

# 6

## The Jump to Universality

The earliest writing systems used stylized pictures – ‘pictograms’ – to represent words or concepts. So a symbol like ‘◎’ might stand for ‘sun’, and ‘▲’ for ‘tree’. But no system ever came close to having a pictogram for *every* word in its spoken language. Why not?

Originally, there was no intention to do so. Writing was for specialized applications such as inventories and tax records. Later, new applications would require larger vocabularies, but by then scribes would increasingly have found it easier to add new *rules* to their writing system rather than new pictograms. For example, in some systems, if a word sounded like two or more other words in sequence, it could be represented by the pictograms for those words. If English were written in pictograms, that would allow us to write the word ‘treason’ as ‘▲◎’. This would not represent the sound of the word precisely (nor does its actual spelling, for that matter), but it

would approximate it well enough for any reader who spoke the language and was aware of the rule.

Following that innovation, there would have been less incentive to coin new pictograms – say ‘’ for ‘treason’. Coining one would always have been tedious, not so much because designing memorable pictograms is hard – though it is – but because, before one could use it, one would somehow have to inform all intended readers of its meaning. That is hard to do: if it had been easy, there would have been much less need for writing in the first place. In cases where the rule could be applied instead, it was more efficient: any scribe could write ‘’ and be understood even by a reader who had never seen the word written before.

However, the rule could not be applied in all cases: it could not represent any new single-syllable words, nor many other words. It seems clumsy and inadequate compared to modern writing systems. Yet there was already something significant about it which no purely pictographic system could achieve: it brought words into the writing system that no one had explicitly added. That means that it had reach. And reach always has an explanation. Just as in science a simple formula may summarize a mass of facts, so a

simple, easily remembered rule can bring many additional words into a writing system, but only if it reflects an underlying regularity. The regularity in this case is that all the words in any given language are built out of only a few dozen ‘elementary sounds’, with each language using a different set chosen from the enormous range of sounds that the human voice can produce. Why? I shall come to that below.

As the rules of a writing system were improved, a significant threshold could be crossed: the system could become *universal* for that language – capable of representing every word in it. For example, consider the following variant of the rule that I have just described: instead of building words out of other words, build them out of the *initial sounds* of other words. So, if English were written in pictograms, the new rule would allow ‘treason’ to be spelled with the pictograms for ‘Tent’, ‘Rock’, ‘EAgle’, ‘Zebra’, ‘Nose’. That tiny change in the rules would make the system universal. It is thought that the earliest alphabets evolved from rules like that.

Universality achieved through rules has a different character from that of a completed list (such as the hypothetical complete set of pictograms). One difference is that the rules can be much simpler than the list. The individual

symbols can be simpler too, because there are fewer of them. But there is more to it than that. Since a rule works by exploiting regularities in the language, it implicitly encodes those regularities, and so contains more knowledge than the list. An alphabet, for instance, contains knowledge of what words sound like. That allows it to be used by a foreigner to learn to *speak* the language, while pictograms could at most be used to learn to write it. Rules can also accommodate inflections such as prefixes and suffixes without adding complexity to the writing system, thus allowing written texts to encode more of the grammar of sentences. Also, a writing system based on an alphabet can cover not only every word but every *possible* word in its language, so that words that have yet to be coined already have a place in it. Then, instead of each new word temporarily breaking the system, the system can itself be used to coin new words, in an easy and decentralized way.

Or, at least, it could have been. It would be nice to think that the unknown scribe who created the first alphabet knew that he was making one of the greatest discoveries of all time. But he may not have. If he did, he certainly failed to pass his enthusiasm on to many others. For, in the event, the power of universality that I have just

described was rarely used in ancient times, even when it was available. Although pictographic writing systems were invented in many societies, and universal alphabets did sometimes evolve from them in the way I have just described, the ‘obvious’ next step – namely to use the alphabet universally and to drop the pictograms – was almost never taken. Alphabets were confined to special purposes such as writing rare words or transliterating foreign names. Some historians believe that the idea of an *alphabet-based* writing system was conceived only once in human history – by some unknown predecessors of the Phoenicians, who then spread it throughout the Mediterranean – so that every alphabet-based writing system that has ever existed is either descended from or inspired by that Phoenician one. But even the Phoenician system had no vowels, which diminished some of the advantages I have mentioned. The Greeks added vowels.

It is sometimes suggested that scribes deliberately limited the use of alphabets for fear that their livelihoods would be threatened by a system that was too easy to learn. But perhaps that is forcing too modern an interpretation on them. I suspect that neither the opportunities nor the pitfalls of universality ever occurred to anyone until much later in history. Those ancient

innovators only ever cared about the specific problems they were confronting – to write particular words – and, in order to do that, one of them invented a rule that happened to be universal. Such an attitude may seem implausibly parochial. But things *were* parochial in those days.

And indeed it seems to be a recurring theme in the early history of many fields that universality, when it was achieved, was not the primary objective, if it was an objective at all. A small change in a system to meet a parochial purpose just happened to make the system universal as well. This is the *jump to universality*.

Just as writing dates back to the dawn of civilization, so do *numerals*. Mathematicians nowadays distinguish between *numbers*, which are abstract entities, and *numerals*, which are physical symbols that represent numbers; but numerals were discovered first. They evolved from ‘tally marks’ (||||, ||||, ||||, . . .) or tokens such as stones, which had been used since prehistoric times to keep track of discrete entities such as animals or days. If one made a mark for each goat released from a pen, and later crossed one out for each goat that returned, then one would have retrieved all the goats when one had crossed out all the marks.

That is a universal system of tallying. But, like

levels of emergence, there is a hierarchy of universality. The next level above tallying is counting, which involves numerals. When tallying goats one is merely thinking ‘another, and another, and another’; but when counting them one is thinking ‘forty, forty-one, forty-two . . . ’

It is only with hindsight that we can regard tally marks as a system of numerals, known as the ‘unary’ system. As such, it is an impractical system. For instance, even the simplest operations on numbers represented by tally marks, such as comparing them, doing arithmetic, and even just copying them, involves repeating the entire tallying process. If you had forty goats, and sold twenty, and had tally-mark records of both those numbers, you would still have to perform twenty individual deletion operations to bring your record up to date. Similarly, checking whether two fairly close numerals were the same would involve tallying them against each other. So people began to improve the system. The earliest improvement may have been simply to group the tally marks – for instance, writing  instead of . This made arithmetic and comparison easier, since one could tally whole groups and see at a glance that  is different from . Later, such groups were themselves represented by shorthand symbols: the ancient Roman system

used symbols like I, V, X,  $\text{V}$ , C, D, and  $\text{DD}$  to represent one, five, ten, fifty, one hundred, five hundred, and one thousand. (So they were not quite the same as the ‘Roman numerals’ we use today.)

So this was another story of incremental improvements intended to solve specific, parochial problems. And, again, it seems that no one aspired to anything more. Even though adding simple rules could make the system much more powerful, and even though the Romans did occasionally add some such rules, they did this without ever aiming for, or achieving, universality. For some centuries, the rules of their system were:

- Placing symbols side by side means adding them together. (This rule was inherited from the tally-mark system.)
- Symbols must be written in order of decreasing value from left to right; and
- Adjacent symbols must be replaced by the symbol for their combined value whenever possible.

(The subtractive rule in today’s ‘Roman numerals’, where IV represents four, was introduced later.) The second and third rules ensure that each number has only one representation, which makes comparison much

easier. Without them, **XIXIXIXIXIX** and **VXVXVXVXV** would both be valid numerals, and one could not tell at a glance that they represent the same number.

By exploiting the universal laws of addition, those rules gave the system some important reach beyond tallying – such as the ability to perform arithmetic. For example, consider the numbers seven (**VII**) and eight (**VIII**). The rules say that placing them side by side – **VIIVIII** – is the same as adding them. Then they tell us to rearrange the symbols in order of decreasing value: **VVIIII**. Then they tell us to replace the two **V**'s by **X**, and the five **I**'s by **V**. The result is **XV**, which is the representation of fifteen. Something new has happened here, which is more than just a matter of shorthand: an abstract truth has been discovered, and proved, about seven, eight and fifteen without anyone having counted or tallied anything. Numbers have been manipulated in their own right, via their numerals.

I mean it literally when I say that it was the *system of numerals* that performed arithmetic. The human users of the system did of course physically enact those transformations. But to do that, they first had to encode the system's rules somewhere in their brains, and then they had to execute them as a computer executes its program.

And it is the program that instructs its computer what to do, not vice versa. Hence the process that we call ‘using Roman numerals to do arithmetic’ also consists of the Roman-numeral system using *us* to do arithmetic.

It was only by causing people to do this that the Roman-numeral system survived – that is to say, caused itself to be copied from generation to generation of Romans: they found it useful, so they passed it on to their offspring. As I have said, knowledge is information which, when it is physically embodied in a suitable environment, tends to cause itself to remain so.

To speak of the Roman-numeral system as controlling us in order to get itself replicated and preserved may sound like relegating humans to the status of slaves. But that would be a misconception. People *consist* of abstract information, including the distinctive ideas, theories, intentions, feelings and other states of mind that characterize an ‘I’. To object to being ‘controlled’ by Roman numerals when we find them helpful is like protesting at being controlled by one’s own intentions. By that argument, it is slavery to escape from slavery. But in fact when I obey the program that constitutes me (or when I obey the laws of physics), ‘obey’ means something different from what a slave does. The two

meanings explain events at different levels of emergence.

Contrary to what is sometimes said, there were also fairly efficient ways of multiplying and dividing Roman numerals. So a ship with XX crates, each containing jars in a V-by-VII grid, could be known to hold ⸿CCC jars altogether without anyone having performed the lengthy count that was implicit in that numeral. And one could tell at a glance that ⸿CCC was less than ⸿CCI. Thus, manipulating numbers independently of tallying or counting opened up applications such as calculating prices, wages, taxes, interest rates and so on. It was also a conceptual advance that opened the door to future progress. However, in regard to these more sophisticated applications, the system was not universal. Since there was no higher-valued symbol than ⸿ (one thousand), the numerals from two thousand onwards all began with a string of ⸿'s, which therefore became nothing more than tally marks for thousands. The more of them there were in a numeral, the more one would have to fall back on tallying (examining many instances of the symbol one by one) in order to do arithmetic.

Just as one could upgrade the vocabulary of an ancient writing system by adding pictograms, so one could add symbols to a system of numerals to

increase its range. And this was done. But the resulting system would still always have a highest-valued symbol, and hence would not be universal for doing arithmetic without tallying.

The only way to emancipate arithmetic from tallying is with rules of universal reach. As with alphabets, a small set of basic rules and symbols is sufficient. The universal system in general use today has ten symbols, the digits 0 to 9, and its universality is due to a rule that the value of a digit depends on its position in the number. For instance, the digit 2 means two when written by itself, but means two hundred in the numeral 204. Such ‘positional’ systems require ‘placeholders’, such as the digit 0 in 204, whose only function is to place the 2 into the position where it means two hundred.

This system originated in India, but it is not known when. It might have been as late as the ninth century, since before that only a few ambiguous documents seem to show it in use. At any rate, its tremendous potential in science, mathematics, engineering and trade was not widely realized. At approximately that time it was embraced by Arab scholars, yet was not generally used in the Arab world until a thousand years later. This curious lack of enthusiasm for universality was repeated in medieval Europe: a

few scholars adopted Indian numerals from the Arabs in the tenth century (resulting in the misnomer ‘Arabic numerals’), but again these numerals did not come into everyday use for centuries.

As early as 1900 BCE the ancient Babylonians had invented what was in effect a universal system of numerals, but they too may not have cared about its universality – nor even been aware of it. It was a positional system, but very cumbersome compared with the Indian one. It had 59 ‘digits’, each of which was itself written as a numeral in a Roman-numeral-like system. So using it for arithmetic with numbers occurring in everyday life was actually more complicated than using Roman numerals. It also had no symbol for zero, so it used spaces as placeholders. It had no way of representing trailing zeros, and no equivalent of the decimal point (as if, in our system, the numbers 200, 20, 2, 0.2 and so on were all written as 2, and were distinguished only by context). All this suggests that universality was not the system’s main design objective, and that it was not greatly valued when it was achieved.

Perhaps an insight into this recurring oddity is provided by a remarkable episode in the third century BCE involving the ancient Greek scientist and mathematician Archimedes. His research in

astronomy and pure mathematics led him to a need to do arithmetic with some rather large numbers, so he had to invent his own system of numerals. His starting point was a Greek system with which he was familiar, similar to the Roman one but with a highest-valued symbol  $\text{M}$  for 10,000 (one myriad). The range of the system had already been extended with the rule that digits written above an  $\text{M}$  would be multiplied by a myriad. For instance, the symbol for twenty was  $\kappa$  and the symbol for four was  $\delta$ , so they could write twenty-four myriad (240,000) as  $\overset{\kappa\delta}{\text{M}}$ .

If only they had allowed that rule to generate multi-tier numerals, so that  $\overset{\kappa\delta}{\text{M}}$  would mean twenty-four myriad myriad, the system would have been universal. But apparently they never did. Even more surprisingly, nor did Archimedes. His system used a different idea, similar to modern ‘scientific notation’ (in which, say, two million is written  $2 \times 10^6$ ), except that instead of powers of ten it used powers of a myriad myriad. But, again, he then required the exponent (the power to which the myriad myriad was raised) to be an existing Greek numeral – that is to say, it could not easily exceed a myriad myriad or so. Hence this construction petered out after the number that we call  $10^{800,000,000}$ . If only he had

not imposed that additional rule, he would have had a universal system, albeit an unnecessarily awkward one.

Even today, only mathematicians ever need numbers above  $10^{800,000,000}$ , and only rarely at that. But that cannot be why Archimedes imposed the restriction, for he did not stop there. Exploring the concept of numbers further, he set up yet another extension, this time amounting to an even more unwieldy system with base  $10^{800,000,000}$ . Yet, once again, he allowed this number to be raised only to powers not exceeding 800,000,000, thus imposing an arbitrary limit somewhere in excess of  $10^{6.4 \times 10^{17}}$ .

Why? Today it seems very perverse of Archimedes to have placed limits on which symbols could be used at which positions in his numerals. There is no mathematical justification for them. But, if Archimedes had been willing to allow his rules to be applied without arbitrary limits, he could have invented a much better universal system just by removing the arbitrary limits from the existing Greek system. A few years later the mathematician Apollonius invented yet another system of numerals which fell short of universality for the same reason. It is as though everyone in the ancient world was avoiding

universality on purpose.

The mathematician Pierre Simon Laplace (1749–1827) wrote, of the Indian system, ‘We shall appreciate the grandeur of this achievement when we remember that it escaped the genius of Archimedes and Apollonius, two of the greatest minds produced by antiquity.’ But was this really something that escaped them, or something that they chose to steer clear of? Archimedes must have been aware that his method of extending a number system – which he used twice in succession – could be continued indefinitely. But perhaps he doubted that the resulting numerals would refer to anything about which one could validly reason. Indeed, one motivation for that whole project was to contradict the idea – which was a truism at the time – that the grains of sand on a beach could literally not be numbered. So he used his system to calculate the number of grains of sand that would be needed to fill the entire celestial sphere. This suggests that he, and ancient Greek culture in general, may not have had the concept of an abstract number at all, so that, for them, numerals could refer only to objects – if only objects of the imagination. In that case universality would have been a difficult property to grasp, let alone to aspire to. Or maybe he merely felt that he had to avoid aspiring to

infinite reach in order to make a convincing case. At any rate, although from our perspective Archimedes' system repeatedly 'tried' to jump to universality, he apparently did not want it to.

Here is an even more speculative possibility. The largest benefits of any universality, beyond whatever parochial problem it is intended to solve, come from its being useful for further innovation. And innovation is unpredictable. So, to appreciate universality at the time of its discovery, one must either value abstract knowledge for its own sake or expect it to yield unforeseeable benefits. In a society that rarely experienced change, both those attitudes would be quite unnatural. But that was reversed with the Enlightenment, whose quintessential idea is, as I have said, that *progress* is both desirable and attainable. And so, therefore, is universality.

Be that as it may, with the Enlightenment, parochialism and all arbitrary exceptions and limitations began to be regarded as inherently problematic – and not only in science. Why should the law treat an aristocrat differently from a commoner? A slave from a master? A woman from a man? Enlightenment philosophers such as Locke set out to free political institutions from arbitrary rules and assumptions. Others tried to derive moral maxims from universal moral explanations

rather than merely to postulate them dogmatically. Thus universal explanatory theories of justice, legitimacy and morality began to take their place alongside universal theories of matter and motion. In all those cases, universality was being sought deliberately, as a desirable feature in its own right – even a necessary feature for an idea to be true – and not just as a means of solving a parochial problem.

A jump to universality that played an important role in the early history of the Enlightenment was the invention of *movable-type printing*. Movable type consisted of individual pieces of metal, each embossed with one letter of the alphabet. Earlier forms of printing had merely streamlined writing in the same way that Roman numerals streamlined tallying: each page was engraved on a printing plate and thus all the symbols on it could be copied in a single action. But, given a supply of movable type with several instances of each letter, one does no further metalwork. One merely arranges the type into words and sentences. One does not have to know, in order to manufacture type, what the documents that it will eventually print are going to say: it is universal.

Even so, movable type did not make much difference when it was invented in China in the eleventh century, perhaps because of the usual

lack of interest in universality, or perhaps because the Chinese writing system used thousands of pictograms, which diminished the immediate advantages of a universal printing system. But when it was reinvented by the printer Johannes Gutenberg in Europe in the fifteenth century, using alphabetic type, it initiated an avalanche of further progress.

Here we see a transition that is typical of the jump to universality: before the jump, one has to make specialized objects for each document to be printed; after the jump, one customizes (or specializes, or programs) a universal object – in this case a printing press with movable type. Similarly, in 1801 Joseph Marie Jacquard invented a general-purpose silk-weaving machine now known as the Jacquard loom. Instead of having to control manually each row of stitches in each individual bolt of patterned silk, one could program an arbitrary pattern on punched cards which would instruct the machine to weave that pattern any number of times.

The most momentous such technology is that of *computers*, on which an increasing proportion of all technology now depends, and which also has deep theoretical and philosophical significance. The jump to computational universality *should* have happened in the 1820s, when the

mathematician Charles Babbage designed a device that he called the *Difference Engine* – a mechanical calculator which represented decimal digits by cogs, each of which could click into one of ten positions. His original purpose was parochial: to automate the production of tables of mathematical functions such as logarithms and cosines, which were heavily used in navigation and engineering. At the time, they were compiled by armies of clerks known as ‘computers’ (which is the origin of the word), and were notoriously error-prone. The Difference Engine would make fewer errors, because the rules of arithmetic would be built into its hardware. To make it print out a table of a given function, one would program it only once with the definition of the function in terms of simple operations. In contrast, human ‘computers’ had to use (or be used by) both the definition and the general rules of arithmetic thousands of times per table, each time being an opportunity for human error.

Unfortunately, despite pouring a fortune of his own money and that of the British government into the project, Babbage was such a poor organizer that he never succeeded in building a Difference Engine. But his design was sound (apart from a few trivial mistakes), and in 1991 a team led by the engineer Doron Swade at London’s

Science Museum successfully implemented it, using engineering tolerances achievable in Babbage's time.

By the standards of today's computers and even calculators, the Difference Engine had an extremely limited repertoire. But the reason it could exist at all is that there is a regularity among all the mathematical functions that occur in physics, and hence in navigation and engineering. These are known as *analytic functions*, and in 1710 the mathematician Brook Taylor had discovered that they can all be approximated arbitrarily well using only repeated additions and multiplications – the operations that the Difference Engine performs. (Special cases had been known before that, but the jump to universality was proved by Taylor.) Thus, to solve the parochial problem of computing the handful of functions that needed to be tabulated, Babbage created a calculator that was universal for calculating analytic functions. It also made use of the universality of movable type, in its typewriter-like printer, without which the process of printing the tables could not have been fully automated.

Babbage originally had no conception of computational universality. Nevertheless, the Difference Engine already comes remarkably close to it – not in its repertoire of computations, but in

its physical constitution. To program it to print out a given table, one initializes certain cogs. Babbage eventually realized that this programming phase could itself be automated: the settings could be prepared on punched cards like Jacquard's, and transferred mechanically into the cogs. This would not only remove the main remaining source of error, but also increase the machine's repertoire. Babbage then realized that if the machine could also punch new cards for its own later use, and could control which punched card it would read next (say, by choosing from a stack of them, depending on the position of its cogs), then something qualitatively new would happen: the jump to universality.

Babbage called this improved machine the *Analytical Engine*. He and his colleague the mathematician Ada, Countess of Lovelace, knew that it would be capable of computing anything that human ‘computers’ could, and that this included more than just arithmetic: it could do algebra, play chess, compose music, process images and so on. It would be what is today called a universal classical computer. (I shall explain the significance of the proviso ‘classical’ in [Chapter 11](#), when I discuss quantum computers, which operate at a still higher level of universality.)

Neither they nor anyone else for over a century

afterwards imagined today's most common uses of computation, such as the internet, word processing, database searching, and games. But another important application that they did foresee was making scientific predictions. The Analytical Engine would be a universal simulator – able to predict the behaviour, to any desired accuracy, of any physical object, given the relevant laws of physics. This is the universality that I mentioned in [Chapter 3](#), through which physical objects that are unlike each other and dominated by different laws of physics (such as brains and quasars) can exhibit the same mathematical relationships.

Babbage and Lovelace were Enlightenment people, and so they understood that the universality of the Analytical Engine would make it an epoch-making technology. Even so, despite great efforts, they failed to pass their enthusiasm on to more than a handful of others, who in turn failed to pass it to anyone. And so the Analytical Engine became one of the tragic might-have-beens of history. If only they had looked around for other implementations, they might have realized that the perfect one was already waiting for them: electrical relays (switches controlled by electric currents). These had been one of the first applications of fundamental research into

electromagnetism, and they were about to be mass produced for the technological revolution of telegraphy. A redesigned Analytical Engine, using on/off electrical currents to represent binary digits and relays to do the computation, would have been faster than Babbage's and also cheaper and easier to construct. (Binary numbers were already well known. The mathematician and philosopher Gottfried Wilhelm Leibniz had even suggested using them for mechanical calculation in the seventeenth century.) So the computer revolution would have happened a century earlier than it did. Because of the technologies of telegraphy and printing that were being developed concurrently, an internet revolution might well have followed. The science-fiction authors William Gibson and Bruce Sterling, in their novel *The Difference Engine*, have given an exciting account of what that might have been like. The journalist Tom Standage, in his book *The Victorian Internet*, maintains that the early telegraph system, even without computers, did create an internet-like phenomenon among the operators, with 'hackers, on-line romances and weddings, chat-rooms, flame wars . . . and so on'.

Babbage and Lovelace also thought about one application of universal computers that has not been achieved to this day, namely so-called *artificial intelligence* (AI). Since human brains are

physical objects obeying the laws of physics, and since the Analytical Engine is a universal simulator, it could be programmed to think, in every sense that humans can (albeit very slowly and requiring an impractically vast number of punched cards). Nevertheless, Babbage and Lovelace denied that it could. Lovelace argued that ‘The Analytical Engine has no pretensions whatever to originate anything. It can do whatever we know how to order it to perform. It can follow analysis; but it has no power of anticipating any analytical relations or truths.’

The mathematician and computer pioneer Alan Turing later called this mistake ‘Lady Lovelace’s objection’. It was not computational universality that Lovelace failed to appreciate, but the universality of the laws of physics. Science at the time had almost no knowledge of the physics of the brain. Also, Darwin’s theory of evolution had not yet been published, and supernatural accounts of the nature of human beings were still prevalent. Today there is less mitigation for the minority of scientists and philosophers who still believe that AI is unattainable. For instance, the philosopher John Searle has placed the AI project in the following historical perspective: for centuries, some people have tried to explain the mind in mechanical terms, using similes and metaphors

based on the most complex machines of the day. First the brain was supposed to be like an immensely complicated set of gears and levers. Then it was hydraulic pipes, then steam engines, then telephone exchanges – and, now that computers are our most impressive technology, brains are said to be computers. But this is still no more than a metaphor, says Searle, and there is no more reason to expect the brain to be a computer than a steam engine.

But there is. A steam engine is not a universal simulator. But a computer is, so expecting it to be able to do whatever neurons can is not a metaphor: it is a known and proven property of the laws of physics as best we know them. (And, as it happens, hydraulic pipes could also be made into a universal classical computer, and so could gears and levers, as Babbage showed.)

Ironically, Lady Lovelace's objection has almost the same logic as Douglas Hofstadter's argument for reductionism ([Chapter 5](#)) – yet Hofstadter is one of today's foremost *proponents* of the possibility of AI. That is because both of them share the mistaken premise that low-level computational steps cannot possibly add up to a higher-level 'I' that affects anything. The difference between them is that they chose opposite horns of the dilemma that that poses:

Lovelace chose the false conclusion that AI is impossible, while Hofstadter chose the false conclusion that no such ‘I’ can exist.

Because of Babbage’s failure either to build a universal computer or to persuade others to do so, an entire century would pass before the first one was built. During that time, what happened was more like the ancient history of universality: although calculating machines similar to the Difference Engine were being built by others even before Babbage had given up, the Analytical Engine was almost entirely ignored even by mathematicians.

In 1936 Turing developed the definitive theory of universal classical computers. His motivation was not to build such a computer, but only to use the theory abstractly to study the nature of mathematical proof. And when the first universal computers were built, a few years later, it was, again, not out of any special intention to implement universality. They were built in Britain and the United States during the Second World War for specific wartime applications. The British computers, named Colossus (in which Turing was involved), were used for code-breaking; the American one, ENIAC, was designed to solve the equations needed for aiming large guns. The technology used in both was electronic vacuum

tubes, which acted like relays but about a hundred times as fast. At the same time, in Germany, the engineer Konrad Zuse was building a programmable calculator out of relays – just as Babbage should have done. All three of these devices had the technological features necessary to be a universal computer, but none of them was quite configured for this. In the event, the Colossus machines never did anything but code-breaking, and most were dismantled after the war. Zuse's machine was destroyed by Allied bombing. But ENIAC *was* allowed to jump to universality: after the war it was put to diverse uses for which it had never been designed, such as weather forecasting and the hydrogen-bomb project.

The history of electronic technology since the Second World War has been dominated by miniaturization, with ever more microscopic switches being implemented in each new device. These improvements led to a jump to universality in about 1970, when several companies independently produced a microprocessor, a universal classical computer on a single silicon chip. From then on, designers of *any* information-processing device could start with a microprocessor and then customize it – program it – to perform the specific tasks needed for that device. Today, your washing machine is almost

certainly controlled by a computer that could be programmed to do astrophysics or word processing instead, if it were given suitable input-output devices and enough memory to hold the necessary data.

It is a remarkable fact that, in that sense (that is to say, ignoring issues of speed, memory capacity and input-output devices), the human ‘computers’ of old, the steam-powered Analytical Engine with its literal bells and whistles, the room-sized vacuum-tube computers of the Second World War, and present-day supercomputers all have an identical repertoire of computations.

Another thing that they have in common is that they are all *digital*: they operate on information in the form of discrete values of physical variables, such as electronic switches being on or off, or cogs being at one of ten positions. The alternative, ‘analogue’, computers, such as slide rules, which represent information as continuous physical variables, were once ubiquitous but are hardly ever used today. That is because a modern digital computer can be programmed to imitate any of them, and to outperform them in almost any application. The jump to universality in digital computers has left analogue computation behind. That was inevitable, because there is no such thing as a universal analogue computer.

That is because of the need for *error correction*: during lengthy computations, the accumulation of errors due to things like imperfectly constructed components, thermal fluctuations, and random outside influences makes analogue computers wander off the intended computational path. This may sound like a minor or parochial consideration. But it is quite the opposite. Without error-correction all information processing, and hence all knowledge-creation, is necessarily bounded. Error-correction is the beginning of infinity.

For example, tallying is universal only if it is digital. Imagine that some ancient goatherds had tried to tally the total *length* of their flock instead of the number. As each goat left the enclosure, they could reel out some string of the same length as the goat. Later, when the goats returned, they could reel that length back in. When the whole length had been reeled back in, that would mean that all the goats had returned. But in practice the outcome would always be at least a little long or short, because of the accumulation of measurement errors. For any given accuracy of measurement, there would be a maximum number of goats that could be reliably tallied by this ‘analogue tallying’ system. The same would be true of all arithmetic performed with those

'tallies'. Whenever the strings representing several flocks were added together, or a string was cut in two to record the splitting of a flock, and whenever a string was 'copied' by making another of the same length, there would be errors. One could mitigate their effect by performing each operation many times, and then keeping only the outcome of median length. But the operations of comparing or duplicating lengths can themselves be performed only with finite accuracy, and so could not reduce the rate of error accumulation per step below that level of accuracy. That would impose a maximum number of consecutive operations that could be performed before the result became useless for a given purpose – which is why analogue computation can never be universal.

What is needed is a system that takes for granted that errors will occur, but *corrects* them once they do – a case of 'problems are inevitable, but they are soluble' at the lowest level of information-processing emergence. But, in analogue computation, error correction runs into the basic logical problem that there is no way of distinguishing an erroneous value from a correct one at sight, because it is in the very nature of analogue computation that every value *could* be correct. Any length of string might be the right

length.

And that is not so in a computation that confines itself to whole numbers. Using the same string, we might represent whole numbers as lengths of string in whole numbers of inches. After each step, we trim or lengthen the resulting strings to the nearest inch. Then errors would no longer accumulate. For example, suppose that the measurements could all be done to a tolerance of a tenth of an inch. Then all errors would be detected and eliminated after each step, which would eliminate the limit on the number of consecutive steps.

So all universal computers are digital; and all use error-correction with the same basic logic that I have just described, though with many different implementations. Thus Babbage's computers assigned only ten different meanings to the whole continuum of angles at which a cogwheel might be oriented. Making the representation digital in that way allowed the cogs to carry out error-correction automatically: after each step, any slight drift in the orientation of the wheel away from its ten ideal positions would immediately be corrected back to the nearest one as it clicked into place. Assigning meanings to the whole continuum of angles would nominally have allowed each wheel to carry (infinitely) more

information; but, in reality, information that cannot be reliably retrieved is not really being stored.

Fortunately, the limitation that the information being processed must be digital does not detract from the universality of digital computers – or of the laws of physics. If measuring the goats in whole numbers of inches is insufficient for a particular application, use whole numbers of *tenths* of inches, or billionths. The same holds for all other applications: the laws of physics are such that the behaviour of any physical object – and that includes any other computer – can be simulated with any desired accuracy by a universal digital computer. It is just a matter of approximating continuously variable quantities by a sufficiently fine grid of discrete ones.

Because of the necessity for error-correction, *all* jumps to universality occur in digital systems. It is why spoken languages build words out of a finite set of elementary sounds: speech would not be intelligible if it were analogue. It would not be possible to repeat, nor even to remember, what anyone had said. Nor, therefore, does it matter that universal writing systems cannot perfectly represent analogue information such as tones of voice. Nothing can represent those perfectly. For the same reason, the sounds themselves can

represent only a finite number of possible meanings. For example, humans can distinguish between only about seven different sound volumes. This is roughly reflected in standard musical notation, which has approximately seven different symbols for loudness (such as *p*, *mf*, *f*, and so on). And, for the same reason, speakers can only *intend* a finite number of possible meanings with each utterance.

Another striking connection between all those diverse jumps to universality is that they all happened on Earth. In fact all known jumps to universality happened under the auspices of human beings – except one, which I have not mentioned yet, and from which all the others, historically, emerged. It happened during the early evolution of life.

Genes in present-day organisms replicate themselves by a complicated and very indirect chemical route. In most species they act as templates for forming stretches of a similar molecule, RNA. Those then act as programs which direct the synthesis of the body's constituent chemicals, especially enzymes, which are *catalysts*. A catalyst is a kind of constructor – it promotes a change among other chemicals while remaining unchanged itself. Those catalysts in turn control all the chemical production and regulatory

functions of an organism, and hence define the organism itself, crucially including a process that makes a copy of the DNA. How that intricate mechanism evolved is not essential here, but for definiteness let me sketch one possibility.

About four billion years ago – soon after the surface of the Earth had cooled sufficiently for liquid water to condense – the oceans were being churned by volcanoes, meteor impacts, storms and much stronger tides than today's (because the moon was closer). They were also highly active chemically, with many kinds of molecules being continually formed and transformed, some spontaneously and some by catalysts. One such catalyst happened to catalyse the formation of some of the very kinds of molecules from which it itself was formed. That catalyst was not alive, but it was the first hint of life.

It had not yet evolved to be a well-targeted catalyst, so it also accelerated the production of some other chemicals, including variants of itself. Those that were best at promoting their own production (and inhibiting their own destruction) relative to other variants became more numerous. They too promoted the construction of variants of themselves, and so evolution continued.

Gradually, the ability of these catalysts to promote their own production became robust and

specific enough for it to be worth calling them replicators. Evolution produced replicators that caused themselves to be replicated ever faster and more reliably.

Different replicators began to join forces in groups, each of whose members specialized in causing one part of a complex web of chemical reactions whose net effect was to construct more copies of the entire group. Such a group was a rudimentary organism. At that point, life was at a stage roughly analogous to that of non-universal printing, or Roman numerals: it was no longer a case of each replicator for itself, but there was still no universal system being customized or programmed to produce specific substances.

The most successful replicators may have been RNA molecules. They have catalytic properties of their own, depending on the precise sequence of their constituent molecules (or bases, which are similar to those of DNA). As a result, the replication process became ever less like straightforward catalysis and ever more like programming – in a language, or genetic code, that used bases as its alphabet.

Genes are replicators that can be interpreted as instructions in a genetic code. Genomes are groups of genes that are dependent on each other for replication. The process of copying a genome is

called a living organism. Thus the genetic code is also a language for specifying organisms. At some point, the system switched to replicators made of DNA, which is more stable than RNA and therefore more suitable for storing large amounts of information.

The familiarity of what happened next can obscure how remarkable and mysterious it is. Initially, the genetic code and the mechanism that interpreted it were both evolving along with everything else in the organisms. But there came a moment when the code stopped evolving yet the organisms continued to do so. At that moment the system was coding for nothing more complex than primitive, single-celled creatures. Yet virtually all subsequent organisms on Earth, to this day, have not only been based on DNA replicators but have used exactly the same alphabet of bases, grouped into three-base ‘words’, with only small variations in the meanings of those ‘words’.

That means that, considered as a language for specifying organisms, the genetic code has displayed phenomenal reach. It evolved only to specify organisms with no nervous systems, no ability to move or exert forces, no internal organs and no sense organs, whose lifestyle consisted of little more than synthesizing their own structural constituents and then dividing in two. And yet the

same language today specifies the hardware and software for countless multicellular behaviours that had no close analogue in those organisms, such as running and flying and breathing and mating and recognizing predators and prey. It also specifies engineering structures such as wings and teeth, and nanotechnology such as immune systems, and even a brain that is capable of explaining quasars, designing other organisms from scratch, and wondering why it exists.

During the entire evolution of the genetic code, it was displaying far less reach. It may be that each successive variant of it was used to specify only a few species that were very similar to each other. At any rate, it must have been a frequent occurrence that a species embodying new knowledge was specified in a new variant of the genetic code. But then the evolution stopped, at a point when it had already attained enormous reach. Why? It looks like a jump to some sort of universality, does it not?

What happened next followed the same sad pattern that I have described in other stories of universality: for well over a billion years after the system had reached universality and stopped evolving, it was *still* only being used to make bacteria. That means that the reach that we can now see that the system had was to remain

unused for longer than the system itself had taken to evolve from non-living precursors. If intelligent extraterrestrials had visited Earth at any time during those billion years they would have seen no evidence that the genetic code could specify anything significantly different from the organisms that it had specified when it first appeared.

Reach always has an explanation. But this time, to the best of my knowledge, the explanation is not yet known. If the reason for the jump in reach was that it was a jump to universality, what was the universality? The genetic code is presumably not universal *for specifying life forms*, since it relies on specific types of chemicals, such as proteins. Could it be a universal constructor? Perhaps. It does manage to build with inorganic materials sometimes, such as the calcium phosphate in bones, or the magnetite in the navigation system inside a pigeon's brain. Biotechnologists are already using it to manufacture hydrogen and to extract uranium from seawater. It can also program organisms to perform constructions outside their bodies: birds build nests; beavers build dams. Perhaps it would be possible to specify, in the genetic code, an organism whose life cycle includes building a nuclear-powered spaceship. Or perhaps not. I guess it has some

lesser, and not yet understood, universality.

In 1994 the computer scientist and molecular biologist Leonard Adleman designed and built a computer composed of DNA together with some simple enzymes, and demonstrated that it was capable of performing some sophisticated computations. At the time, Adleman's DNA computer was arguably the fastest computer in the world. Further, it was clear that a *universal* classical computer could be made in a similar way. Hence we know that, whatever that other universality of the DNA system was, the universality of computation had also been inherent in it for billions of years, without ever being used – until Adleman used it.

The mysterious universality of DNA as a constructor may have been the first universality to exist. But, of all the different forms of universality, the most significant physically is the characteristic universality of people, namely that they are universal explainers, which makes them universal constructors as well. The effects of that universality are, as I have explained, explicable only by means of the full gamut of fundamental explanations. It is also the only kind of universality capable of transcending its parochial origins: universal computers cannot really be universal unless there are people present to

provide energy and maintenance – indefinitely. And the same is true of all those other technologies. Even life on Earth will eventually be extinguished, unless people decide otherwise. Only people can rely on themselves into the unbounded future.

## TERMINOLOGY

*The jump to universality* The tendency of gradually improving systems to undergo a sudden large increase in functionality, becoming universal in some domain.

## MEANINGS OF ‘THE BEGINNING OF INFINITY’ ENCOUNTERED IN THIS CHAPTER

- The existence of universality in many fields.
- The jump to universality.
- Error-correction in computation.
- The fact that people are universal explainers.
- The origin of life.
- The mysterious universality to which the genetic code jumped.

## SUMMARY

All knowledge growth is by incremental improvement, but in many fields there comes a

point when one of the incremental improvements in a system of knowledge or technology causes a sudden increase in reach, making it a universal system in the relevant domain. In the past, innovators who brought about such a jump to universality had rarely been seeking it, but since the Enlightenment they have been, and universal explanations have been valued both for their own sake and for their usefulness. Because error-correction is essential in processes of potentially unlimited length, the jump to universality only ever happens in digital systems.