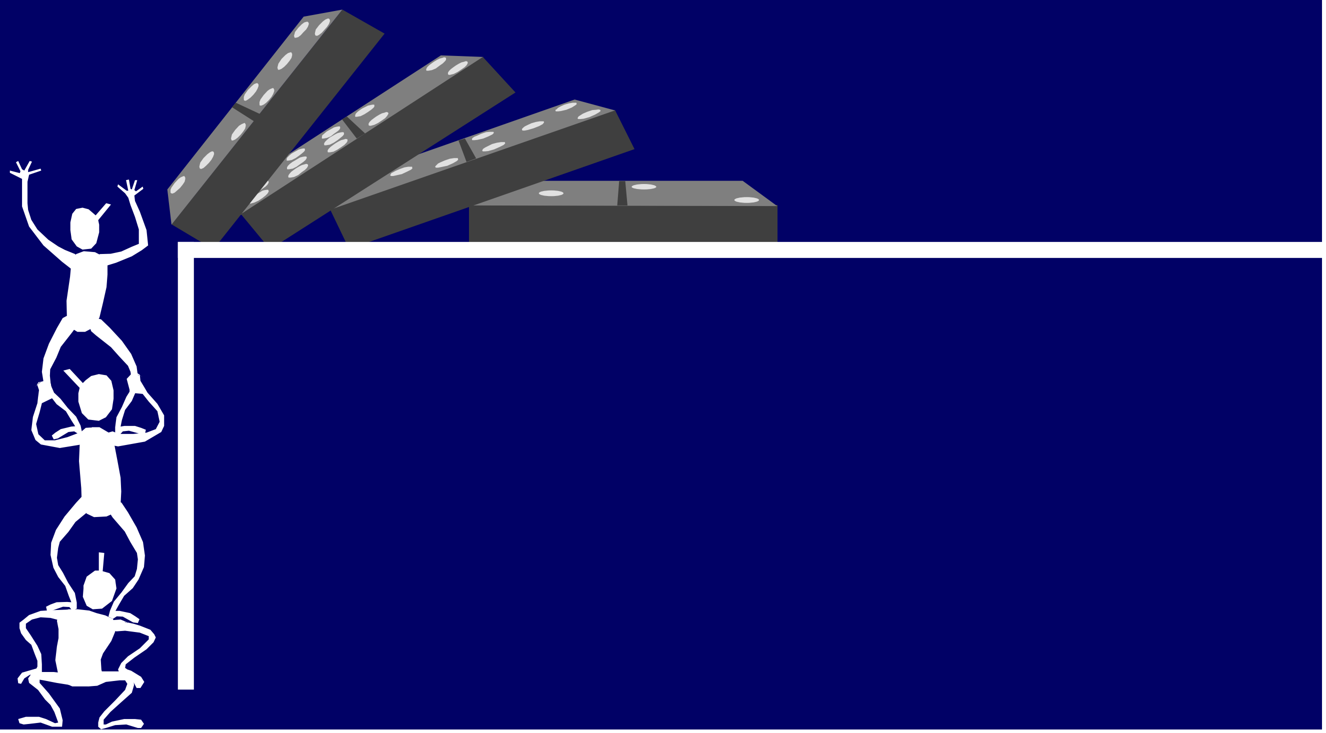
Steven Rudich

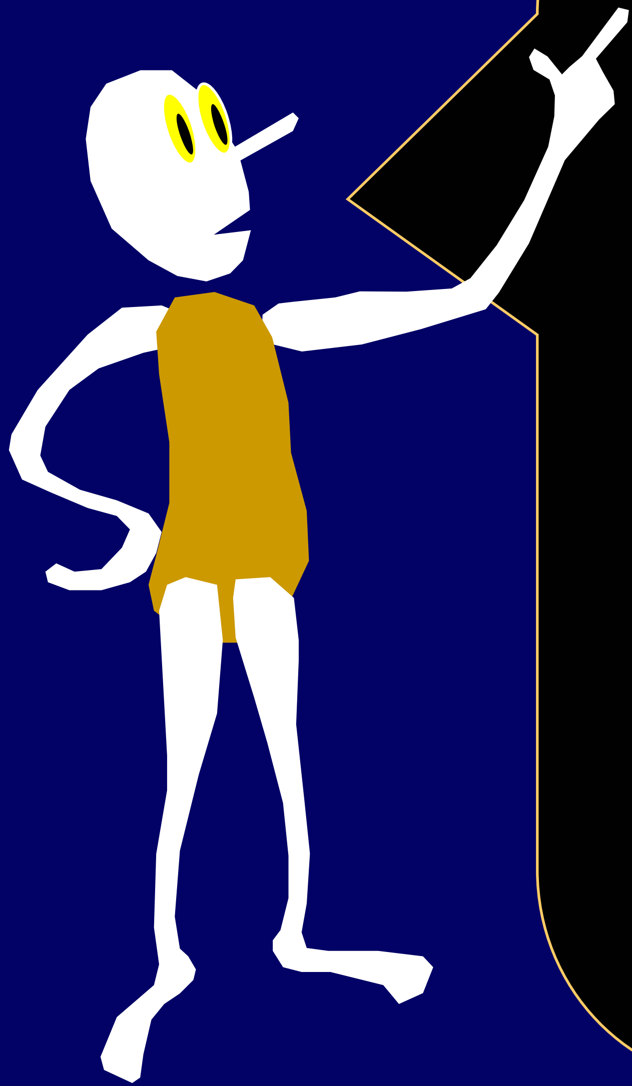
CS 15-251 Spring 2004

Lecture 4 Jan 22, 2004

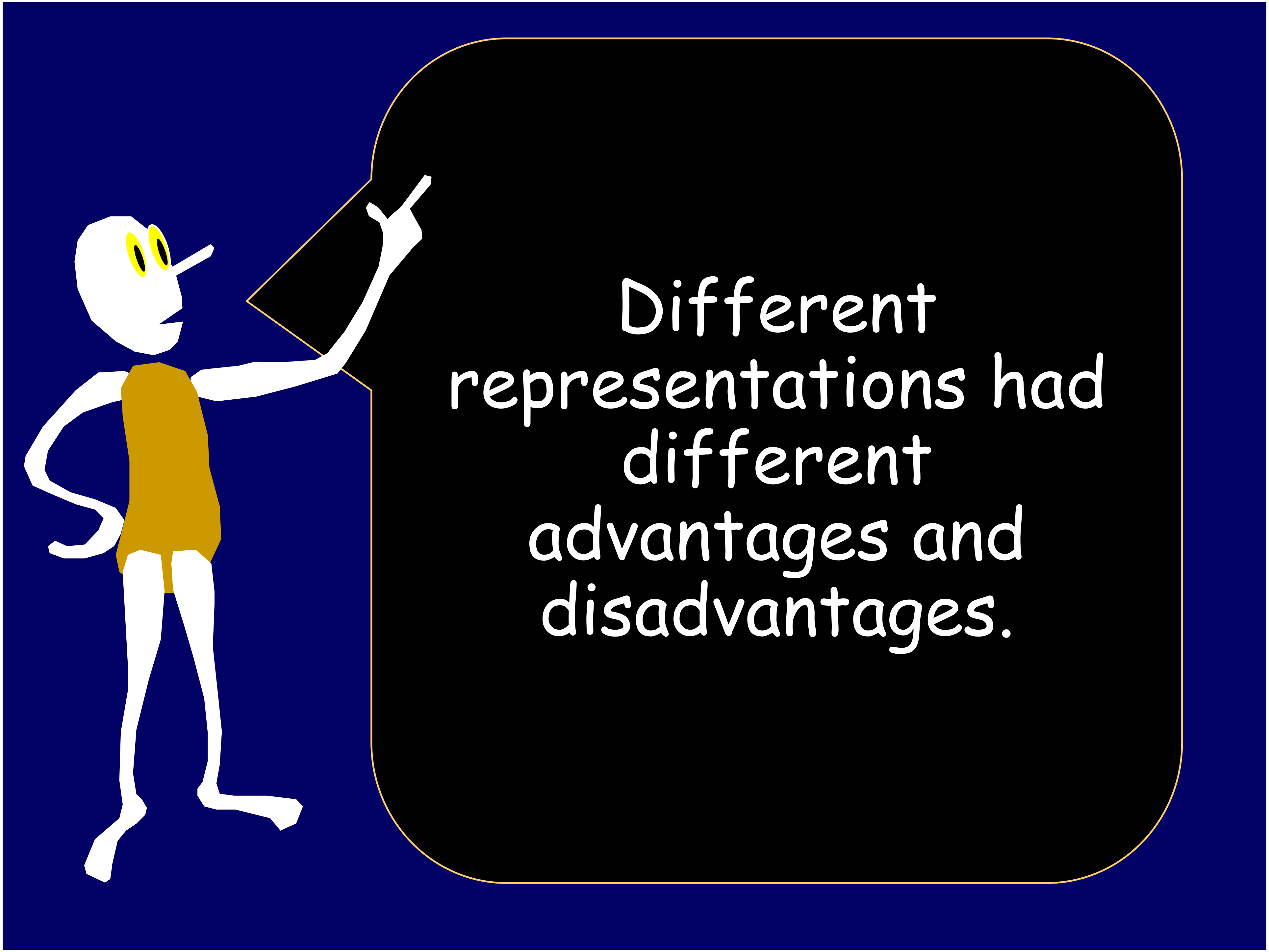
Carnegie Mellon University

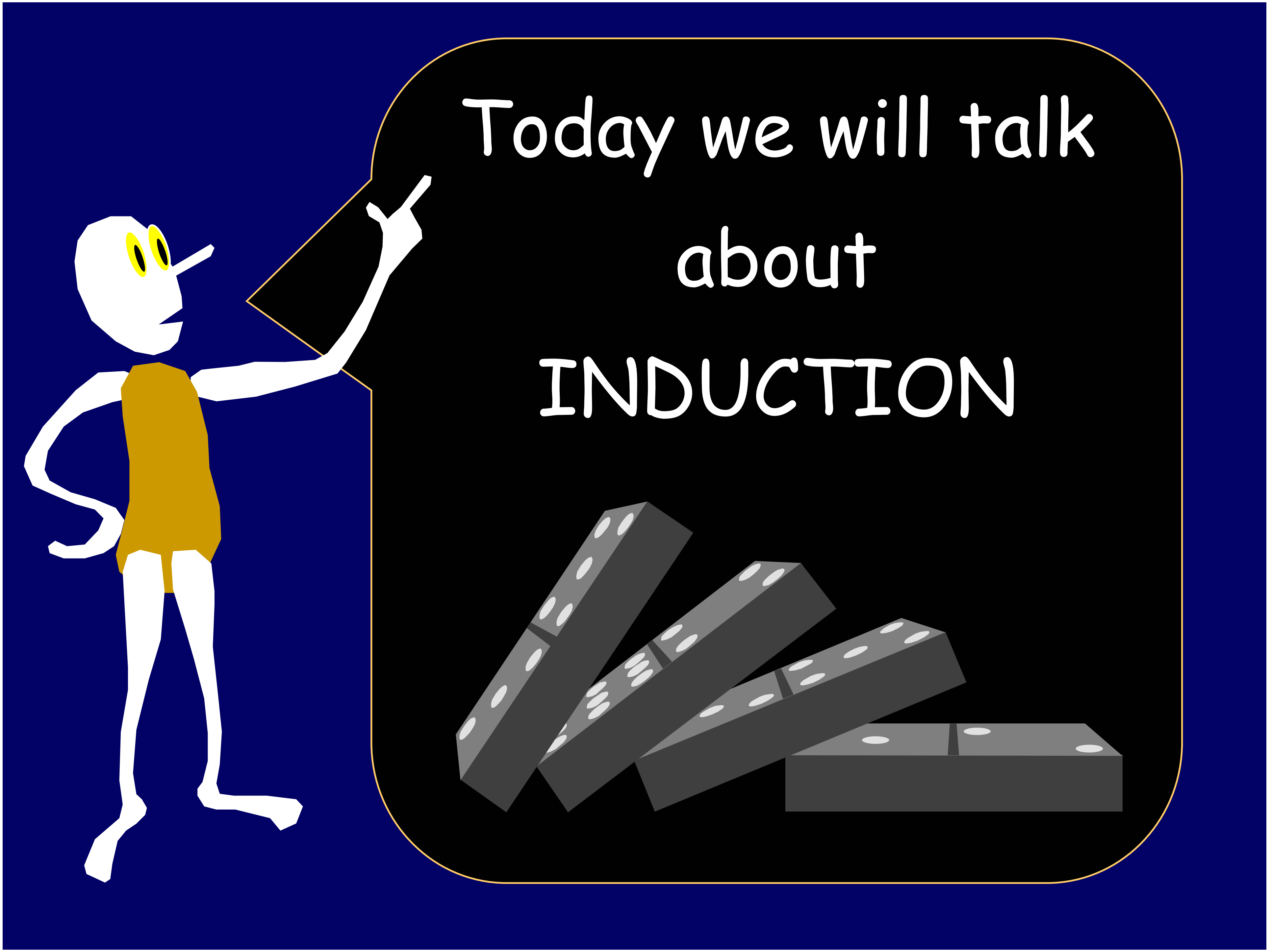
Induction: One Step At A Time

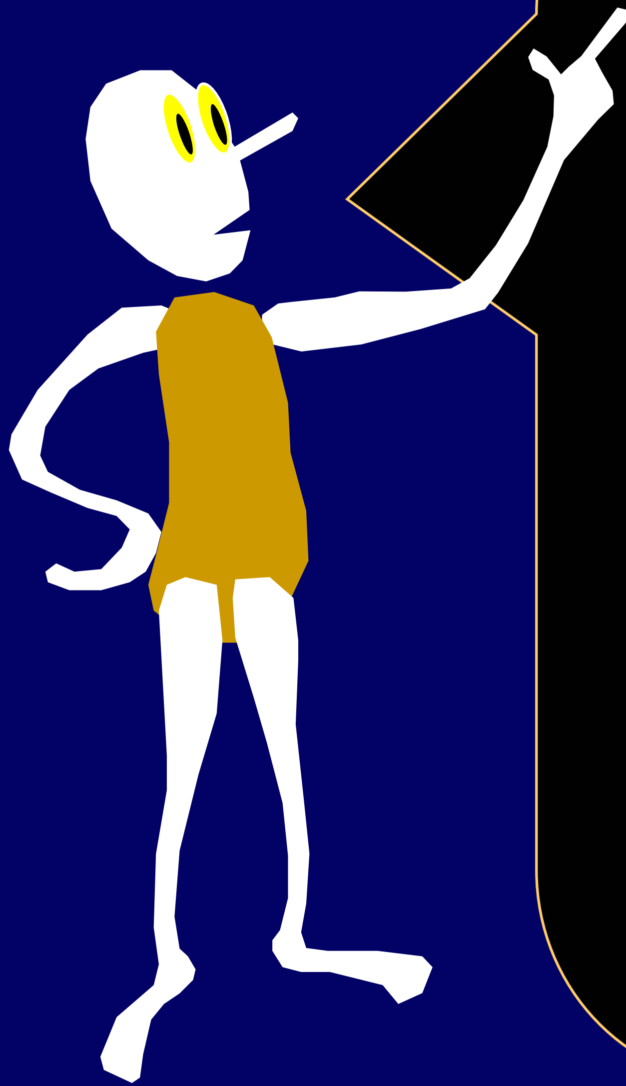




Last time we talked about different ways to represent numbers: unary, binary, decimal, base b, plus/minus binary, Egyptian binary . . .

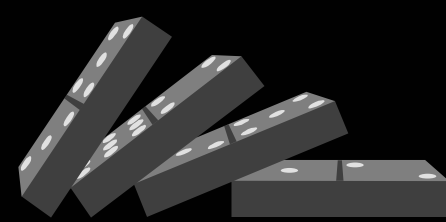

Different representations had different advantages and disadvantages.

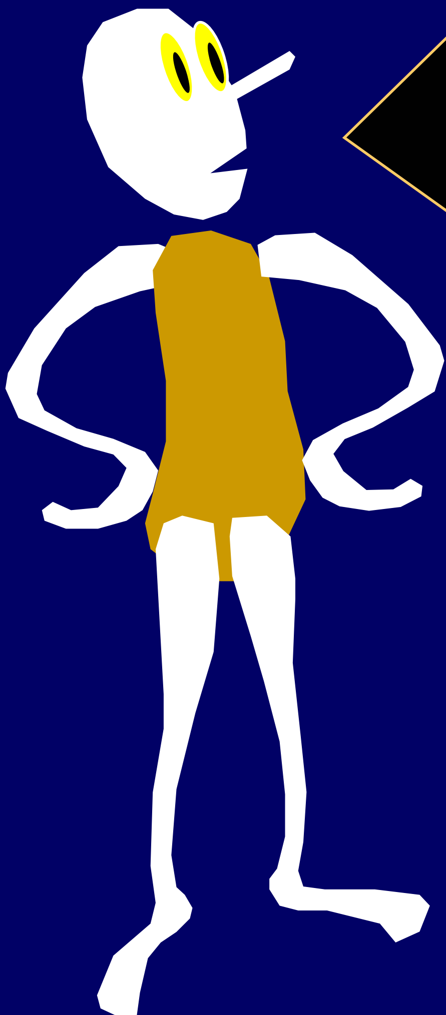

Today we will talk about INDUCTION



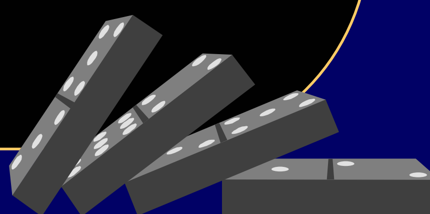
Induction is the primary way we:

1. Prove theorems
2. Construct and define objects



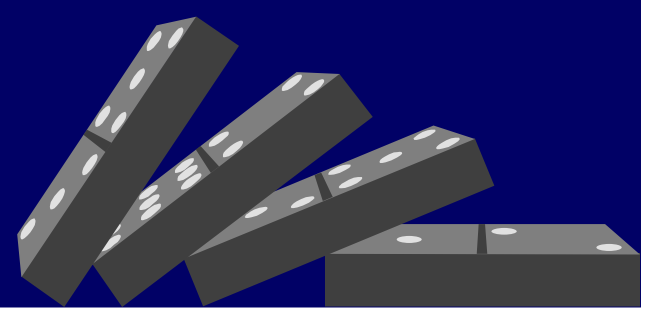


Representing a problem or object inductively is one of the most fundamental abstract representations.



Let’s start with dominoes

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



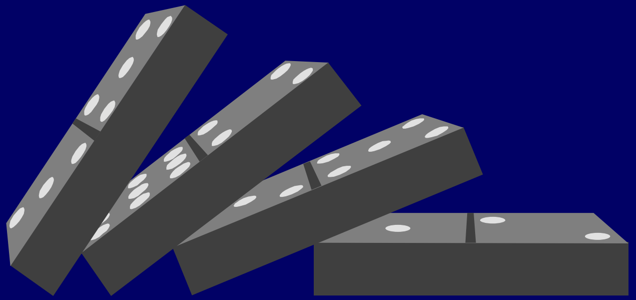
n dominoes numbered 1 to n

$F_k \equiv$ The k^{th} domino will fall

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 $1 \leq k < n$:

$$F_k \Rightarrow F_{k+1}$$



n dominoes numbered 1 to n

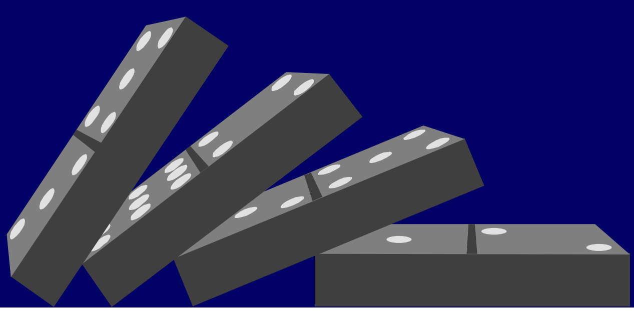
$F_k \equiv$ The k^{th} domino will fall

For all $1 \leq k < n$:

$$F_k \Rightarrow F_{k+1}$$

$F_1 \Rightarrow F_2 \Rightarrow F_3 \Rightarrow \ldots$

$F_1 \Rightarrow$ All Dominoes Fall



n dominoes numbered 0 to n-1

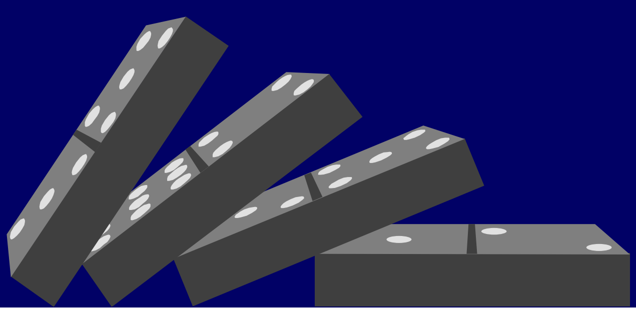
$F_k \equiv$ The k^{th} domino will fall

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

$$F_k \Rightarrow F_{k+1}$$

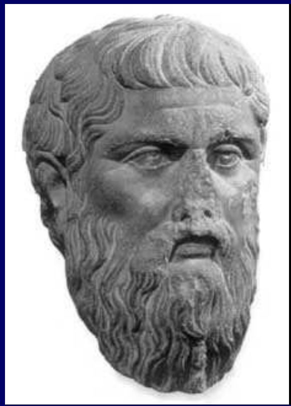
$F_0 \Rightarrow F_1 \Rightarrow F_2 \Rightarrow \ldots$

$F_0 \Rightarrow$ All Dominoes Fall



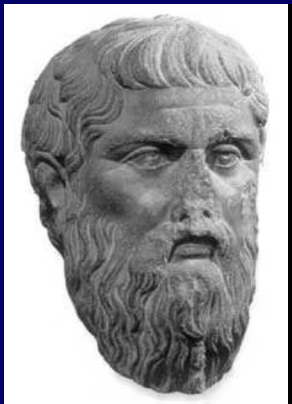
The Natural Numbers

$N = \{ 0, 1, 2, 3, . . . \}$



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

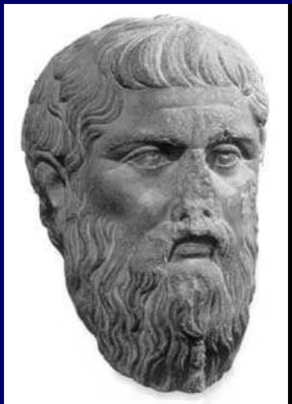
Aristotle: Never seen an infinite number of anything, much less dominoes.



Plato's Dominoes
One for each natural number

An infinite row, 0, 1, 2, … of dominoes, one domino for each natural number. Knock the first domino over 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.



The Infinite Domino Principle

$F_k \equiv$ The k^{th} domino will fall

Assume we know that

for every natural number k,

$$F_k \Rightarrow F_{k+1}$$

$F_0 \Rightarrow F_1 \Rightarrow F_2 \Rightarrow \ldots$

$F_0 \Rightarrow$ All Dominoes Fall

Mathematical Induction: statements proved instead of dominoes fallen

Infinite sequence of dominoes.

$F_k \equiv$ domino k fell

Infinite sequence of statements: $S_0, S_1, \ldots$

$F_k \equiv S_k$ 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 Proof / Reasoning
To Prove $\forall k, S_k$

Establish "Base Case": S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

$\forall k, S_k \Rightarrow S_{k+1}$

<u>Assume</u> hypothetically that S_k for *any* particular k;

<u>Conclude</u> that S_{k+1}

Inductive Proof / Reasoning
To Prove $\forall k, S_k$

Establish "Base Case": S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

$$\forall k, S_k \Rightarrow S_{k+1} \begin{cases} \text{\underline{"Induction Hypothesis"} } S_k \\ \text{\underline{Use} I.H. to show } S_{k+1} \end{cases}$$

Inductive Proof / Reasoning
To Prove $\forall k \geq b, S_k$

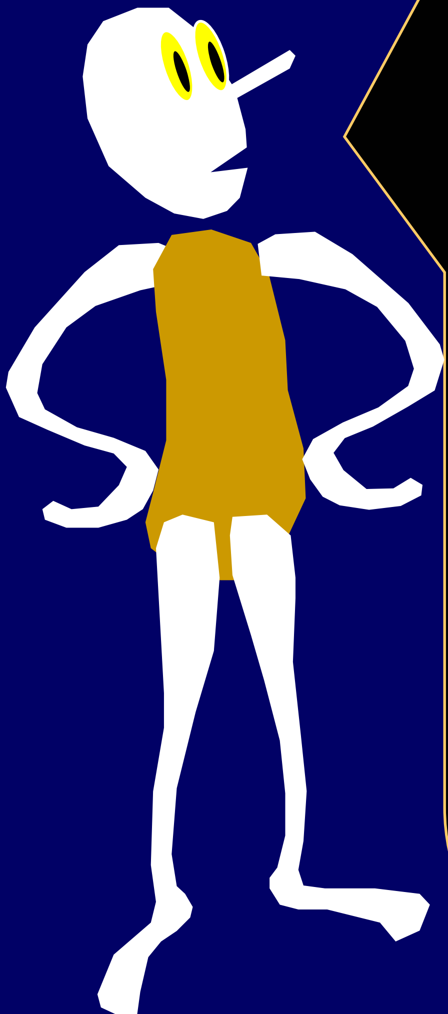
Establish "Base Case": S_b

Establish that $\forall k \geq b, S_k \Rightarrow S_{k+1}$

Assume $k \geq b$

Assume "Inductive Hypothesis": S_k

Prove that S_{k+1} follows



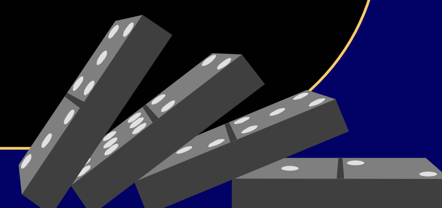
We already know that

$\forall n,$

$\Delta_n = 1 + 2 + 3 + \ldots + n\text{-}1 + n$
$= n(n+1)/2.$

Let's prove it by induction:

Let $S_n \equiv$
"$\Delta_n = n(n+1)/2$"



$S_n \equiv$ "$\Delta_n = n(n+1)/2$"

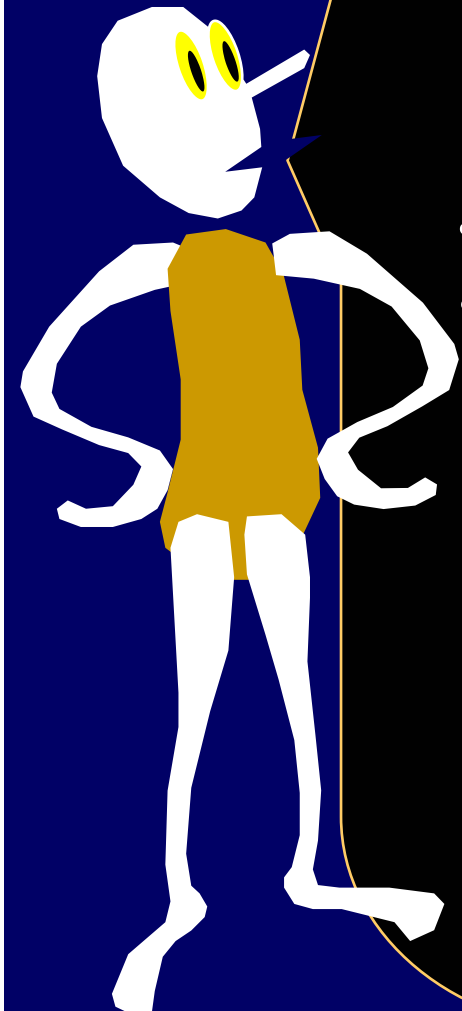
Use induction to prove $\forall k \geq 0, S_k$

<u>Establish "Base Case":</u> S_0. Δ_0=The sum of the first 0 numbers = 0. Setting n=0 the formula gives 0(0+1)/2 = 0.

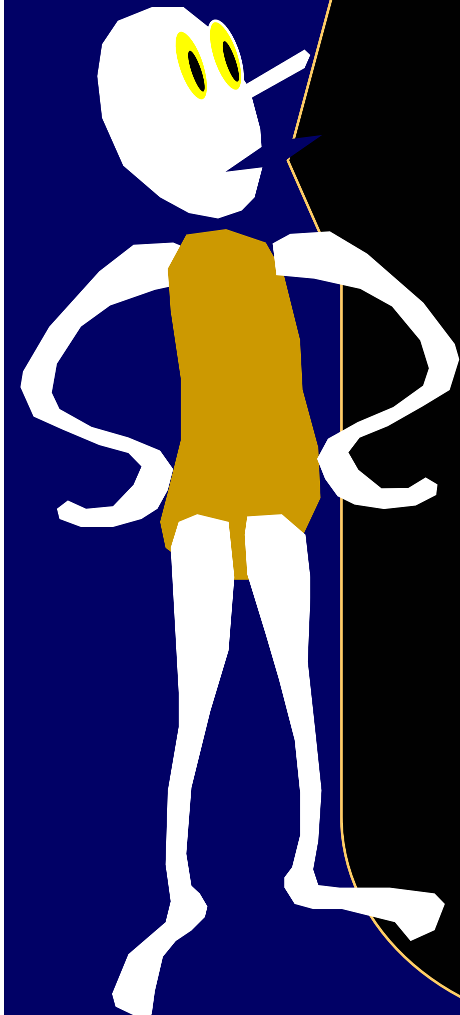
<u>Establish that $\forall k \geq 0, S_k \Rightarrow S_{k+1}$</u>

"Inductive Hypothesis" S_k: $\Delta_k = k(k+1)/2$

$$\begin{aligned}\Delta_{k+1} &= \Delta_k + (k+1) \\ &= k(k+1)/2 + (k+1) \quad \text{[Using I.H.]} \\ &= (k+1)(k+2)/2 \quad \text{[which proves } S_{k+1}\text{]}\end{aligned}$$



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.


Well, almost always

Inductive Definition Of Functions

Stage 0, Initial Condition, or Base Case:

Declare the value of the function on some subset of the domain.

Inductive Rules

Define new values of the function in terms of previously defined values of the function

F(x) is defined if and only if it is implied by finite iteration of the rules.

Inductive Definition
Recurrence Relation for F(X)

Initial Condition, or Base Case:

F(0) = 1

Inductive Rule

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

n	0	1	2	3	4	5	6	7
F(n)	1							

Inductive Definition
Recurrence Relation for F(X)

Initial Condition, or Base Case:

$F(0) = 1$

Inductive Rule

For $n>0$, $F(n) = F(n-1) + F(n-1)$

n	0	1	2	3	4	5	6	7
F(n)	1	2						

Inductive Definition
Recurrence Relation for F(X)

Initial Condition, or Base Case:

$F(0) = 1$

Inductive Rule

For $n>0$, $F(n) = F(n-1) + F(n-1)$

n	0	1	2	3	4	5	6	7
F(n)	1	2	4					

Inductive Definition
Recurrence Relation for F(X)

Initial Condition, or Base Case:

$F(0) = 1$

Inductive Rule

For $n>0$, $F(n) = F(n-1) + F(n-1)$

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

Inductive Definition
Recurrence Relation for $F(X) = 2^X$

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

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

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

Inductive Definition
Recurrence Relation

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

Inductive Rule
For n>1, F(n) = F(n/2) + F(n/2)

n	0	1	2	3	4	5	6	7
F(n)		1						

Inductive Definition
Recurrence Relation

Initial Condition, or Base Case:

$F(1) = 1$

Inductive Rule

For $n>1$, $F(n) = F(n/2) + F(n/2)$

n	0	1	2	3	4	5	6	7
F(n)		1	2					

Inductive Definition
Recurrence Relation

Initial Condition, or Base Case:

$F(1) = 1$

Inductive Rule

For $n>1$, $F(n) = F(n/2) + F(n/2)$

n	0	1	2	3	4	5	6	7
F(n)		1	2		4			

Inductive Definition
Recurrence Relation

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

Inductive Rule
For $n>1$, $F(n) = F(n/2) + F(n/2)$

n	0	1	2	3	4	5	6	7
F(n)	%	1	2	%	4	%	%	%

Inductive Definition
Recurrence Relation
F(X) = X for X a whole power of 2.

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

Inductive Rule
For $n>1$, $F(n) = F(n/2) + F(n/2)$

n	0	1	2	3	4	5	6	7
F(n)	%	1	2	%	4	%	%	%

Base Case: $\forall x \in N\ P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2								
3								

Base Case: $\forall x \in N\ P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0	0							
1	1							
2	2							
3	3							

Base Case: $\forall x \in N \; P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0	0	1						
1	1	2						
2	2	3						
3	3	4						

Base Case: $\forall x \in N \; P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0	0	1	2					
1	1	2	3					
2	2	3	4					
3	3	4	5					

Base Case: $\forall x \in \mathrm{N}\ P(X,0) = X$

Inductive Rule:

$\forall x,y \in \mathrm{N},\ y>0,\ P(x,y) = P(x,y-1) + 1$

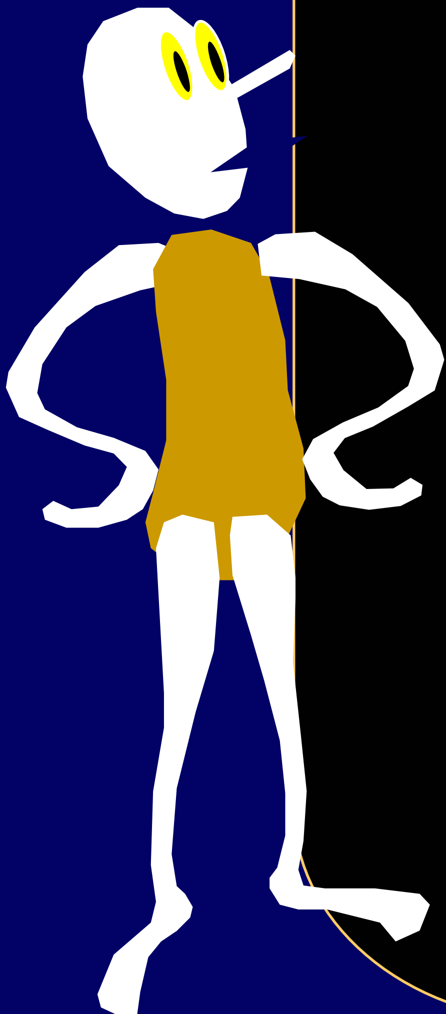
P(x,y)	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	8
2	2	3	4	5	6	7	8	9
3	3	4	5	6	7	8	9	10

Base Case: $\forall x \in N \; P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

X+Y	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	8
2	2	3	4	5	6	7	8	9
3	3	4	5	6	7	8	9	10



Definition of P:

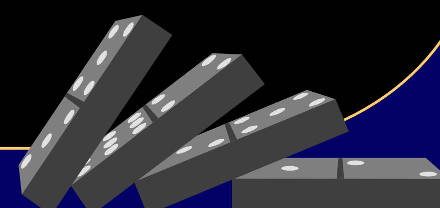
$\forall x \in N\ P(X,0) = X$

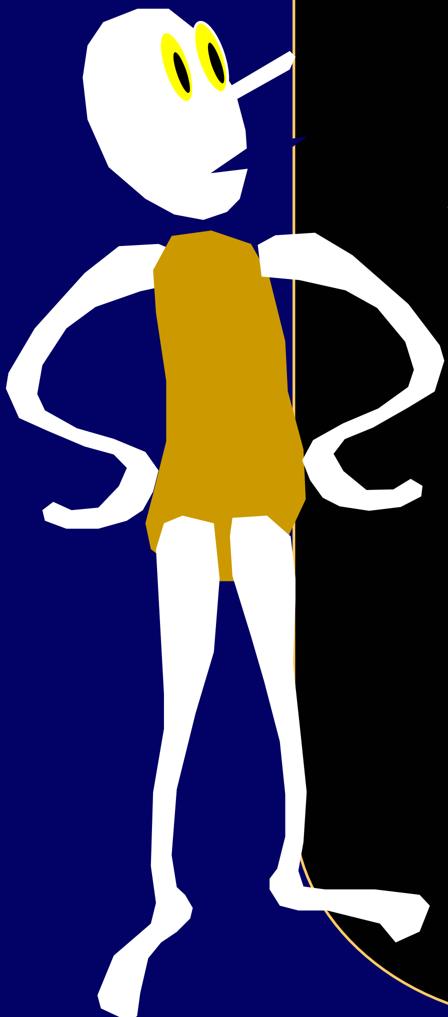
$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

Peano's Definition of "+":

$x + 0 = x$

$x + Sy = S(x + y)$

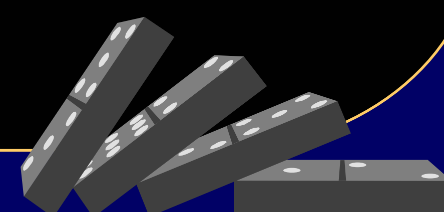


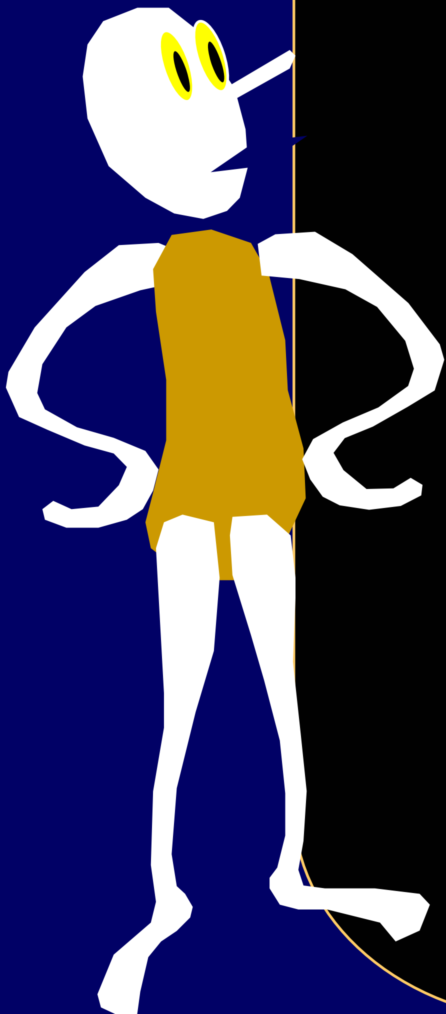


Definition of P:

$$\forall x \in N\ P(X,0) = X$$
$$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$$

Any inductive definition can be translated into a program. The program simply calculates from the base cases on up.



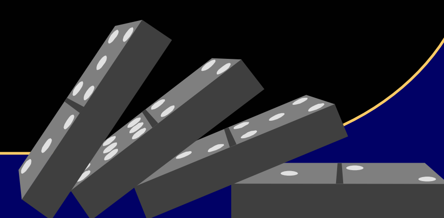


Definition of P:

$\forall x \in N \; P(X,0) = X$

$\forall x,y \in N,\; y>0,\; P(x,y) = P(x,y-1) + 1$

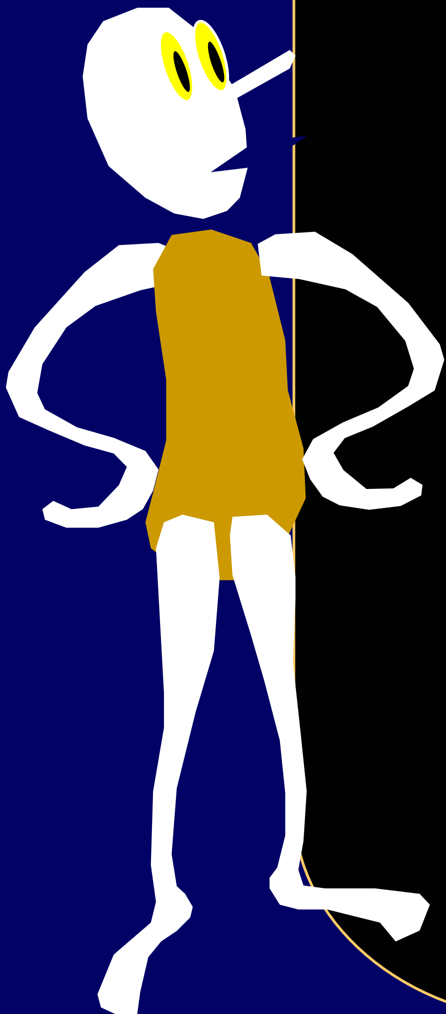
What would be the <u>bottom up</u> implementation of P?



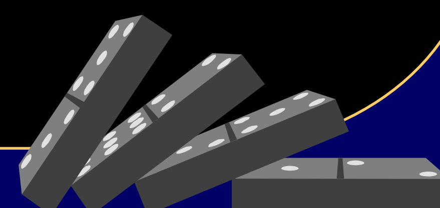
Bottom-Up Program for P

```
For k = 0 to 3
        P(k,0)=k
For j = 1 to 7
        For k = 0 to 3
                P(k,j) = P(k,j-1) + 1
```

P(x,y)	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	8
2	2	3	4	5	6	7	8	9
3	3	4	5	6	7	8	9	10



Suppose we wanted to know P(2,3) in particular, but we had not yet done any calculation.



Base Case: $\forall x \in N\ P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2				?				
3								

Base Case: $\forall x \in N\ P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2			?	?				
3								

Base Case: $\forall x \in N$ $P(X,0) = X$

Inductive Rule:

$\forall x,y \in N$, $y>0$, $P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2		?	?	?				
3								

Base Case: $\forall x \in N \; P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\; y>0,\; P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2	?	?	?	?				
3								

Base Case: $\forall x \in N\ P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2	2	?	?	?				
3								

Base Case: $\forall x \in N\ P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2	2	3	?	?				
3								

Base Case: $\forall x \in N \; P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\; y>0,\; P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2	2	3	4	?				
3								

Base Case: $\forall x \in N\ P(X,0) = X$

Inductive Rule:

$\forall x,y \in N,\ y>0,\ P(x,y) = P(x,y-1) + 1$

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2	2	3	4	5				
3								

Procedure P(x,y):

If y=0 return x

Otherwise return P(x,y-1)+1;

Top Down

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2	2	3	4	5				
3								

Recursive Programming

Procedure P(x,y):

If y=0 return x

Otherwise return P(x,y-1)+1;

P(x,y)	0	1	2	3	4	5	6	7
0								
1								
2	2	3	4	5				
3								

Inductive Definition:

$\forall x \in \mathbb{N}\ P(X,0) = X$

$\forall x,y \in \mathbb{N},\ y>0,\ P(x,y) = P(x,y-1) + 1$

Bottom-Up, Iterative Program:

```
For k = 0 to 3
        P(k,0)=k
For j = 1 to 7
        For k = 0 to 3
                P(k,j) = P(k,j-1) + 1
```

Top-Down, Recursive Program:

```
        Procedure P(x,y):
                If y=0 return x
                Otherwise return P(x,y-1)+1;
```

"God Made The Naturals.
Everything Else Is The Work Of Man."

Kronecker

"God Made <u>Induction</u> On The Naturals.
Everything Else Is The Work Of Man."

Peano

Giuseppe Peano [1889] Axiom's For $\mathbb{N}$

A1. $Sx \neq 0$

A2. $[Sx = Sy] \Rightarrow [x=y]$

A3. For any proposition $P(x)$ where $x \in \mathbb{N}$.
Mathematical Induction Applies To P:

$[P(0) \quad \text{and} \quad \forall\, x \in \mathbb{N},\, P(x) \Rightarrow P(Sx)]$

$\Rightarrow \quad \forall\, x\, P(x)$

Giuseppe Peano [1889] Axiom's For $\mathbb{N}$

A1. $Sx \neq 0$

A2. $[Sx = Sy] \Rightarrow [x=y]$

A3. For any proposition $P(x)$ where $x \in \mathbb{N}$. Mathematical Induction Applies To P:

$$[P(0) \text{ and } \forall\, x \in \mathbb{N},\ P(x) \Rightarrow P(Sx)] \Rightarrow \forall\, x\, P(x)$$

Inductive Definition of +:

$x + 0 = x$

$x + Sy = S(x + y)$

Let's prove the Commutativity of addition:

$x + y = y + x$

Lemma: $0 + x = x$

Let $P(x)$ be the proposition that

"$0 + x = x$"

$P(0)$ is "$0 + 0 = 0$"

Assume $P(x)$: "$0 + x = x$"

Show $P(Sx)$:

$0+Sx = S(0+x) = S(x) = Sx$

Lemma: $Sx + y = S(x+y)$

Let $P(y)$ be the proposition that
"$\forall\, x,\ Sx + y = S(x+y)$"

$P(0)$ is "$\forall\, x,\ Sx + 0 = S(x+0) = Sx$"

Assume $P(y)$: "$\forall\, x,\ Sx+y = S(x+y)$"

Show $P(Sy)$:

$Sx+Sy = S(Sx+y) = S(S(x+y)\ ) = S(x+Sy)$

Theorem: Commutative Property Of Addition: $x + y = y + x$

Let P(y) be the proposition that

"$\forall\ x, x + y = y + x$"

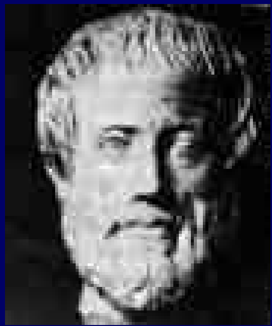
P(0) is "$\forall\ x, x + 0 = 0 + x$"

Assume P(y): "$\forall\ x, x + y = y + x$"

Show P(Sy):

$x+Sy = S(x+y) = S(y+x) = Sy + x$


Let's return our attention to the technique of inductive proofs.



Aristotle's Contrapositive

Let S be a sentence of the form "$A \Rightarrow B$".

The <u>Contrapositive</u> of S is the sentence "$\neg B \Rightarrow \neg A$".

$A \Rightarrow B$: When A is true, B is true.

$\neg B \Rightarrow \neg A$: When B is false, A is false.

Aristotle's Contrapositive

Logically equivalent:

A	B	"$A \Rightarrow B$"	"$\neg B \Rightarrow \neg A$".
False	False	True	True
False	True	True	True
True	False	False	False
True	True	True	True

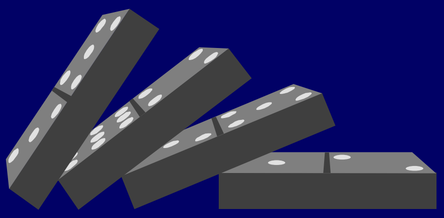
Advice from the master.

To prove S, it is often easier to prove the contrapositive of S.

Contrapositive Dominos

Suppose there is a least domino k that does not fall.

If k did not fall, then k-1 did not fall. k-1 would be a smaller, standing domino, contradicting our assumption.



Contrapositive or Least Counter-Example Induction to Prove $\forall k, S_k$

Establish "Base Case": S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

Let k>0 be the least number such that S_k is false.

Prove that $\boxed{\neg S_k \Rightarrow \neg S_{k-1}}$

[Contradiction of k being the least counter-example]

“Strong” Induction
To Prove $\forall k, S_k$

Establish “Base Case”: S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

Let k be any natural number.

Assume $\forall j<k, S_j$

Prove S_k

All Previous Induction
To Prove $\forall k, S_k$

Establish "Base Case": S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

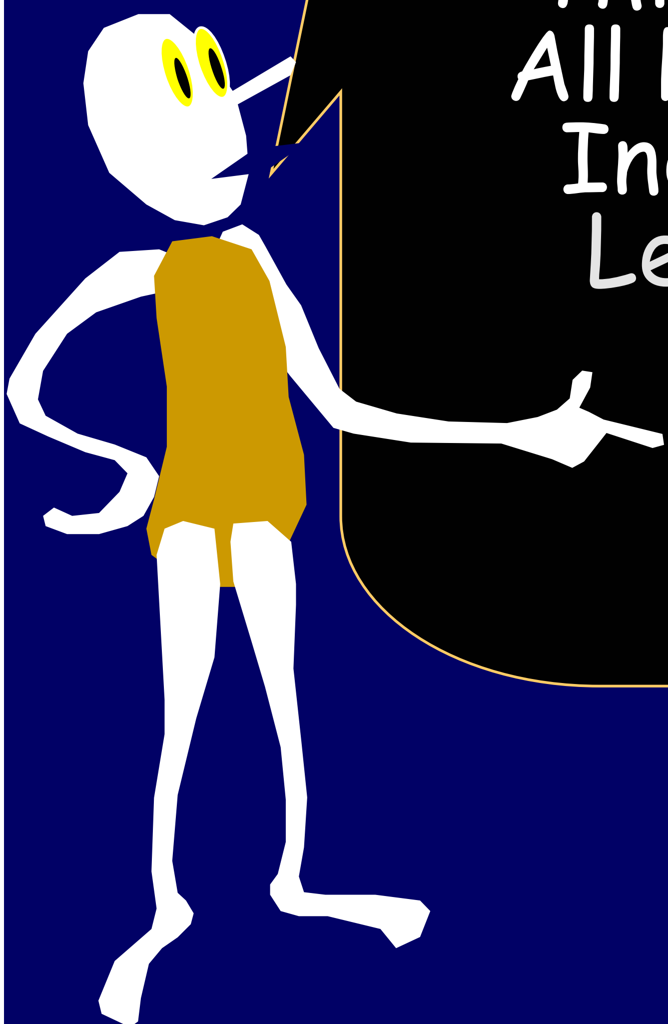
Let k be any natural number.

Assume $\forall j<k, S_j$

Prove S_k

All Previous, Contrapositive Induction:

Assume there is a least counter-example. Derive the existence of a smaller counter-example. Conclude there is no counter-example.

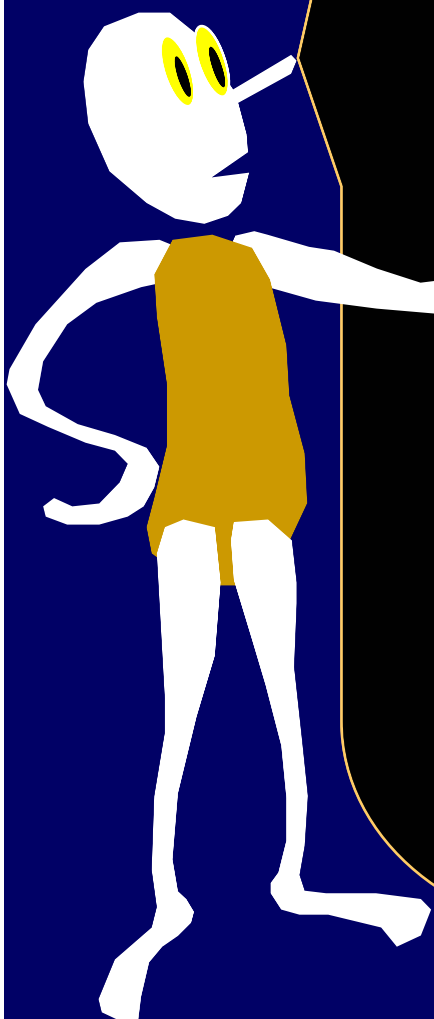


This is why we tend to call
All Previous, Contrapositive
Induction the Method of
Least-Counter Example.

Rene Descartes [1596-1650] "Method Of Infinite Decent"

Show that for any counter-example you find a smaller one. If a counter-example exists there would be an infinite sequence of smaller and smaller counter examples.

Euclid's theorem on the unique factorization of a number into primes.



Assume there is a least counter-example. Derive the existence of a smaller counter-example.

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Definition: A number > 1 is prime if it has no other factors, besides 1 and itself.

Primes: 2, 3, 5, 7, 11, 13, 17, ...

Factorizations:

42 = 2 * 3 * 7

84 = 2 * 2 * 3 * 7

13 = 13

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Let n be the least counter-example. n has at least two ways of being written as a product of primes:

$$n = p_1\, p_2 \,..\, p_k = q_1\, q_2 \,...\, q_t$$

The p's must be totally different primes than the q's or else we could divide both sides by one of a common prime and get a smaller counter-example. Without loss of generality, assume $p_1 > q_1$.

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Let n be the least counter-example.

$n = p_1 \, p_2 \, .. \, p_k = q_1 \, q_2 \, \dots \, q_t$ [$p_1 > q_1$]

$n \geq p_1 p_1 > p_1 \, q_1 + 1$ [Since $p_1 > q_1$]

.

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Let n be the least counter-example.

$n = p_1\ p_2\ ..\ p_k = q_1\ q_2\ ...\ q_t$ [$p_1 > q_1$]

$n \geq p_1 p_1 > p_1\ q_1 + 1$ [Since $p_1 > q_1$]

$m = n - p_1 q_1$ [Thus $1 < m < n$]

Notice: $m = p_1(p_2\ ..\ p_k - q_1) = q_1(q_2\ ...\ q_t - p_1)$

Thus, $p_1|m$ and $q_1|m$

By unique factorization of m, $p_1 q_1|m$, thus $m = p_1 q_1 z$

Theorem: Each natural has a unique factorization into primes written in non-decreasing order.

Let n be the least counter-example.

$n = p_1\, p_2\, ..\, p_k = q_1\, q_2\, ...\, q_t$ [$p_1 > q_1$]

$n \geq p_1 p_1 > p_1\, q_1 + 1$ [Since $p_1 > q_1$]

$m = n - p_1 q_1$ [Thus $1 < m < n$]

Notice: $m = p_1(p_2\, ..\, p_k - q_1) = q_1(q_2\, ...\, q_t - p_1)$

Thus, $p_1 | m$ and $q_1 | m$

By unique factorization of m, $p_1 q_1 | m$, thus $m = p_1 q_1 z$

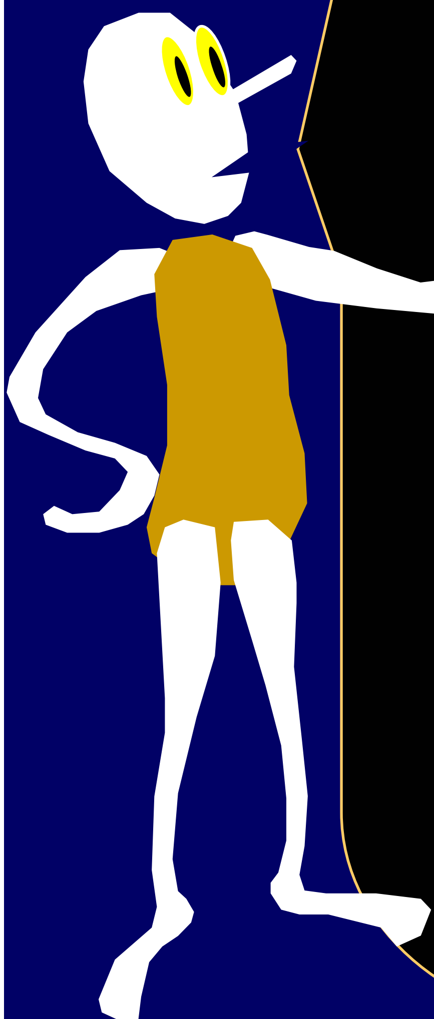
We have: $m = n - p_1 q_1 = p_1(p_2\, ..\, p_k - q_1) = p_1 q_1 z$

Dividing by p_1 we obtain: $(p_2\, ..\, p_k - q_1) = q_1 z$

$p_2\, ..\, p_k = q_1 z + q_1 = q_1(z+1)$ so $q_1 | p_2 ... p_k$

And hence, by unique factorization of $p_2 ... p_k$,

q_1 must be one of the primes $p_2, ..., p_k$. Contradiction of $q_1 < p_1$.



Inductive reasoning is the high level idea:

"Standard" Induction

"Contrapositive" Induction

"Strong" Induction are just different packaging.

“Strong” Induction Can Be Repackaged As Standard Induction

Establish “Base Case”: S_0

Establish that $\forall k, S_k \Rightarrow S_{k+1}$

Let k be any natural number.

Assume $\forall j<k, S_j$

Prove S_k

Define $T_i = \forall j \leq i, S_j$

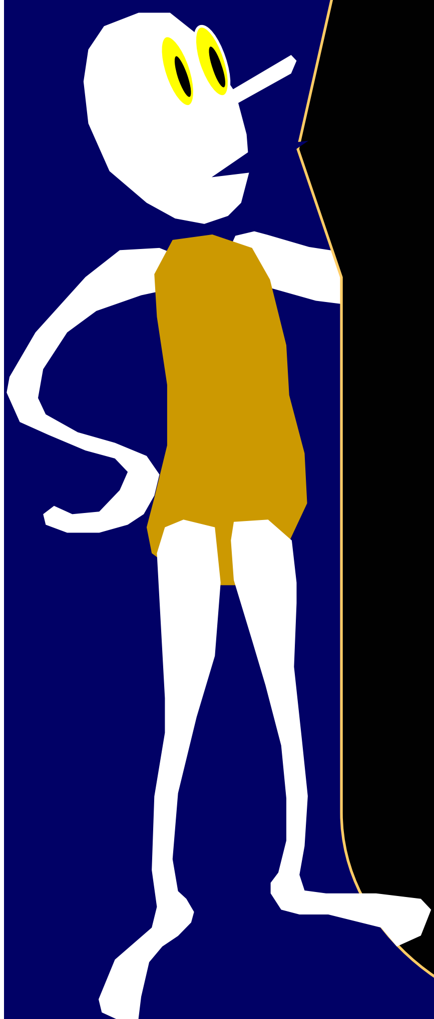
Establish “Base Case”: T_0

Establish that $\forall k, T_k \Rightarrow T_{k+1}$

Let k be any natural number.

Assume T_{k-1}

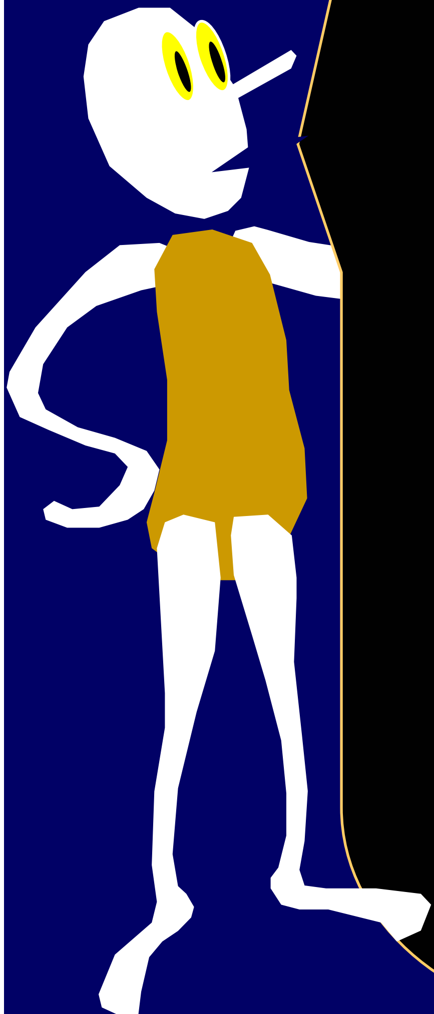
Prove T_k



Yet another way of packaging inductive reasoning is to define an "invariant".

Invariant:

1. Not varying; constant.
2. *Mathematics.* Unaffected by a designated operation, as a transformation of coordinates.



Yet another way of packaging inductive reasoning is to define an “invariant”.

Invariant:

3. <u>programming</u> A rule, such as the ordering an ordered list or heap, that applies throughout the life of a data structure or procedure. Each change to the data structure must maintain the correctness of the invariant.

Invariant Induction

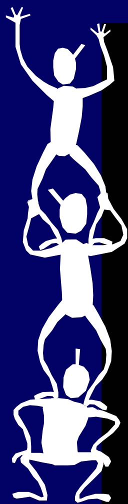
Suppose we have a time varying world state: W_0, W_1, W_2, …

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.



Invariant Induction

Suppose we have a time varying world state: W_0, W_1, W_2, …

Each state change is assumed to come from a list of permissible operations.

Let S be a statement true of W_0.

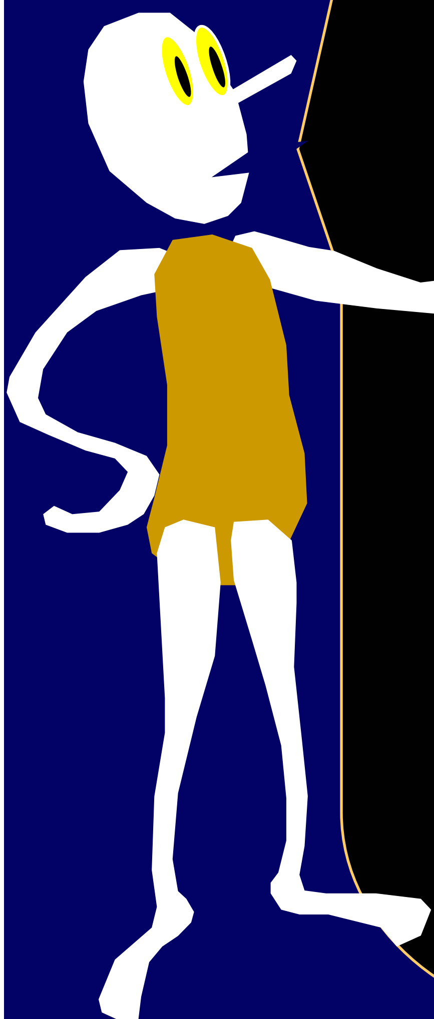
Let W be any possible future world state.

Assume S is true of W.

Show that S is true of any world W′ obtained by applying a permissible operation to W.

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.

Initial case: Zero hands have been shaken at the start of a party, so zero people have odd parity. If 2 people of different parities shake, then they both swap parities and the odd parity count is unchanged. If 2 people of the same parity shake, they both change and hence the odd parity count changes by 2 – and remains even.



Inductive reasoning is the high level idea:

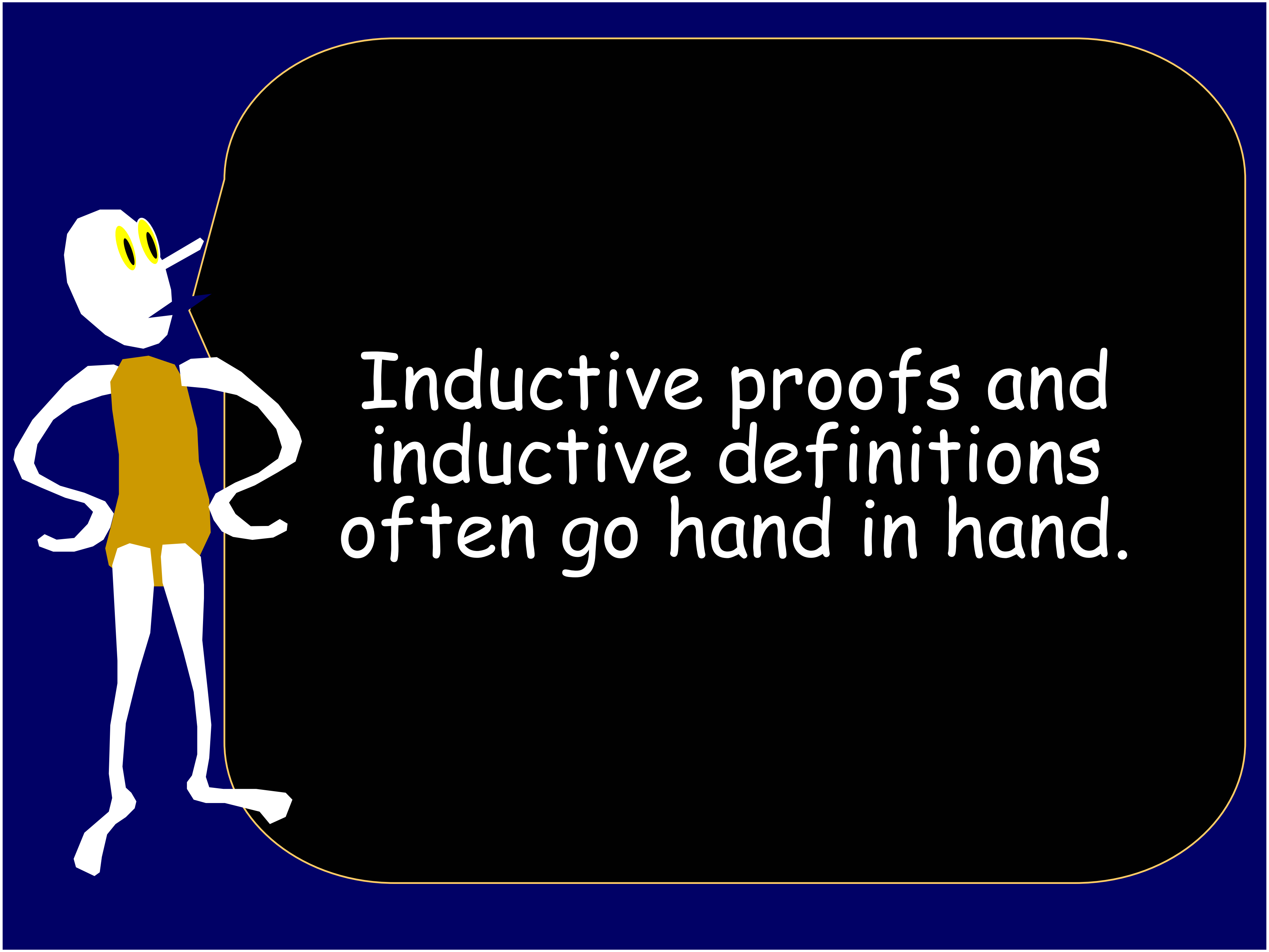
"Standard" Induction

"Contrapositive" Induction

"Strong" Induction

and Invariants

are just different packaging.


Inductive proofs and inductive definitions often go hand in hand.

Inductive Definition of T(n)

$T(1) = 1$

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

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

Inductive Definition of T(n)

$T(1) = 1$

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

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

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

Closed Form
Definition of G(n)

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

Domain of G are the powers of two.

Two equivalent functions?

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

Domain of G are the powers of two.

$T(1) = 1$

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

Domain of T are the powers of two.

Prove equivalence by induction on n: Assume T(x) = G(x) for x < n

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

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

Assuming $T(n/2) = G(n/2) = 2(n/2)^2 - n/2$

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

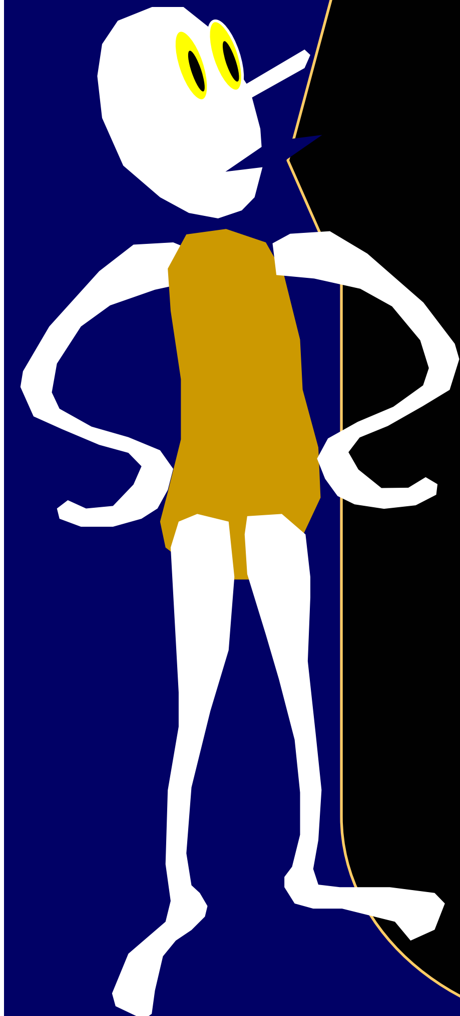
$\qquad =4[G(n/2) + n]$

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

$= \qquad 2n^2 - 2n \qquad + n$

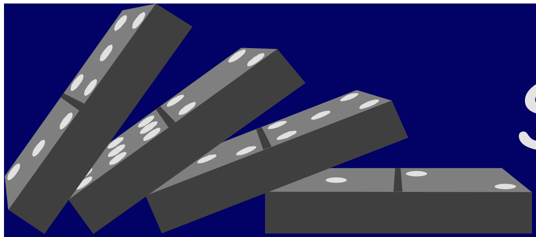
$= \qquad 2n^2 - n$

$= \qquad G(n)$



We inductively proved the assertion that $G(n) = T(n)$.

Giving a formula for T with no sums or recurrences is called solving the recurrence T.



Solving Recurrences
Guess and Verify

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

Verify: $G(1) = 1$ and $G(n) = 4\,G(n/2) + n$

Similarly: $T(1) = 1$ and $T(n) = 4\,T(n/2) + n$

So $T(n) = G(n)$

Study Bee

Inductive Proof

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

Inductive Definition

- Bottom-Up Programming
- Top-Down Programming
- Recurrence Relations
- Solving a Recurrence

Logic

- Contrapositive Form of S