For centuries, people treated mathematics as untouchable, certain, complete, and built on unshakeable foundations. In 1931, Kurt Gödel's incompleteness theorems shattered this ideal, revealing the truth of the limitations of the most stable branch of human understanding.

To understand what Gödel proved, we must start somewhere unexpected.

You've probably come across the Liar Paradox: "This sentence is false," which is true when false and false when true. It goes around in circles, never making sense. An earlier version of this idea is linked to the prophet Epimenides, a Cretan who supposedly said: "All Cretans are liars." These puzzles matter because they highlight the deep-rooted problem in self-referentiality, which Gödel set out to explore.

Of course, Gödel didn't just play around with paradoxes, he managed to formulate self-reference within arithmetic itself through a technique called Gödel numbering.

Gödel numbering is a way of encoding symbols as numbers, and then using these numbers to formulate new numbers that represent strings of symbols. Mathematical relationships can be found between these numbers, which comment on the relationships between them.

The first step of Gödel's arithmetisation was to inventorise the symbols of the system. (Gödel used a combination of Peano Arithmetic and Principia Mathematica, which he named P). Then, Gödel assigned symbols natural numbers (1,2,3, etc). A familiar example of encoding symbols as numbers is the A1Z26 cipher which substitutes letters A-Z with numbers 1-26. Gödel's own method for doing this was more complex since the language of P is far richer than the alphabet. Gödel went on to arithmetise the entire formal system, including not only the symbols used in the system, but also the rules of inference and syntax, allowing him to convert proofs about numbers into proofs about proofs themselves.

We can encode strings of symbols as follows:

We will assign the capital letters of the alphabet numbers 1-26, give the symbol for zero the number 27, the symbol for equals the number 28, and the blank space the number 29.

Now let's assign a Gödel number to the string: S0 = SS0.

S is the 19th letter of the alphabet, so its Gödel number is 19.

S, the symbols in the string are: S, 0, space, =, space, S, S, 0

The symbols that correspond are: 19, 27, 29, 28, 29, 19, 19, 27

We could now put these numbers together to form the number 1927292829191927 however, this wouldn't be suitable since this number isn't unique. How can we know that 19 encodes an S and not an A followed by an I? Instead, Gödel makes use of the unique factorisation theorem by using the Gödel number of the symbol as the exponent for consecutive primes. So instead of 1927292829191927, we get 2^19 x 3^27 x 5^29 x 7^28 x 11^29 x 13^19 x 17^19 x 19^27. Because every number has a unique prime factorisation, the resulting number can only ever represent this string. Axioms of the system can be expressed in this way, and the same method works for encoding whole proofs. Gödel's own numbering system was similar, but more complex, and composed such that he could ensure numbers had certain desirable properties.

Arithmetising the rules of inference and syntax for the system cannot be done in the same way since they are about symbols and strings, but are not strings or symbols themselves. Instead, Gödel identified correspondence between the rules of the system and the relationships between the Gödel numbers, encoding them as statements about numbers.

An axiom of PA is that if t is a term, then S(t) is a term (the successor of t). 0 is a term, S0 is a term (1), along with SSS0 (3). These strings are called successor expressions. If we reformulate the original rule in this language, then we can say that if t is a successor expression, then S(t) is also a successor expression.

Using Gödel numbering, the successor expressions can be represented by the following successor numbers:

0 = 2^{27}

S0 = $2^{19} \times 3^{27}$

SS0 = $2^{19} \times 3^{19} \times 5^{27}$

A pattern is emerging, and we can now utilise this to define the relationship between the successor numbers. We can say that if the Gödel number of an expression is a successor number, then if we change the last power to 19, and multiply by the next prime, raised to the power 27, then the Gödel number we have formulated is also a successor number. This exemplifies how Gödel's arithmetisation of rules of inference and syntax works. In the same way, he was able to represent the whole structure of P within arithmetic itself. Importantly, he built a formula which was in fact an actual mathematical predicate for provability: Prov (x,y), where y is the Gödel number of some formula in P, and x is the Gödel number of a sequence of formulas, which we call a proof candidate.

Prov(x, y) asserts a relation between the numbers x and y that is true if and only if the sequence of statements that x codes is a valid proof of the formula that y codes.

Using his method, Gödel constructed a 'Gödel sentence' (G). It essentially states that: "This statement cannot be proven within this system." It's not a paradox, but it is self-referential. If the system is consistent, then G cannot be proven within it. As observers from outside the system, we can see that G is true, the sentence indeed says that it cannot be proven. From this astonishing proof, we conclude that there are true statements in arithmetic that the system simply cannot prove within its own framework.

It is important to note the parameters of Gödel's theorems. Gödel's first incompleteness theorem applies solely to formal systems that are:

- Consistent (there are no contradictions that can be derived from its axioms)
- Formally axiomatised (the rules of the system can be mechanically listed)
- Able to express basic arithmetic

In such a system, Gödel showed that there are true statements about arithmetic that cannot be proven within the system itself.

The Second Incompleteness Theorem follows from the first theorem to explain why no system to which the first theorem applies can prove its own consistency. To prove it, we start with what the first theorem gives us: “this statement is not provable within this system”. If the system is consistent, this is unprovable in the system, therefore it’s true, but if we were able to formally prove this, we would also prove the sentence that we have determined to be unprovable within the system, showing that the system is contradictory and inconsistent. Therefore, if the system is consistent, then it cannot prove its own consistency.

This seems big, but maybe not to the extent that it might initially appear. Gödel’s theorems don’t mean mathematics is broken or unreliable. They just draw a precise line around what formal systems can do. Mathematics itself is of course still extraordinarily powerful, the theorems simply show that no single formal system can capture every mathematical truth. Staggering claims such as “science can never fully explain reality” or “objective truth doesn’t exist” don’t truly follow from Gödel’s findings. His results apply specifically to formal systems with the stated characteristics. You can’t stretch them to cover all knowledge, philosophy, or everyday reasoning. This is often the initial fault of popular interpretations.

In 1961, philosopher John Lucas took Gödel’s work in a bold direction. In ‘Minds, Machines and Gödel’, he argued that the human mind can never be equated to a machine. His reasoning: for any formal system, and indeed any machine that is built upon one, there’s a Gödel sentence it cannot prove, but which mathematicians can recognise as true. So, Lucas concluded that human reasoning mustn’t be purely mechanical.

It’s undeniably a fascinating argument, but ultimately, there are some flaws. For a start, it assumes we can reliably recognise the truth of Gödel sentences, which isn’t necessarily correct. Our own reasoning can certainly be inconsistent, and we are unable to prove our own consistency. Perhaps a deeper issue is what we mean when we say a Gödel sentence is “true.” If we’re appealing to some intuitive sense of truth that sits outside of any formal system, then maybe we haven’t escaped Gödel’s reach at all, we’ve just moved beyond the boundaries of one single system.

Alan Turing had a sharp and poigniant response to all of this. He pointed out in his 1950 paper Mind: “our superiority can only be felt on such an occasion in relation to the one machine over which we have scored our petty triumph. There would be no question of triumphing simultaneously over all machines.” He is right to remark that it is always possible to construct a new system based on a previous system, which can prove what the first machine could not. We should also point out that machines can prove things that we cannot, since the proofs are too long and complex for human minds.

Lucas objects, arguing that Turing’s reply shifts attention from the issue: the question isn’t about whether a machine can outperform a mind in some contest of problem solving, but whether a mind and a machine are the same kind of thing. His argument isn’t about superiority but rather non-identity.

“True,” he says, “the machine can do many things that a human mind cannot do: but if there is of necessity something that the machine cannot do, though the mind can, then, however trivial the matter is, we cannot equate the two, and cannot hope ever to have a mechanical model that will adequately represent the mind. Nor does it signify that it's only an individual machine we have triumphed over: for the triumph is not over only an individual machine, but over any individual machine that anybody cares to specify."

What about modern AI? It's a fair question. This line of thought is intriguing. In a world where many people fear the rise of AI and the potential dangers it may pose to humanity, could Lucas' argument provide some hope to us, that perhaps these machines may never be able to surpass humankind because they will never be able to truly replicate the human mind? Whilst AI systems are grounded in formal procedures, they're not easily pinned down as a single fixed formal system. They can learn and adapt to operate on higher-level systems with other properties. That said, any well-defined formal arithmetical component within them would still run into the limits defined by Gödel. There's another wrinkle: many AI systems are not strictly consistent, they often contradict themselves, and you can get them to admit it. In classical logic, an inconsistent system can, in principle, prove any statement. That makes applying Gödel's theorems to real-world AI more complicated than it might first appear.

So, is it right to model the human mind as a formal system? This is inconclusive. Human reasoning is often inconsistent and context-dependent, resisting straightforward formalisation. Applying incompleteness directly to cognition may be no more than a metaphor.

What did Gödel think? Gödel was a mathematical Platonist, he believed that when we think or talk about mathematics, the things we are talking about exist independently of us and the things that we do. He saw his theorems as evidence of this ideology. Concerning Lucas' argument, he suggested two possibilities:

Either a formal system cannot fully capture the human mind, or the human mind is itself limited and not fully reliable. He argued that humans have an intellectual intuition that allows us to grasp abstract truths. Overall, his position was cautious, and he left the issue open.

Gödel's theorems don't show that mathematics is limited, they highlight its open-ended nature. Every attempt to capture all mathematical truth within a fixed system will ultimately leave something out. But Gödel's message is to not see this as a weakness, but to see that that's exactly what makes mathematics alive, it will always be an inexhaustible discipline that keeps growing beyond any framework we try to contain it in.