Meeting 1: Talking Drums, Babbage and Lovelace, the Political Economy of Mathematics, and the "Laws of Thought"

CS198: The Poetry of Computer Science, the Computer Science of Poetry
Philosophy of Computation at Berkeley
pocab.org

September 18, 2017

1 Drums that Talk

Make your feet come back the way they went,
Make your legs come back the way they went,
Plant your feet and your legs below,
In the village which belongs to us.
–*The Information*, p13

In this section, Gleick discusses African talking drums, Morse code, and a formal definition of information.

- The talking drums convey information. So does Morse code. How are they different? Specifically, consider how Morse code undergoes several different layers of encoding before delivering meaning: from the code of dots and dashes, to letters in the alphabet, to words, to phrases, and finally to meaning. Do messages in talking drums do something similar? If so, how? If not, why not? It may help to consider the following quote in this context:

  "*Allocate extra bits for disambiguation and error correction ...* is what the drum langauge did. Redundancy – inefficient by definition – serves as the antitode to confusion. It provides second chances."

- A lipogram is a type of constrained writing where the writer omits a certain letter, or letters, of the alphabet. Below is an excerpt from Douglas Hofstadter's "Autoportrait with Constraint, or, Vita in Form of a Lipogram", without the letter "e". How is a lipogram similar to a talking drums message?

  > At thirty-two, with my book on its way but still not out, I took a job at Indiana U. in Bloomington, thanks in part to its famous music school, and also to its florid, woodsy campus, but most of all to its warmth and cordiality. "Go for folks who go for you!", was my Dads simplistic but catchy motto (I'm paraphrasing his words to adapt to this situation, naturally, but that was its gist) – and I took his tip, for though it was corny, it was sagacious, too.
  >
  > At IU, my goal was to work in AI, most of all trying to mimic faithfully, in programs, how thought actually works. Crucial to my philosophy of computationally mimicking a mind was my constant focus on how humans think – which is to say, fluidly but also fallibly that is, not logically, but analogically. Also, I was scrambling madly to finish up my big book – a most unusual book, flip-flopping back and forth from fanciful contrapuntal dialogs – canonical and fugal to fairly straightforward monographical writings, and also chock-full of mind-twisting prints by an almost unknown paradox-loving Dutch graphic artist. Upon publication, my book was a surprisingly big hit and won a major national book award, assuring my job stability. I was thirty-four (or so), and still high and dry.
  >
  > But I'd had a hunch that IU was promising in that most chancy of all domains, and in fact, I was right. I was oh-so-lucky to bump fortuitously into Carol Ann Brush in an auditorium lobby during a film. Carol was an Italian and art-history major doing grad work in librarianship. My oh my! Although our liaison had a bit of a bumpy start, Carol and I had a lot

in common and soon hit it off in grand fashion. Thus, at long last – at thirty-six – I had a most happy romantic affair. What a turning point!
– (https://prelectur.stanford.edu/lecturers/hofstadter/autolipography.html)

- Consider the equation

$$H = n \log s$$

H is the amount of information.
n is the number of symbols in the message.
s is the number of symbols in the language.

Why is the equation the way it is? Lets look at the part where it says $\log s$ first.

s is the number of symbols in the language. In the alphabet, for example, there are 26 symbols. Suppose I only have a bunch of sticks and stones, and need to write a poem with them. I can scream at the top of my lungs the following:

00000000000000000000000000 corresponds to "a";
00000000000000000000000010 corresponds to "b";
00000000000000000000000100 corresponds to "c";
00000000000000000000001000 corresponds to "d";
00000000000000000000010000 corresponds to "e";

And so on. Then, if I wanted to write "babe", I would arrange the sticks and stones like 00000000000000000000000010 00000000000000000000000000 00000000000000000000000010 00000000000000000000010000.

In this scheme, I need 26 sticks or stones to represent each letter of the alphabet. But do I really need this many? No, in fact, just 5 sticks or stones suffice for each letter, because the following thirty-two combinations are possible:

00000, 00001, 00010, 00011, 00100, 00101, 00110, 00111, 01000, 01001, 01010, 01011, 01100, 01101, 01110, 01111, 10000, 10001, 10010, 10011, 10100, 10101, 10110, 10111, 11000, 11001, 11010, 11011, 11100, 11101, 11110, 11111

So I can scream instead

00000 corresponds to "a";
00001 corresponds to "b";
00010 corresponds to "c";
00011 corresponds to "d";
00100 corresponds to "e";

And so on. Then I can write "babe" much more easily: 00001 00000 00001 00100. How did I know that 5 sticks or stones suffice? Because $\log 26 = 4.7$.

Now, lets understand the equation as a whole.

$$H = n \log s$$

H is the amount of information.
n is the number of symbols in the message.
s is the number of symbols in the language.

$\log s$ is the number of sticks or stones one needs to write one symbol in the language. There are n of those symbols in the message, so we multiply $\log s$ with n. Which is the amount of *information* ... whatever that means.

Im belaboring this point because it is important to understand how logarithms are related to the notion of "reference". An alphabet has 26 characters, and one might think I would need 26 sticks or stones to refer to a single character; however, I need only log 26 sticks or stones, which is exponentially less than 26.

Similarly, the number 1,000,000 is a gigantic quantity which, if I wanted to count up to it, would take several days, but I can refer to the number with exponentially less effort. By exponentially less effort I

mean this. If I refer to the number 1, it takes exactly the same amount of time to refer to it as it does to count up to it. If I refer to the number 10, it takes less amount of time to refer to it as it does to count up to it, but not by an inordinate amount. If I refer to the number 100, the difference between the amount of time it takes to count up to it and the amount of time it takes to refer to it is pretty substantial. With 1,000, 10,000, and so on, the difference is ginormous.

The central questions in theoretical computer science and the philosophy of computation may hinge on something that we do not understand about the distinction between n and 2^n; in other words, between $\log n$ and n; so to speak, between a *reference of a thing* and a *thing*. Profound concepts like P vs. NP, uncomputability, Hofstadter's Strange Loop, and even differences in cultural patterns of thinking, may all hinge on this core question, as we will discuss in upcoming meetings.

2 To Throw the Powers of Thought into Wheel-Work

Wrong logarithmic tables destroy merchant ships. The British government grants the eccentric genius Babbage with a lot of money to build an infallible that can compute these tables. Babbage works on his Analytical Engine for decades, but ultimately fails. The prestige and political power of mathematics, and its corollary sexism and bigotry, are touched upon. Lovelace develops a friendship with Babbage.

Ada Lovelace, born in 1815, was arguably the first computer scientist, though she didn't call herself that. Instead, she christened herself the "poetical scientist". She was the first person to understand the distinction between a computer and a calculator. That is, while computers churn numbers, the numbers they manipulate may represent something other than numbers, such as music, poetry, (or even) intelligence. She died of cancer at the age of 36. At her deathbed, she mused, *[I will have] the most* harmoniously *disciplined troops; consisting of vast* numbers*, and marching in irresistible power to the sound of* Music*. Is not this very mysterious?...But then, what* are *these Numbers? There is a riddle –*

When she was not in her deathbed, she also said these things:

> [The Analytical Engine][1] might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine...Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.
>
> We may say most aptly, that the Analytical Engine *weaves algebraical patterns* just as the Jacquard-loom weaves flowers and leaves.
>
> I do not believe that my father was such a poet as I shall be an analyst; for with me the two go together indissolubly.

- It seems that this dichotomy between "poet" and "analyst" persists today. (1) Why do you think the dichotomy exists, (2) do you think the dichotomy necessarily exists and so will continue to exist for no less than a million more years, and (3) do you think the dichotomy ought to exist?

- Back in the days in East Asia, it was considered common sense, an obvious fact, an unquestioned assumption, that any educationally enlightened, and thus moral, person should be able to compose beautiful poetry. In fact, one of the biggest qualifications for the Chinese and Korean Civil Service Exam was to compose poetry. Do you think one must be moral to be good at computer science? Similarly, do you think one must be moral to compose good poetry? Do you think there is any inherent difference?

- Take your favorite algorithm, such as mergesort, and write a poem about it no less than five lines with a ABCAC rhyming scheme.

[1] the first computer by the eccentric visionary Charles Babbage, never fully finished

3 A Nervous System for the Earth

> Is it a fact – or have I dreamt it – that, by means of electricity, the world of matter has become a great nerve, vibrating thousands of miles in a breathless point of time? Rather, the round globe is a vast head, a brain, instinct with intelligence! Or, shall we say, it is itself a thought, nothing but thought, and no longer the substance which we deemed it!
> – Nathaniel Hawthorne (1851)

- With this quote in mind, consider last week's question on whether UCBMFET is a conscious organism.

> Pure mathematics was discovered by Boole, in a work which he called the *Laws of Thought*. He was also mistaken in supposing that he was dealing with the laws of thought: the question how people actually think was quite irrelevant to him, and if his book had really contained the laws of thought, it was curious that no one should have ever thought in such a way before."
> – Bertrand Russell

- Do you think Boole indeed discovered the Laws of Thought? In other words, do you believe logic to be the Laws of Thought? Here, we must make a careful distinction, what is sometimes called the "is-ought" problem: that to assert what *is* is distinct from what *ought* to be, but the two are frequently confused. With this problem in mind, what do you think of Boole's ideas?