The Essence of Computation

Information

> Information is 'news' or 'surprise'
>
> – Girvin

Alice sends a message to Bob indicating whether it will be rainy or sunny with either a "True" or "False."

We choose to encode "True" as 1 and "False" as 0. This fundamental unit of information is the **bit** - "binary digit"

Taken from Chapter 1 of Girvin's PHYS 345 notes [1]

Bits can be represented by anything from heads/tails on a coin, on/off on a light switch, even the spin of an electron or GC/AT bases in DNA. Computers use **transistors**—which are made of semiconductors—to represent information.

Binary Math

Base 10

$$3141_{10} = 3 \cdot 10^3 + 1 \cdot 10^2 + 4 \cdot 10^1 + 1 \cdot 10^0$$

Count from 1 to 10. Notice what happens after you reach 9 (highest base-10 digit). We go to the next place value and start at the lowest base-10 digit.

Nota Bene: The subscript indicates base

Base 2

$$\begin{aligned} 101_2 &= 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 5_{10} \end{aligned}$$

Try counting in binary! `0, 1, 10, 11, 100, 101, 110, 111, 1000`. Notice the base-10 rule holds true.

Puzzle - Base 8

Show that $9_{10} + 10_8 = 21_8$.

Multiple Bits

Our first example works if there are only two types of weather. What if there are 8 possibilities for the weather, and Alice wants to uniquely tell Bob what the weather is?

Code	0	1	2	3	4	5	6	7	8
Weather	Clear	Partly Cloudy	Overcast	Showers	Rain	Heavy Rain	Thunderstorm	Snow	Hail

One True/False isn't enough—we need more bits. How many more, though? If we have 2 bits, we can represent `00, 01, 10, 11`. That's $2^2 = 4$ states. If you have n bits, you can represent 2^n possible states [1-1].

If there are 8 possible weather conditions, we use the relationship to find $log_2(8) = 3$ bits of information is sufficient for Bob to uniquely identify the weather from Alice's message.

Modern computers use lots of bits, so we use the **byte**, which is just 8 bits. You can prefix it with *kilo-*, *mega-*, *tera-*, to represent higher orders of magnitude.

Entropy and Compression

Introduction

Remember that information is something new - it is *disorder*.

None of bits in `00000000` tell me anything new; they're all the same. One of the bits in `0000100`, on the other hand, does. These bits could encode the weather or the president's nuclear codes—it doesn't really matter for this example.

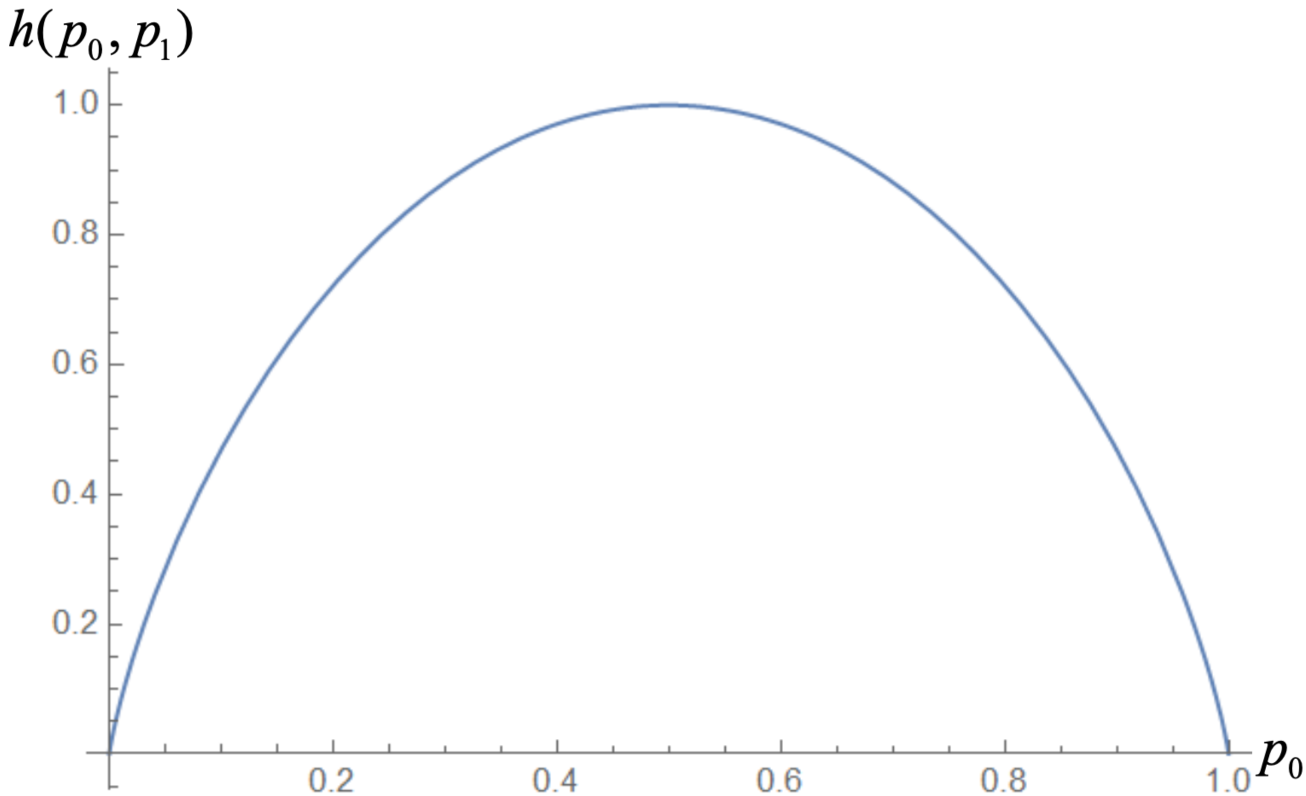
We could **compress** the first example into `'0'*8` ("there are 8 zeros") and the second into `'0'*4 + '1' + '0'*2` ("there are 4 zeros, a 1, followed by 2 more zeros"). The second one is more annoying because there is more disorder and more information —more **entropy**. There is less predictability[2].

Note: This is why longer passwords tend to be more secure: See this XKCD. This principle applies to encryption algorithms too. Unless... (Shor's Algorithm)

Shannon Entropy

As we saw with compression, not all bits in the binary string `000100` are "surprising". **Shannon Entropy** can quantify how "surprising" any bit is in the string.

It turns out that there's a relationship between the probability of a bit in the string being a 0 or a 1 and the Shannon Entropy of the string. At a 50-50 chance of being a 0 or a 1, that probability is maximized—there is maximal entropy. See below figure[1-2].



Booleans, Operations, and Logic

Booleans and Operations

A boolean—named after George Boole—is simply a True/False value, a 0 or 1 that can be represented with a bit.

There are 4 possible things we can do to a single bit:

1. Identity - do nothing
2. Negate - flip the bit
3. Set it to 0
4. Set it to 1

Logic

What if we had more bits? We could do more complex operations: computations. The key stepping stone from bits to classical computations is the **logic gate**, which performs a boolean logic operation.

But first, let's try doing some formal boolean logic. Are these statements logically true or false?

1. The sun rises in the east, and pigs can fly.

2. Bananas are yellow, or the sky is purple.

We can represent each statement with a boolean (indicating its truth) and logical operations. We'll get into what these operations mean momentarily.

1. True AND False = False
2. True OR False = True

Logic Gates: Truth Tables

Logic gates are circuit implementations of logical operations—using transistors. (Draw diagrams, do some exercises)

NOT Gate

Input 1	Output
0	1
1	0

AND Gate

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

OR Gate

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

XOR Gate

Input 1	Input 2	Output
0	0	0

Input 1	Input 2	Output
0	1	1
1	0	1
1	1	0

NAND Gate

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

Classical Computation

1. Anatomy of a Computer
2. Big O Notation

Quantum Computation

1. The Qubit
2. Quantum Computing: Past, Present, Future

Programming Exercises

- Recursive XOR Problem

---

1. Girvin, Steven M. *Introduction to Quantum Information, Computation and Communication PHYS 345 Spring 2021*. 11 May 2021.↩↩↩
2. Veritasium. *What Is NOT Random?* 16 Jul 2014. YouTube, https://www.youtube.com/watch?v=sMb00lz-IfE.↩