Numbers and Information

INFO/CSE 100, Autumn 2004

Fluency in Information Technology

http://www.cs.washington.edu/100

Readings and References

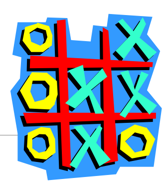
- Reading
  - » *Fluency with Information Technology*
    - Chapter 11, Representing Multimedia Digitally
- References
  - » Some clip art is from the Open Clip Art Library
    - permission to use is granted on their web site
    - http://www.openclipart.org/index.php
  - » Wolfram Research
    - http://mathworld.wolfram.com/
    - http://www.wolfram.com/

Recall: Info Representation

- Digitization: representing information by any fixed set of symbols
  - » decide how many different items of information you want to represent
    - Tic Tac Toe: 3 items - *empty cell* or *player 1* or *player 2*
  - » decide how many "digits" or positions you want to use
    - Tic Tac Toe: 9 positions - one per board square
  - » decide on a set of symbols

player 1: ✕ empty cell: ⊗

player 2: ○



Empty position: ⊗

use this set of symbols
- empty cell: ⊗
- player 1: ✕
- player 2: ○

○	⊗	⊗
✕	✕	○
⊗	⊗	⊗

- We can represent this game as one 9-digit string:

  ○ ⊗ ⊗ ✕ ✕ ○ ⊗ ⊗ ⊗
- How many possible game states are there?
  - » $3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 \times 3 = 3^9 = 19683$

Another encoding

use a different set of symbols
- empty cell: 0
- player 1: **1**
- player 2: **2**

2	0	0
1	**1**	**2**
0	0	0

- We can represent this game as one 9-digit number:
  2 0 0 1 1 2 0 0 0
- How many possible game states are there?
  » $3\times3\times3\times3\times3\times3\times3\times3\times3 = 3^9 = 19683$

Info Representation

- Adult humans have 32 teeth
  » sometimes a tooth or two is missing!
- How can we represent a set of teeth?
  » How many different items of information?
  - 2 items - *tooth* or *no tooth*

  » How many "digits" or positions to use?
  - 32 positions - one per tooth socket

  » Choose a set of symbols
  no tooth: 0 *tooth*: 1

What's your tooth number?

incisors | canines | pre-molars | molars

0000 0000 | 0000 | 0000 0000 | 0000 0000 0000

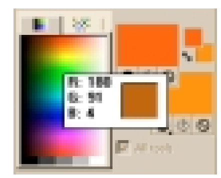
no teeth ↔ 0000 0000 0000 0000 0000 0000 0000 0000

1111 1111 1111 1111 1111 0000 0000 0000

no molars ↔ 1111 1111 1111 1111 1111 0000 0000 0000

How many possible combinations? $2\times2\times2\times2\times\ldots\times2 = 2^{32} \approx$ 4 Billion

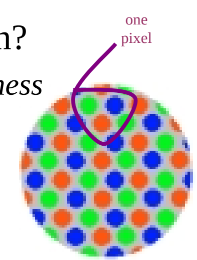
Info Representation



- Color monitors combine light from Red, Green, and Blue phosphors to show us colors
- How can we represent a particular color?
  » How many different items of information?
  - 256 items - *distinguish 256 levels of brightness*

  » How many "digits" or positions to use?
  - 3 positions - *one Red, one Green, one Blue*

  » Choose a set of symbols
  brightness level represented by the numbers 0 to 255



What is the pixel's color?

	red	green	blue
(red)	255	0	0
(orange)	255	102	0
(gray)	128	128	128
(black)	0	0	0

How many possible combinations?
$256 \times 256 \times 256 = 256^3 \approx 16$ Million

16 M colors is often called "True Color"

How can we store numbers?

- We want to store numbers
  - » 0 to 255 for color brightness
  - » 0 to 4B for tooth configuration
  - » 0 to 255 for ASCII character codes
- What do we have available in memory?
  - » *Binary digits*
    - 0 *or* 1
    - on *or* off
    - clockwise *or* counter-clockwise

…0010 0111 0000 1001 0000 1001 0000 1001…

The hardware is binary

- 0 and 1 are the only symbols the computer can actually store directly in memory
  - » a single bit is either *off* or *on*
- How many numbers can we represent with 0 and 1?
  - » How many different items of information?
    - 2 items - *off* or *on*
  - » How many "digits" or positions to use?
    - let's think about that on the next slide
  - » Choose a set of symbols
    - already chosen: 0 and 1

How many positions should we use?

It depends: how many numbers do we need?

one position: two numbers

0
1

two positions: four numbers

0	0
0	1
1	0
1	1

three positions: eight numbers

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

The sky's the limit

… 1101010001011110010101010100001110101001010101

- We can get as many numbers as we need by allocating enough positions
  - » each additional position means that we get *twice* as many values because we can represent *two* numbers in each position
  - » these are *base 2* or *binary* numbers
    - each position can represent two different values
- How many different numbers can we represent in base 2 using 4 positions?

How can we read binary numbers?

Let's look at the equivalent *decimal* (ie, *base 10*) numbers.

binary base 2	⇔	decimal base 10
0		0
1		1

binary base 2		⇔	decimal base 10
0	0		0
0	1		1
1	0		2
1	1		3

binary base 2			⇔	decimal base 10
0	0	0		0
0	0	1		1
0	1	0		2
0	1	1		3
1	0	0		4
1	0	1		5
1	1	0		6
1	1	1		7

111_2 represents *exactly the same quantity* as 7_{10}
They are just different ways of representing the same number.

Position matters!

binary base 2				⇔	decimal base 10
	0	0	0		0
	0	0	1		1
	0	1	0		2
	0	1	1		3
	1	0	0		4
	1	0	1		5
	1	1	0		6
	1	1	1		7
1	0	0	0		

position represents _	position represents 4	position represents 2	position represents 1	base 10
1	0	1	0	base 2

What do the positions represent?

$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	2×2×2×2 $2^4 = 16$	2×2×2 $2^3 = 8$	2×2 $2^2 = 4$	2 $2^1 = 2$	1 $2^0 = 1$	base 10
1	0	0	0	1	0	1	0	base 2

Each position represents one more multiplication by the base value.
For binary numbers, the base value is 2, so each new column represents a multiplication by 2.

What base 10 decimal value is equivalent to the base 2 binary value 10001010_2 shown above?

Some Examples

$2^7 = 128$	$2^6 = 64$	$2^5 = 32$	$2^4 = 16$	$2^3 = 8$	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	base 10
_	_	_	_	_	_	_	_	base 2

$$10_2 = 2_{10}$$
$$100_2 = 4_{10}$$
$$110_2 = 4_{10} + 2_{10} = 6_{10}$$
$$111_2 = 4_{10} + 2_{10} + 1_{10} = 7_{10}$$
$$1000_2 = 8_{10}$$
$$1001_2 = 8_{10} + 1_{10} = 9_{10}$$

This is an old and very important idea

- "You see, more than 5000 years ago, the Babylonians--and probably the Sumerians before them--had the idea of positional notation for numbers. They mostly used base 60--not base 10--which is actually presumably where our hours, minutes, seconds scheme comes from. But they had the idea of using the same digits to represent multiples of different powers of 60. "
- "Well, this fine abstract Babylonian scheme for doing things was almost forgotten for nearly 3000 years. And instead, what mostly was used, I suspect, were more natural-language-based schemes, where there were different symbols for tens, hundreds, etc."
- Quoted from Mathematical Notation: Past and Future Keynote address presented by Stephen Wolfram at MathML and Math on the Web: MathML International Conference 2000
  - » http://www.stephenwolfram.com/publications/talks/mathml/