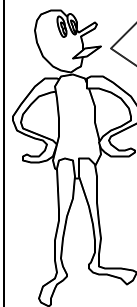
Great Theoretical Ideas In Computer Science		
Steven Rudich		CS 15-251 Spring 2005
Lecture 4	Jan 20, 2005	Carnegie Mellon University

Unary and Binary



Your Ancient Heritage

Let's take a historical view on abstract representations.

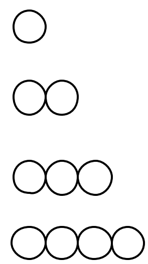
Mathematical Prehistory: 30,000 BC

Paleolithic peoples in Europe record unary numbers on bones.

1 represented by 1 mark
2 represented by 2 marks
3 represented by 3 marks
4 represented by 4 marks
...

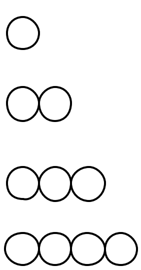
Prehistoric Unary

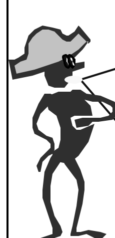
1
2
3
4



PowerPoint Unary

1
2
3
4

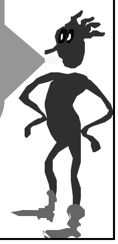




Hang on a minute!

Isn't unary a bit *literal* as a representation? Does it deserve to be viewed as an "abstract" representation?

In fact, it is important to respect the status of each representation, no matter how primitive. Unary is a perfect object lesson.

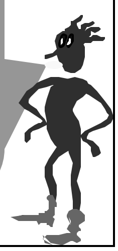


Consider the problem of finding a formula for the sum of the first n numbers.

We already used induction to verify that the answer is ½n(n+1)

Consider the problem of finding a formula for the sum of the first n numbers.

First, we will give the standard high school algebra proof….



1 + 2 + 3 + . . . + n-1 + n = S

1 + 2 + 3 + . . . + n-1 + n = S

n + n-1 + n-2 + . . . + 2 + 1 = S

1 + 2 + 3 + . . . + n-1 + n = S

n + n-1 + n-2 + . . . + 2 + 1 = S

(n+1) + (n+1) + (n+1) + . . . + (n+1) + (n+1) = 2S

n (n+1) = 2S

$$1 + 2 + 3 + \ldots + n-1 + n = S$$

$$n + n-1 + n-2 + \ldots + 2 + 1 = S$$

$$(n+1) + (n+1) + (n+1) + \ldots + (n+1) + (n+1) = 2S$$

$$n\,(n+1) = 2S$$

$$\boxed{S = \frac{n\,(n+1)}{2}}$$

$$1 + 2 + 3 + \ldots + n-1 + n = S$$

$$n + n-1 + n-2 + \ldots + 2 + 1 = S$$

$$(n+1) + (n+1) + (n+1) + \ldots + (n+1) + (n+1) = 2S$$

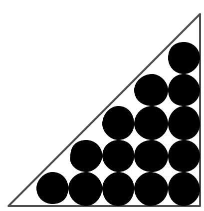
$$n\,(n+1) = 2S$$

Algebraic argument

Let's restate this argument using a UNARY representation

$$\boxed{1 + 2 + 3 + \ldots + n-1 + n = S}$$

= number of white dots.

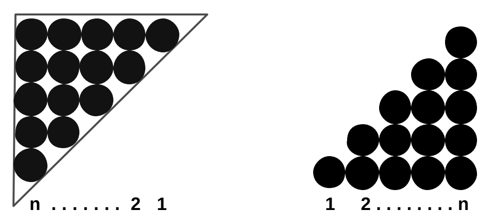


$$1 + 2 + 3 + \ldots + n-1 + n = S$$

= number of white dots

$$\boxed{n + n-1 + n-2 + \ldots + 2 + 1 = S}$$

= number of yellow dots

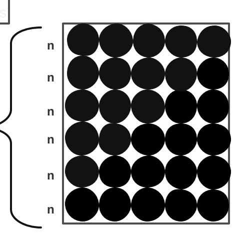


$$1 + 2 + 3 + \ldots + n-1 + n = S$$

= number of white dots

= number of yellow dots

There are n(n+1) dots in the grid

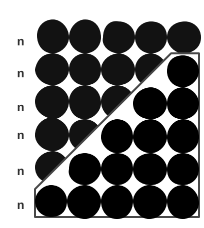


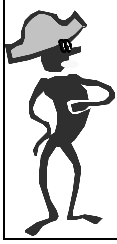
$$1 + 2 + 3 + \ldots + n-1 + n = S$$

= number of white dots

= number of yellow dots

$$\boxed{S = \frac{n\,(n+1)}{2}}$$





Very convincing! The unary representation brings out the geometry of the problem and makes each step look very natural.

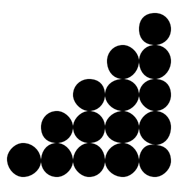
By the way, my name is Bonzo. And you are?

Odette.

Yes, Bonzo. Let's take it even further...

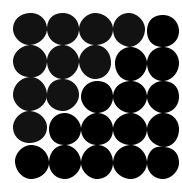
n^{th} Triangular Number

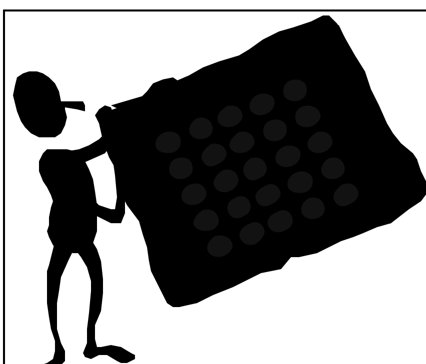
$$\Delta_n = 1 + 2 + 3 + . . . + n\text{-}1 + n$$
$$= n(n+1)/2$$



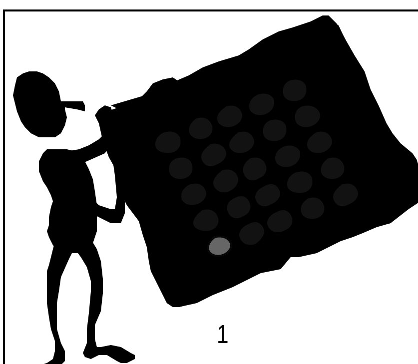
n^{th} Square Number

$$\tilde{}_n = \Delta_n + \Delta_{n-1}$$
$$= n^2$$

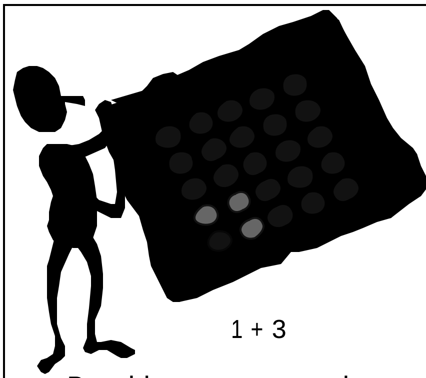




Breaking a square up in a new way.

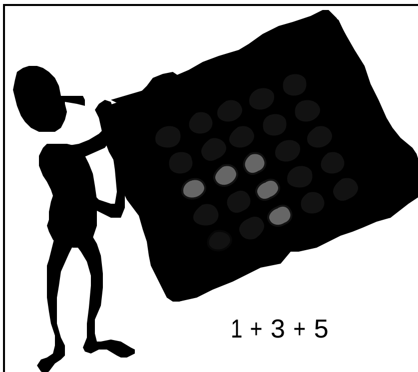


Breaking a square up in a new way.



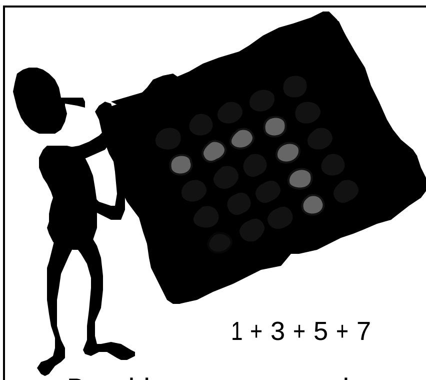
1 + 3

Breaking a square up in a new way.



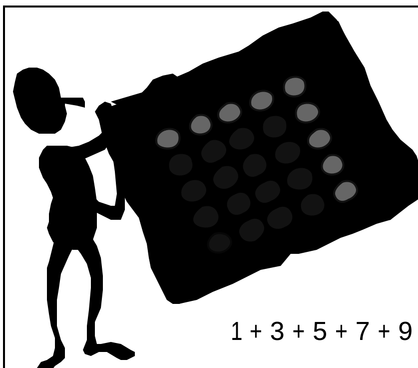
1 + 3 + 5

Breaking a square up in a new way.



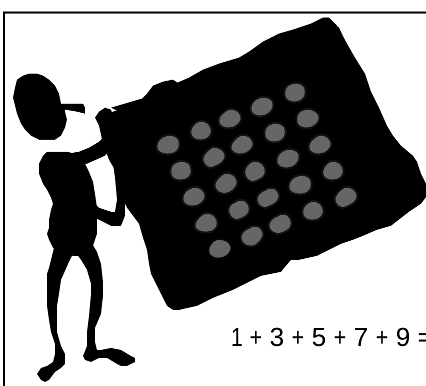
1 + 3 + 5 + 7

Breaking a square up in a new way.



1 + 3 + 5 + 7 + 9

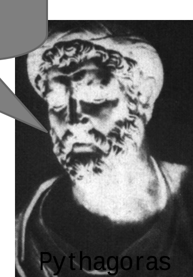
Breaking a square up in a new way.

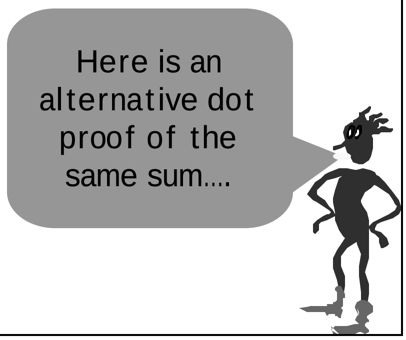


$1 + 3 + 5 + 7 + 9 = 5^2$

The sum of the first 5 odd numbers is 5 squared

The sum of the
first n odd numbers
is n squared.

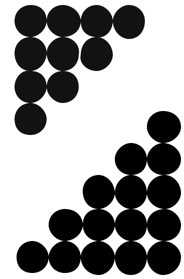




nth Square Number

$\sim_n = \Delta_n + \Delta_{n-1}$

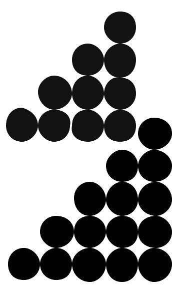
$= n^2$



nth Square Number

$\sim_n = \Delta_n + \Delta_{n-1}$

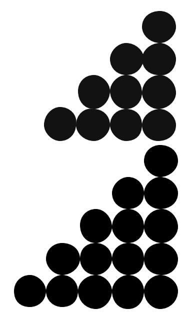
$= n^2$



Look at the columns!

$\sim_n = \Delta_n + \Delta_{n-1}$

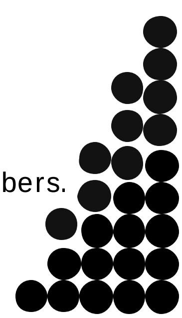
.



Look at the columns!

$\sim_n = \Delta_n + \Delta_{n-1}$

= Sum of first n odd numbers.



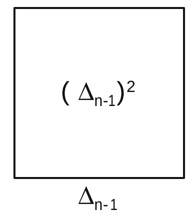
High School Notation

$$\Delta_n + \Delta_{n-1} =$$

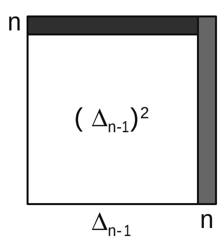
$$\begin{array}{rl} & 1 + 2 + 3 + 4 + 5 \ldots \\ + & \quad 1 + 2 + 3 + 4 \ldots \\ \hline & 1 + 3 + 5 + 7 + 9 \ldots \end{array}$$

Sum of odd numbers

$(\Delta_{n-1})^2$ = area of square



$(\Delta_n)^2$ = area of square



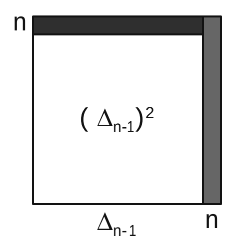
$n\Delta_n + n\Delta_{n-1}$

$= n(\Delta_n + \Delta_{n-1})$

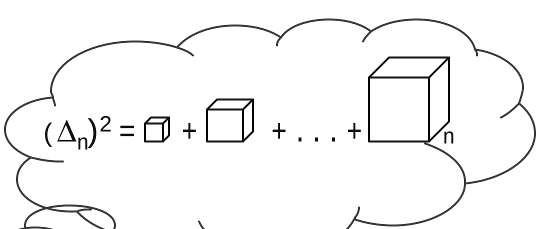
$= n$ ~ $_n$

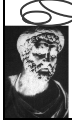
$= \square_n$ = area of pieces

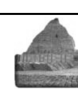
$(\Delta_n)^2 = (\Delta_{n-1})^2 + \square_n$

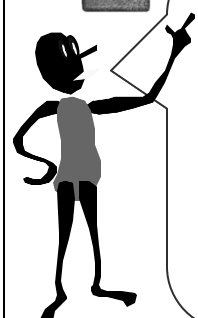


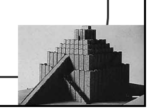
$(\Delta_n)^2 = (\Delta_{n-1})^2 + \square_n$

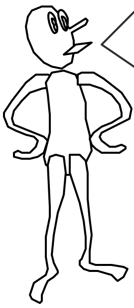












The ancients grappled with problems of abstraction in representation and reasoning.

Let's look back to the dawn of symbols...

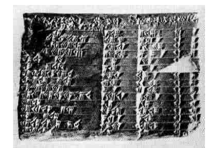
Sumerians [modern Iraq]

8000 BC Sumerian tokens use multiple symbols to represent numbers

3100 BC Develop Cuneiform writing

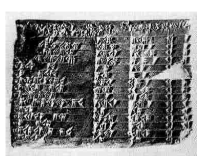
2000 BC Sumerian tablet demonstrates:
- base 10 notation (no zero)
- solving linear equations
- simple quadratic equations

Biblical timing: Abraham born in the Sumerian city of Ur



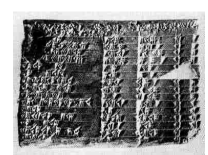
Babylonians absorb Sumerians

1900 BC Sumerian/Babylonian Tablet
- Sum of first n numbers
- Sum of first n squares
- "Pythagorean Theorem"
- "Pythagorean Triplets", e.g., 3-4-5
- some bivariate equations



Babylonians

1600 BC Babylonian Tablet
- Take square roots
- Solve system of n linear equations



Egyptians

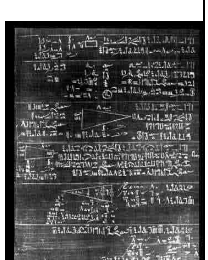
6000 BC Multiple symbols for numbers

3300 BC Developed Hieroglyphics

1850 BC Moscow Papyrus
- Volume of truncated pyramid

1650 BC Rhind Papyrus [Ahmose]
- Binary Multiplication/Division
- Sum of 1 to n
- Square roots
- Linear equations

Biblical timing: Joseph is Governor of Egypt.



Harrappans [Indus Valley Culture] Pakistan/India

3500 BC Perhaps the first writing system?!

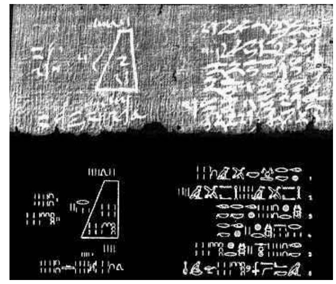
2000 BC Had a uniform decimal system of weights and measures

China

1200 BC Independent writing system
Surprisingly late.

1200 BC I Ching [Book of changes]
Binary system developed to do numerology.

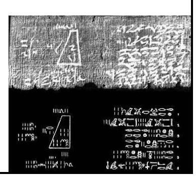
Rhind Papyrus
Scribe Ahmose was the Martin Gardener of his day!



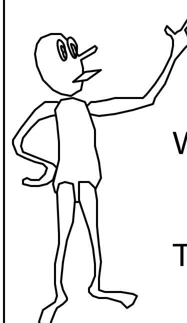
Rhind Papyrus
87 Problems.

A man has seven houses,
Each house contains seven cats,
Each cat has killed seven mice,
Each mouse had eaten seven ears of spelt,
Each ear had seven grains on it.
What is the total of all of these?

Sum of first five powers of 7



$$1 + X^1 + X^2 + X^3 + \dots + X^{n-2} + X^{n-1} = \frac{X^n - 1}{X - 1}$$



We will soon need this fundamental sum:

The Geometric Series

A Frequently Arising Calculation

$(X-1)\,(1 + X^1 + X^2 + X^3 + \dots + X^{n-2} + X^{n-1})$

$= \quad X^1 + X^2 + X^3 + \dots \quad + X^{n-1} + X^n$

$\quad - 1 - X^1 - X^2 - X^3 - \dots - X^{n-2} - X^{n-1}$

$= \quad -1 \quad + \quad X^n$

$= \quad X^n - 1$

Action Shot: Mult by X is a SHIFT

$X\,(1 + X^1 + X^2 + X^3 + \dots \dots\dots + X^{n-2} + X^{n-1})$

$= \quad + X^1 + X^2 + X^3 + \dots \quad + X^{n-1} + X^n$

The Geometric Series

$$(X-1)\,(1 + X^1 + X^2 + X^3 + \ldots + X^{n-2} + X^{n-1}) = X^n - 1$$

$$1 + X^1 + X^2 + X^3 + \ldots + X^{n-2} + X^{n-1} = \frac{X^n - 1}{X - 1}$$

when $X \neq 1$

The Geometric Series When X=2

$$1 + 2^1 + 2^2 + 2^3 + \ldots\ 2^{n-1} = 2^n - 1$$

$$1 + X^1 + X^2 + X^3 + \ldots + X^{n-2} + X^{n-1} = \frac{X^n - 1}{X - 1}$$

when $X \neq 1$

The Geometric Series When X=3

$$1 + 3^1 + 3^2 + 3^3 + \ldots\ 3^{n-1} = (3^n - 1)/2$$

$$1 + X^1 + X^2 + X^3 + \ldots + X^{n-2} + X^{n-1} = \frac{X^n - 1}{X - 1}$$

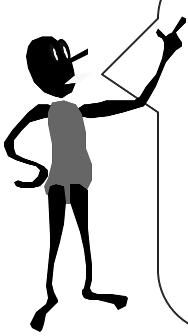
when $X \neq 1$

The Geometric Series When X= ½

$$1 + \tfrac{1}{2}^1 + \tfrac{1}{2}^2 + \tfrac{1}{2}^3 + \ldots\ \tfrac{1}{2}^{n-1} = (\tfrac{1}{2}^n - 1) / -\tfrac{1}{2} = 2 - (\tfrac{1}{2})^{n-1}$$

$$1 + X^1 + X^2 + X^3 + \ldots + X^{n-2} + X^{n-1} = \frac{X^n - 1}{X - 1}$$

when $X \neq 1$



Numbers and their properties can be represented as strings of symbols.

Strings Of Symbols.

We take the idea of symbol and sequence of symbols as primitive.

Let Σ be any fixed finite set of symbols.
Σ is called an alphabet, or a set of symbols.
Examples:

- $\Sigma = \{0,1,2,3,4\}$
- $\Sigma = \{a,b,c,d, \ldots, z\}$
- Σ = all typewriter symbols.
- Σ = { ♋ , ♌ , ♍ , ♎ , ..., ⌘ }

Strings over the alphabet Σ.

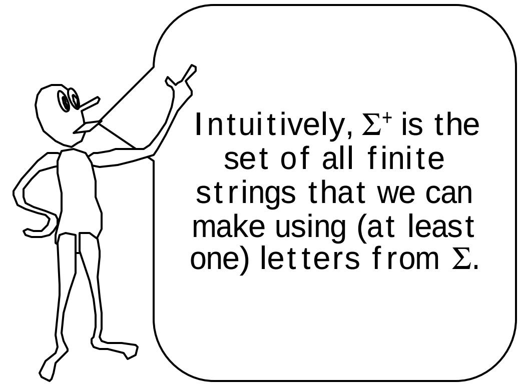
A string is a sequence of symbols from Σ.

Let s and t be strings. Then st denotes the concatenation of s and t; i.e., the string obtained by the string s followed by the string t.

Now define Σ^+ by these inductive rules:

$$x \in \Sigma \Rightarrow x \in \Sigma^+$$

$$s,t \in \Sigma^+ \Rightarrow st \in \Sigma^+$$



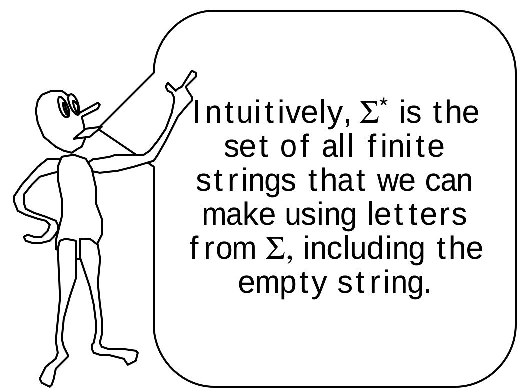
The set Σ^*

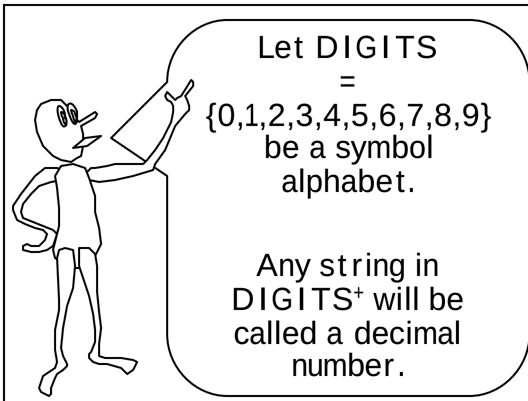
Define ε be the empty string.
I.e., $X\varepsilon Y = XY$ for all strings X and Y.

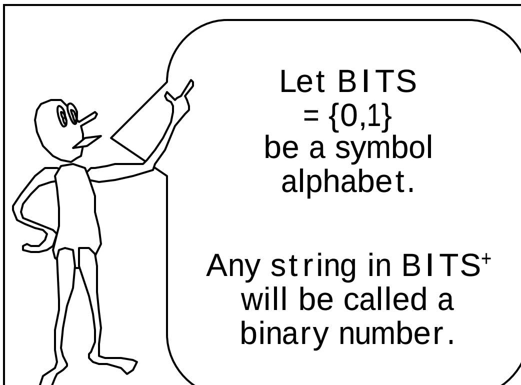
ε is also called the string of length 0.

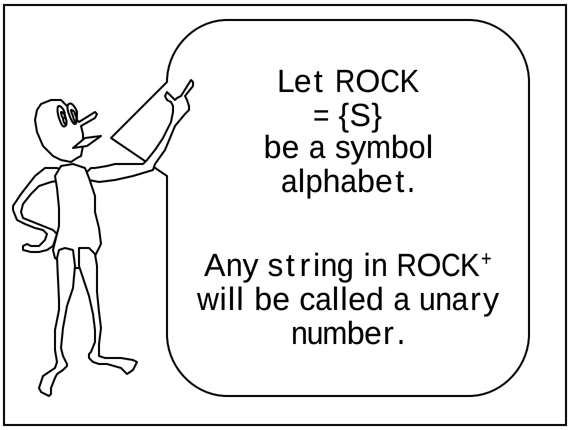
Define $\Sigma^0 = \{ \varepsilon \}$

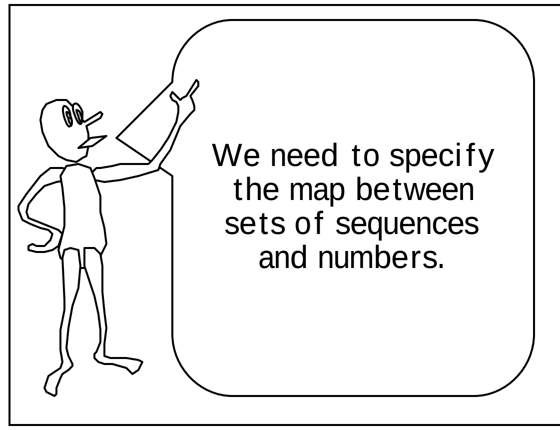
Define $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$

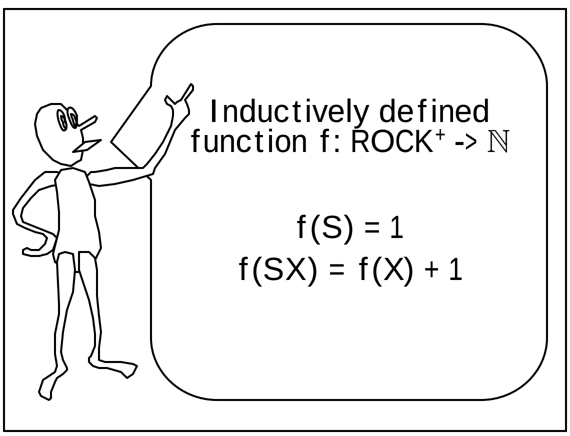


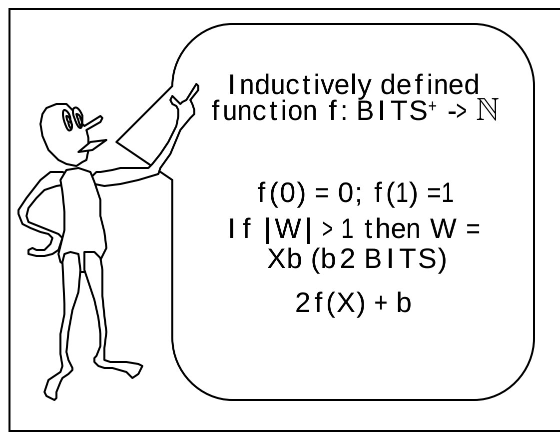


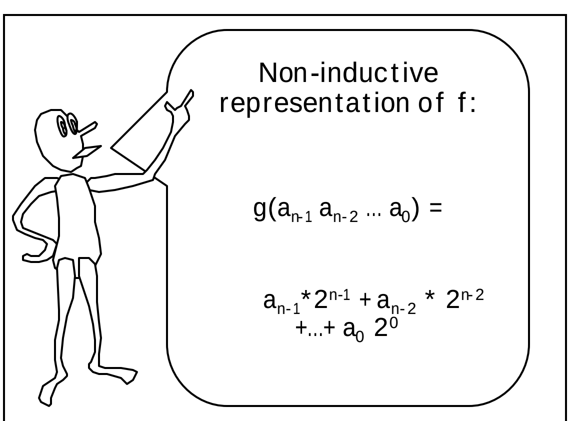


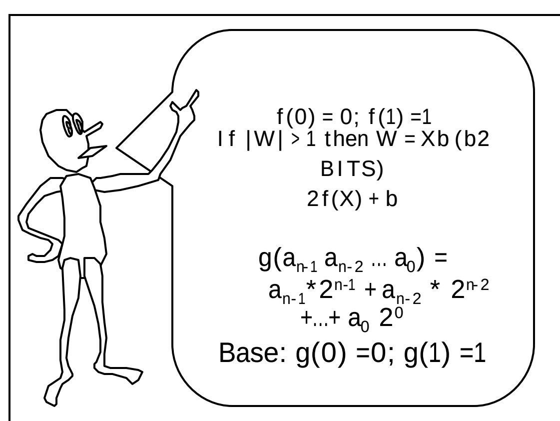

Let ROCK
= {S}
be a symbol
alphabet.
Any string in ROCK$^+$
will be called a unary
number.

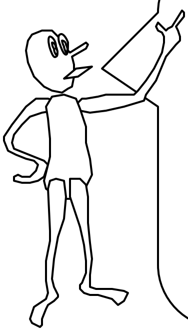

We need to specify
the map between
sets of sequences
and numbers.


Inductively defined
function f: ROCK$^+$ -> $\mathbb{N}$
f(S) = 1
f(SX) = f(X) + 1


Inductively defined
function f: BITS$^+$ -> $\mathbb{N}$
f(0) = 0; f(1) =1
If |W| > 1 then W =
Xb (b2 BITS)
2f(X) + b


Non-inductive
representation of f:
$g(a_{n-1}\ a_{n-2} \dots a_0) =$
$a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2}$
$+ \dots + a_0\ 2^0$


f(0) = 0; f(1) =1
If |W| > 1 then W = Xb (b2
BITS)
2f(X) + b
$g(a_{n-1}\ a_{n-2} \dots a_0) =$
$a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2}$
$+ \dots + a_0\ 2^0$
Base: g(0) =0; g(1) =1



$f(0) = 0$; $f(1) = 1$
If $|W| > 1$ then $W = Xb$ (b2 BITS)
$2f(X) + b$

$g(a_{n-1}\, a_{n-2} \ldots a_0) = a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_0\, 2^0$

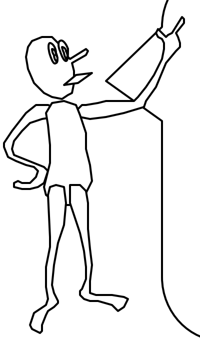
$g(a_{n-1}\, a_{n-2} \ldots a_0) = 2g(a_{n-1}\, a_{n-2} \ldots a_1) + a_0$

$g(a_{n-1}\, a_{n-2} \ldots a_0) = 2g(a_{n-1}\, a_{n-2} \ldots a_1) + a_0$

ACTION SHOT: Mult by 2 as SHIFT

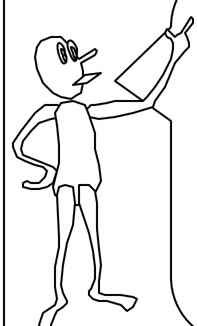
$2\,(a_{n-1} * 2^{n-2} + a_{n-2} * 2^{n-3} + \ldots + a_1\, 2^0) =$

$a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_1\, 2^1$



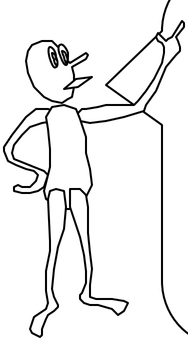
TWO IDENTICAL MAPS FROM SEQUENCES TO NUMBERS:
$f(0) = 0$; $f(1) = 1$
If $|W| > 1$ then $W = Xb$ (b2 BITS)
$2f(X) + b$

$f(a_{n-1}\, a_{n-2} \ldots a_0) = a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_0\, 2^0$



The symbol a_0 is called the Least Significant Bit or the Parity Bit. $a_0 = 0$ iff $a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_0\, 2^0$ is an even number.

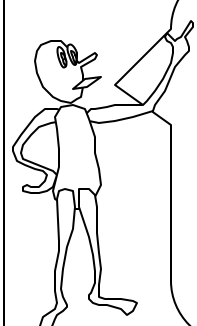
$f(a_{n-1}\, a_{n-2} \ldots a_0) = a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_0\, 2^0$



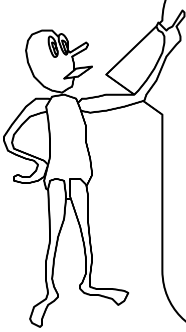
Theorem: Each natural has at least one binary representation.

Base Case: 0 and 1 do.
Induction hypothesis: Suppose all natural numbers less than n have a binary representation. Note that n=2m+b for some m<n,

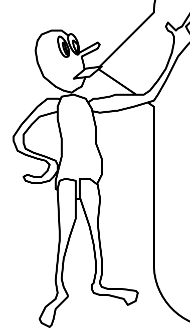
b=0 or 1. Represent n as the left-shifted sequence for m concatenated with the symbol for b.



Theorem: Each natural has a unique binary representation. Base: 0 and 1 do. Induction Hypothesis: Every natural number less than n has a unique binary representation. Suppose n=2m+b has 2 binary representations W and V. Their parity bit b must be identical. Hence, m also has two distinct binary representations, which contradicts the induction hypothesis. So n must have a unique representation.



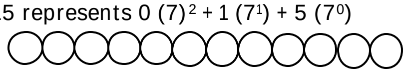
Inductive definition is great for showing UNIQUE representation:
If |W| > 1 then W = Xb (b2 BITS)
2f(X) + b
Let n be the smallest number reprinted by two different binary sequences. They must have the same parity bit, thus we can make a smaller number that has distinct representations.



EACH NATURAL NUMBER HAS A UNIQUE REPRESENTATION AS A BINARY NUMBER.

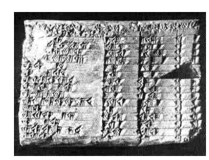
BASE X:
$S = a_{n-1}, a_{n-2}, \ldots, a_0$
represents the number:
$a_{n-1} X^{n-1} + a_{n-2} X^{n-2} + \ldots + a_0 X^0$

Base 2 [Binary Notation]
101 represents $1 (2)^2 + 0 (2^1) + 1 (2^0)$
=


Base 7
015 represents $0 (7)^2 + 1 (7^1) + 5 (7^0)$
=


Bases In Different Cultures

Sumerian-Babylonian: 10, 60, 360
Egyptians: 3, 7, 10, 60
Africans: 5, 10
French: 10, 20
English: 10, 12,20



Fundamental Theorem For Binary:

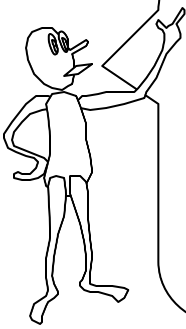
Each of the numbers from 0 to 2^n-1 is uniquely represented by an n-digit number in binary.

k uses $\lfloor \log_2 k \rfloor + 1$ digits in base 2.

Fundamental Theorem For Base X:

Each of the numbers from 0 to X^n-1 is uniquely represented by an n-digit number in base X.

k uses $\lfloor \log_x k \rfloor + 1$ digits in base X.



n has length n in unary, but has length $\lfloor \log_2 n \rfloor + 1$ in binary.

Unary is exponentially longer than binary.

Egyptian Multiplication

The Egyptians used decimal numbers but multiplied and divided in binary

Egyptian Multiplication a times b by repeated doubling

b has some n-bit representation: $b_n..b_0$

Starting with a,
repeatedly double largest so far to obtain: a, 2a, 4a, …., $2^n a$

Sum together all $2^k a$ where $b_k = 1$

Egyptian Multiplication 15 times 5 by repeated doubling

5 has some 3-bit representation: 101

Starting with 15,
repeatedly double largest so far to obtain: 15, 30, 60

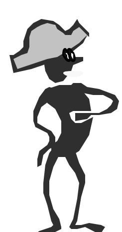
Sum together all $2^k(15)$ where $b_k = 1$:

15 + 60 = 75

Why does that work?

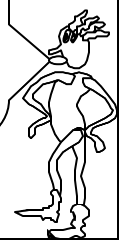
$$b = b_0 2^0 + b_1 2^1 + b_2 2^2 + \ldots + b_n 2^n$$
$$ab = b_0 2^0 a + b_1 2^1 a + b_2 2^2 a + \ldots + b_n 2^n a$$

If b_k is 1 then $2^k a$ is in the sum.
Otherwise that term will be 0.



Wait! How did the Egyptians do the part where they converted b to binary?

They used repeated halving to do base conversion. Consider …



Egyptian Base Conversion

Output stream will print right to left.
Input X.
Repeat until X=0
{
    If X is even then Output O;
        Otherwise {X:=X-1; Output 1}

    X:=X/2
}

Egyptian Base Conversion

Output stream will print right to left.
Input X.
Repeat until X=0
{
    If X is even then Output O;
        Otherwise Output 1

    X:= $\lfloor X/2 \rfloor$
}

Start the algorithm

010101

1

Repeat until X=0
{ If X is even then Output O;
Otherwise Output 1;
X:= $\lfloor X/2 \rfloor$
}

Start the algorithm

01010

1

Repeat until X=0
{ If X is even then Output O;
Otherwise Output 1;
X:= $\lfloor X/2 \rfloor$
}

Start the algorithm

01010

01

Repeat until X=0
{ If X is even then Output O;
Otherwise Output 1;
X:= $\lfloor X/2 \rfloor$
}

Start the algorithm

0101

01

Repeat until X=0
{ If X is even then Output O;
Otherwise Output 1;
X:= $\lfloor X/2 \rfloor$
}

Start the algorithm

0101

101

Repeat until X=0
{ If X is even then Output O;
Otherwise Output 1;
X:= $\lfloor X/2 \rfloor$
}

Start the algorithm

010

101

Repeat until X=0
{ If X is even then Output O;
Otherwise Output 1;
X:= $\lfloor X/2 \rfloor$
}

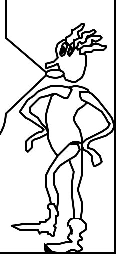
And Keep Going until 0

0

010101

Repeat until X=0
{ If X is even then Output O;
Otherwise Output 1;
X:= $\lfloor X/2 \rfloor$
}

Sometimes the Egyptian combined the base conversion by halving and the multiplication by doubling into one algorithm



Rhind Papyrus (1650 BC)
70*13

70	13	*	70
140	6		
280	3	*	350
560	1	*	910

Rhind Papyrus (1650 BC)
70*13

70	13 *	70
140	6	
280	3 *	350
560	1 *	910

Binary for 13 is 1101 = $2^3 + 2^2 + 2^0$

$70*13 = 70*2^3 + 70*2^2 + 70*2^0$

Rhind Papyrus (1650 BC)

17	1
34	2 *
68	4
136	8 *

184 48 14

Rhind Papyrus (1650 BC)

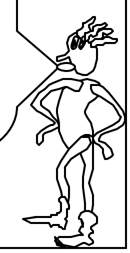
17	1
34	2 *
68	4
136	8 *

184 48 14

184 = 17*8 + 17*2 + 14

184/17 = 10 with remainder 14

This method is called “Egyptian Multiplication/Division” or “Russian Peasant Multiplication/Division”.



Wow. Those Russian peasants were pretty smart.

Standard Binary Multiplication = Egyptian Multiplication

```
 X   * * * * * * * *
                 101
 --------------------
     * * * * * * * *
   * * * * * * * *
 --------------------
 * * * * * * * * * * *
```