compiled on 18 November 2025

Diagonalisation - The Road to Infinities, Truth and Gödel

Mahin Hossain and Rasmus Bakken

18 November 2025

Contents

Lecture I: Infinities

Summary: We introduce number, one-to-one correspondence, and cardinality. We show how infinite sets like the naturals, evens, and rationals can be the same size. We present Cantor's diagonal argument and Russell's paradox.

No one shall expel us from the paradise that Cantor has created for us.

– David Hilbert

It is not obvious how to explain the concept of number to someone who doesn't already know what numbers are.

You know what the *names* of the numbers are. You also know that there can be many names for the same number: the Roman 'XXV' is another name for the Arabic '25'.[1] We claim that imparting these names is not the same as imparting the concept of number. If we are asked "What is a planet?" and we reply with a list of names for planets, we have not been of much help unless the named entities were already clearly understood. Maybe we are allowed to assume that our questioner has a good grasp of 'Earth', 'Mars', 'Venus', etc. — and we can answer "What is a planet?" by saying "Earth and Mars and Venus and ... are planets". But if our questioner has no such grasp of the named entities, we must provide a more fundamental explanation than this list of names.

Is any such fundamental explanation possible for the concept of number?

Yes. A simple idea does much of the work: *matching.* Imagine laying out two collections in parallel and pairing each item from the first collection with exactly one item from the second, leaving nothing unmatched on either side. If this can be done, we say the two collections are the *same size.* This way of thinking needs no numerals at all. If we have a given collection of sheep and a given collection of trees, and we can tie exactly one sheep to one tree with no leftover trees or sheep, then there are as many sheep as there are trees.

I.1: Number via matching

Same size (or "one-to-one correspondence"): Two collections A and B are of equivalent size if their members can be paired up perfectly: each member of A goes with

1 Not all names for the numbers are equally useable: try calculating XXV × XXXII by hand without first converting the Roman numerals into Arabic. In this lecture course you will learn some unusual new names for the numbers; these names will have their uses.

one and only one member of B, and each member of B goes with one and only one member of A. No leftovers.

Here is how we can check that this behaves like an honest notion of equivalence of size. Equivalence is famously characterised by the three properties (reflexivity, symmetry, transitivity) which are all met:

- **Reflexivity**: Any collection can be matched with itself (pair each item to itself). Obvious, but reassuring.
- **Symmetry**: If A matches B, then B matches A (just reverse the pairings).
- **Transitivity**: If A matches B and B matches C, then A matches C (follow the lines in two hops).

We now introduce some technical terms for what we just discussed. What we have referred to as a perfect pairing is what mathematicians call a *bijection.* When a perfect pairing exists between two collections, mathematicians say that they are *equinumerous.*

> DEFINITION I.1 (BIJECTION)
> A bijection between two collections A and B is simply a perfect pairing between their members: a way to match each item of A with exactly one item of B, and each item of B with exactly one item of A. No item is left unmatched, and no item is used twice. When such a pairing exists, we say that A and B are *in bijection* (alternatively: *in one-to-one correspondence*).

> DEFINITION I.2 (EQUINUMEROSITY)
> Two collections A and B are equinumerous if there is a perfect pairing between them: each item of A is matched with exactly one item of B, and each item of B is matched with exactly one item of A. No leftovers and no repeats. (Notation: $A \sim B$)

With these two concepts defined, we are closer to exhibiting the concept of number without any vicious circularity. The trick is to deploy this notion of *size* that is revealed by our practice of matching. Clearly, not all collections are of the same size. So there are different sizes. We might then explain—to some entity that has no prior conception of numbers—that numbers are simply the different sizes that there can be. The technical term for size is *cardinality.*

When a collection can be matched with a finite list like

$$\{1\},\ \{1,2\},\ \{1,2,3\},\quad \text{and so on,}$$

we say it has a cardinality of 1, or 2, or 3, and so on.

In this usage, the numeral “3” is just a widely agreed label for the cardinality that all collections equinumerous with the set $\{1,2,3\}$. Let’s call “3” a cardinal number. The label is a convenience and we do not depend on it to explain *three-ness*; likewise for every other possible cardinal number. They emerge, ultimately, from the idea of matching.

I.2: First encounters with infinity

A collection is *infinite* if it cannot be matched with any finished list $\{1,2,\dots,n\}$. One striking sign of being infinite is this: an infinite collection can sometimes be matched with a proper *part* of itself. That would never happen for a finite collection, but for infinite ones it can—and will.

Let $\mathbb{N} = \{0,1,2,3,\dots\}$ be the natural numbers. This is an infinite collection. We will meet three surprises.

I.2.1: The even naturals are the same size as the naturals

Let $E = \{0,2,4,6,\dots\}$ be the even numbers. The even numbers, of course, are wholly contained within the collection of natural numbers. Now here is a perfect pairing:

$$\begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & \cdots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ 0 & 2 & 4 & 6 & 8 & \cdots \end{array}$$

Each natural number is paired with its double. No even number is left out, and no two naturals share the same even partner. So $\mathbb{N}$ and E are the same size. This already shows infinite size behaves unlike finite size: a whole can be the same size as a proper part. **The even naturals have the same cardinality as the naturals.**

I.2.2: The integers are the same size as the naturals

What if we compared the natural numbers to the collection of integers (i.e. positive *and* negative numbers)? The integers might seem to be 'double' the size of the naturals. Let $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$. And yet it turns out that we can pair $\mathbb{N}$ with $\mathbb{Z}$ by "zig–zagging" out from 0:

$$\begin{array}{cccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & \cdots \\ \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \updownarrow & \\ 0 & 1 & -1 & 2 & -2 & 3 & -3 & \cdots \end{array}$$

Every integer appears exactly once, so again the sizes match. **The integers have the same cardinality as the naturals.**

I.2.3: The rationals are the same size as the naturals

A rational number is a fraction p/q where p is an integer and q is a positive whole number, written in lowest terms (so 1/2 rather than 2/4). To see that the rationals can be listed in a single line like the naturals, imagine a grid:

$$\begin{array}{cccc} \frac{0}{1} & \frac{1}{1} & \frac{2}{1} & \cdots \\ \frac{0}{2} & \frac{1}{2} & \frac{2}{2} & \cdots \\ \frac{0}{3} & \frac{1}{3} & \frac{2}{3} & \cdots \\ \vdots & \vdots & \vdots & \end{array}$$

Now sweep through this grid along diagonals (first the short diagonal, then the next, and so on), and *skip repeats* like 2/2 (which is the same as 1/1). In this way you eventually meet every rational number, exactly once. So the rationals are the same size as the naturals, despite feeling "denser".

In each of the three demonstrations above, we took an infinite collection and showed that it actually has the same cardinality as the natural numbers by arranging every element of the set in a single infinite line that mimics the natural number line. We now introduce the technical term for such collections:

> Definition I.3 (Countable infinity)
> Any collection that can be listed in a single infinite line—first, second, third, and

so on—is called *countably infinite.* We have just seen that $\mathbb{N}$, the even numbers in $\mathbb{N}$, $\mathbb{Z}$ (the integers), and $\mathbb{Q}$ (the rationals) are all countably infinite.

I.3: Cantor's diagonal argument: there are bigger infinities

So far, various infinite sets have turned out to be countably infinite—that is, to have the same cardinality as the natural numbers. We might wonder at this point: do *all* infinites have this cardinality?

They do not. Cantor's great discovery was that there are *different sizes* of infinity. In particular, the real numbers—the points on a line—form a larger infinity than the natural numbers.

Decimal expansions. Every real number between 0 and 1 has a decimal form

$$x = 0.d_1 d_2 d_3 \dots, \qquad d_k \in \{0, 1, \dots, 9\}.$$

A few numbers have two forms (e.g. $0.5000\dots = 0.4999\dots$). To avoid ambiguity, we agree to *never* use the version that ends with endless 9s.

Theorem I.4 (Cantor's Theorem)
The real numbers between 0 and 1 are not listable as $x_1, x_2, x_3, \dots$; i.e. they are uncountable.

Proof Imagine, for contradiction, that we *have* listed all reals in $[0,1]$:

$$x_1,\ x_2,\ x_3,\ \dots$$

Write their decimals as $x_i = 0.a_{i1}a_{i2}a_{i3}\dots$ The underlined entries mark the *diagonal* positions $a_{11}, a_{22}, \dots, a_{66}$.

	1st digit	2nd digit	3rd digit	4th digit	5th digit	6th digit	...
x_1	$\underline{a_{11}}$	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	$\cdots$
x_2	a_{21}	$\underline{a_{22}}$	a_{23}	a_{24}	a_{25}	a_{26}	$\cdots$
x_3	a_{31}	a_{32}	$\underline{a_{33}}$	a_{34}	a_{35}	a_{36}	$\cdots$
x_4	a_{41}	a_{42}	a_{43}	$\underline{a_{44}}$	a_{45}	a_{46}	$\cdots$
x_5	a_{51}	a_{52}	a_{53}	a_{54}	$\underline{a_{55}}$	a_{56}	$\cdots$
x_6	a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	$\underline{a_{66}}$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	

Building the "anti-diagonal" number. Define a new decimal

$$y = 0.b_1 b_2 b_3 b_4 b_5 b_6 \dots$$

by *changing each underlined diagonal digit*:

$$b_n = \begin{cases} 5 & \text{if } a_{nn} \neq 5, \\ 4 & \text{if } a_{nn} = 5. \end{cases}$$

Thus $b_n \neq a_{nn}$ for every n. Visually: b_1 disagrees with the $\underline{a_{11}}$, b_2 disagrees with the $\underline{a_{22}}$, $\dots$, b_6 disagrees with the $\underline{a_{66}}$, and so on down the diagonal.

Because y differs from x_n in its n-th digit, we have $y \neq x_n$ for *every* n. Yet y is still a real number in $[0, 1]$. So the list was not complete—a contradiction.

Takeaway. No list can capture all reals. The diagonal construction always finds a missing one, so

$$|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}| \quad \text{but} \quad |\mathbb{R}| \text{ is strictly larger.}$$

I.4: Russell's paradox and why we need rules for forming sets

A tempting but dangerous idea is: "For any clear condition, there is a set of all things that satisfy it." Russell showed that this sweeping principle cannot be right.

In the late 19th century, many mathematicians and philosophers hoped to rebuild all of arithmetic on purely logical foundations (the "logicist" project). Cantor had just opened the door to talking rigorously about infinite collections, and people freely used an informal comprehension principle: for any sensible condition, there is a set of all things satisfying it. Warning signs appeared (for example, Burali–Forti's 1897 observation that the "set of all ordinals" leads to trouble), but the principle still guided much routine reasoning.

Russell noticed in 1901 that the comprehension principle, taken at face value, collapses on itself.

> **Russell's paradox.** Consider the collection R of all collections that are *not* members of themselves. Is R a member of itself?

If you answer "yes", then by definition R should *not* be a member of itself. If you answer "no", then by definition R *is* a member of itself. Either way, contradiction.

A more homely version is the "barber" story: in a certain town, the barber shaves all and only those who do not shave themselves. Does the barber shave himself? The story ties itself in a knot for the very same reason.

In 1902 Russell wrote to Frege about the problem, just as Frege was completing the second volume of his *Grundgesetze*; Frege added an appendix acknowledging that the paradox undermined his system. Russell's own response was to propose restrictions on formation of collections (type theory) and, later with Whitehead, to develop these ideas in *Principia Mathematica*.

The lesson of Russell's Paradox is that we must be careful about what we allow as a "set." Modern set theories (such as the Zermelo–Fraenkel system) keep the fruitful parts of set talk while placing modest restrictions on how new sets may be formed. Within such systems, all the results above about sizes of infinity go through cleanly, and the paradoxes are blocked.

I.5: Takeaways

- The basic idea of number can be grounded in everyday *matching*: same number means a perfect pairing is possible.
- Infinite collections can match some of their own proper parts; this never happens with finite ones.
- Many familiar infinities (natural numbers, whole integers, rational numbers) are the same "listable" size: countably infinite.
- Cantor's diagonal method shows that the real numbers form a strictly larger infinity: they cannot be captured by any list.
- Russell's paradox warns that not every condition defines a legitimate set; some discipline is required.

Lecture II: Numbers are Alive

Summary: We motivate Gödel-numbering as a precise way to achieve self-reference. We outline a Gödel-encoding scheme for symbols, sentences, and proofs, and illustrate it through worked examples.

There is a difference between a thing and talking about a thing.

– Kurt Gödel

Self-reference is at the heart of this lecture series, and in this lecture we are going to have a look at how Gödel managed to get numbers to talk about themselves. We are going to do this in a formal way, which means we have two tasks in front of us.

Firstly, we need to define a formal language of arithmetic. That is, a formal language that can express statements such as 2+2=4 and "there are infinitely many primes". We would like our formal language to be able to express, in theory, as much as possible of what can meaningfully be said about the natural numbers. We will do this in section II.1.

Secondly, we are going to show how we can get our formal language of arithmetic to talk about itself. We have to achieve such a feat under the restriction that a formal language of arithmetic can only talk about numbers — so if we want the language to talk about itself, we need to find a way of coding sentences of our language into numbers. The idea of coding up sentences of a formal language into numbers so that the formal language can talk about itself is one of the two genius tricks that Gödel's Incompleteness Theorems are based on. Giving such an encoding is usually called giving a Gödel encoding, as Gödel was the first one to do this when he proved his famous theorems. This will be done, with ample examples, in section II.2.

II.1: The Formal Language

In this section, we will introduce the formal language we will study. It is meant to capture most of what we can informally say about arithmetic. We will call this language $\mathcal{L}$, and in $\mathcal{L}$ we want to be able to express sentences such as 2+2=4 and "there are infinitely many primes". The former sentence will be a lot easier to express than the latter sentence.

We start by writing down which symbols we are allowed to use. The symbols of $\mathcal{L}$ are $0, S, +, *, exp, (,), \neg, \rightarrow, \forall, =, \#$ and v_i for any natural number i. This means that v_0, v_1 and v_{419} are all valid symbols of $\mathcal{L}$. An *expression* is any finite string of symbols. So the following are expressions of $\mathcal{L}$.

(i) $S(0) + S(0) = S(S(0))$.
(ii) $\forall\neg v_8)$.
(iii) $v_2 v_3 + + =$.

So far, we haven't done anything interesting. (i) might seem like something intelligible, but (ii) and (iii) are pure rubbish. We need a way of distinguishing well-formed formulae from rubbish. This is something we have in English, there are plenty of rules telling us which symbols we are allowed to put after each other, and we need the same here as well.

Very loosely speaking, the sentences of our language are supposed to tell us something about numbers. Hence, we need one way of picking out numbers and one way of relating them to each other in different ways. The first of these we will call *terms*. Roughly, a term is something that denotes a number. 0 is supposed to pick out the number zero, so that needs to be a term. Furthermore, the variables are supposed to stand for numbers, so any variable v_i, where i is a natural number, is a term. S is supposed to be the successor function, which gives us the successor of a natural number. That is, if n is a natural number, then $n + 1$ is the successor of n. 6 is therefore the successor of 5, which is the successor of 4 and so on. Hence, we say that if x is a term, then $S(x)$ is also a term. $+, *$ and exp are supposed to be addition, multiplication and exponentiation respectively. Hence, we say that if x and y are terms, then so are $x + y$, $x * y$ and $exp(x, y)$. Lastly, we conclude that nothing else is a term. Hence, the terms will be constructed *recursively* starting from 0 or a variable v_i and applying some of the rules just introduced.

This way of defining a concept is called a recursive definition, and you'll see it several times in this lecture series. A recursive definition first tells you that some basic things belong to the definition, and then shows you how to construct more complex things that belong to the definition out of the things already in the definition. If this sounds a bit mysterious, don't worry. You'll see more examples soon!

Before we move on, let us construct some terms. One of the most basic term we can construct is 0. From here, we have four different options, each relating to one of our four symbols $S, +, *, exp$. Hence, from 0 we can create the following four terms: $S(0)$, $0 + 0$, $0 * 0$ and $exp(0, 0)$. Having five different terms at our disposal, we can start creating even more terms. We can also have start with the term v_5, and then construct the terms $S(v_5)$, $v_5 + v_5$, $v_5 * v_5$ and $exp(v_5, v_5)$. From here, we can start combining the terms into even more complicated terms such as $S(0) + S(v_5)$ and $exp(0 + 0, v_5 * v_5)$ and so on for as long as we want.

Now, we need to point out a subtle thing. $0 + 0$ and $0 * 0$ are really two distinct terms. It might be tempting to say that they both are 0, but this is not true. $0 + 0$ and $0 * 0$ both *evaluate* to 0, but as syntactic objects they are both distinct from each other and from 0.

It is just like how the capital of Norway evaluates to Oslo, but the string of symbols "the capital of Norway" is not the same as the string of symbols "Oslo".

There is one class of terms we will be particularly interested in, and that is the *numerals*. A numeral is any term which is reach by starting with 0 and repeatedly applying S. Hence, 0 is a numeral, $S(0)$ is a numeral and $S(S(S(S(S(S(0))))))$ is a numeral. When talking about numbers in $\mathcal{L}$, we will for the most part express them with numerals. Hence, when referring to the number 2, we will use $S(S(0))$. It can be tiresome to write (and read) many occurrences of S one after the other, and so we will introduce some shorthand notation. For any number n, we write $\overline{n}$ for the numeral that has n occurrences of S. Hence, $\overline{0}$ is 0, $\overline{1}$ is $S(0)$ and $\overline{3}$ is $S(S(S(0)))$.

So far, we have seen that terms allow us to refer to numbers. Now, we need a way of talking about the numbers. This is where the concept of a formula comes in. A formula is a way of expressing something about numbers. The simplest expressions we can make are saying that two numbers are equal. Hence, if σ and τ are terms, then $\sigma = \tau$ is a formula. These very simple formulae get a special name, we call them *atomic* formulae. The intuition is that these are the atomic building blocks that our more complex formulae will be built up with.

Continuing on with the recursive definition of a formula, we say that if φ and ψ are formulae, and i is a natural number, then $\neg\varphi$, $(\varphi \to \psi)$, $\forall v_i \varphi$ are formulae. You might remember formulae from other logic courses such as $\varphi \wedge \psi$, $\varphi \vee \psi$ and $\exists v_i \varphi$. These are not officially formulae in $\mathcal{L}$, however there are formulae in $\mathcal{L}$ that expresses them. For instance, $(\varphi \wedge \psi)$ can be taken as shorthand for $\neg(\varphi \to \neg\psi)$. Similar shorthands are available for the other two[2], and so we will frequently "cheat" and use the symbols $\wedge, \vee$ and $\exists$ although they are not officially a part of $\mathcal{L}$. In practice, we also cheat slightly with brackets. We might add them or remove them, depending on what we think make it easier to read and understand the expression. Lastly, $\sigma \neq \tau$ is shorthand for $\neg\sigma = \tau$.

Hence, we can now recursively construct formulae. Some examples are:

(i) $S(v_5) \neq 0$.
(ii) $S(v_2) = S(v_3) \to v_2 = v_3$.
(iii) $(\overline{2} + \overline{2} = \overline{4})$.
(iv) $\forall v_1(\forall v_2 \forall v_3((v_1 = v_2 * v_3) \to (v_3 = \overline{1} \vee v_3 = v_1)) \to (\exists v_4 \forall v_5 \forall v_6((v_4 = v_5 * v_6) \to (v_6 = \overline{1} \vee v_6 = v_4)) \wedge \exists v_7(v_7 \neq 0 \wedge v_1 + v_7 = v_4)))$.

The last two formulae say, respectively, $2 + 2 = 4$ and "there are infinitely many primes". There is also another difference between (i) and (ii), on the one side, and (iii) and (iv)

2 We leave the task of finding these shorthands to you, our dear reader.

on the other side. The latter pair has no variables that are not *bounded* by a quantifier. In the former pair, all the variables occur *free* in the formulae. If a formula has no free variables, then we call it a sentence.

II.2: The Gödel Encoding

In this section, we will show how $\mathcal{L}$ can talk about itself. The idea is to give a *Gödel encoding* of $\mathcal{L}$. That is, a way of assigning numbers to formulae of $\mathcal{L}$ that is reasonably efficient and clever. We will keep the terms reasonably efficient and clever purposefully vague. However, we should say that there are several ways of writing down a Gödel encoding, and, as opposed to what Orwel will have you believe, not all Gödel encodings are equal. We will use a Gödel encoding that is largely inspired by Peter Smith[Smith 2020]. The idea is relatively simple. We are going to assign each symbol in $\mathcal{L}$ a number, or a code if you like, and then devise a way of assigning an expression of $\mathcal{L}$ a number based on the numbers we have assigned to the symbols of that expression.

Let us start with the first task. As we remember, the symbols of $\mathcal{L}$ are $0, S, +, *, exp, (,), \neg, \rightarrow, \forall, =, \#$ and v_i for any natural number i. If we ignore the variables, there are only finitely many symbols and so no difficulty of assigning them a unique number. We start by assigning the number 1 to the symbol 0, the number 3 to the symbol S, the number 5 to the symbol $+$ and so on. Putting it all together, we get the following table.

0	S	$+$	$*$	exp	$($	$)$	$\neg$	$\rightarrow$	$\forall$	$=$	$\#$
1	3	5	7	9	11	13	15	17	19	21	23

We recall that our variables are of the form v_i for a natural number i. We therefore say that v_i get the Gödel code $2(i+1)$. Hence, v_0 get the code 2, v_1 get the code 4, v_3 get the code 6 and so on. To write this symbolically, we use the $\ulcorner$ and $\urcorner$ around a symbol to denote its Gödel code. Hence, $\ulcorner S \urcorner = 3$ and $\ulcorner v_5 \urcorner = 12$.

Our next task is to find a way to expand our Gödel encoding from symbols to expressions. Consider the expression $0 = 0$. We have assigned each symbol in that expression a Gödel code, so it is natural to think of the code of the whole expression as a tuple of the codes of its symbols. That is, to think of $\ulcorner 0 = 0 \urcorner$ as $\langle \ulcorner 0 \urcorner, \ulcorner = \urcorner, \ulcorner 0 \urcorner \rangle = \langle 1, 21, 1 \rangle$. If we had an elegant way of encoding a tuple of numbers into one number, then we could let $\ulcorner 0 = 0 \urcorner$ be the code of the tuple $\langle 1, 21, 1 \rangle$. The next task, therefore, is to find a way of encoding tuples of numbers into one number.

Before we can do this, we need to do a bit of number theory. Recall that a *prime number* n is a natural number larger than 1 such that it is only divisible by 1 and itself. That is, if $m * k = n$ for natural numbers m and k, then we know that either $m = 1$ and $k = n$ or $m = n$ and $k = 1$. 3 and 5 are prime numbers, but 4 is not a prime number because $2 * 2 = 4$. A *composite* number is a natural number larger than 1 that is not a prime number. The intuition is that a composite number is built up of smaller prime numbers. For instance, 4 is a composite number that is built up of 2 and 2. Similarly, 18 is a composite number as $18 = 9 * 2$. However, 9 is also a composite number as $9 = 3 * 3$, so we might say that 18 is built up of 3, 3 and 2. Since there is a repetition of 3, we might instead say that 18 is built up of 3^2 and 2. This way of looking at a composite number is called a *prime composition.* This prime composition of 18 is unique, which means that there is no other way of writing 18 down as a product of primes than $3^2 * 2$. This is no accident. In fact there is a theorem of number theory, called the *Fundamental Theorem of Arithmetic*, which states that this is the case for every number. Let us write it down explicitly, before we unpack it.

> Theorem II.1 (The Fundamental Theorem of Arithmetic)
> Every natural number greater than 1 is either a prime, or has a unique prime composition.

The theorem says that for every natural number $n > 1$, there exists a unique set of primes $p_1, \ldots, p_k$ and a unique set of non-zero exponents $e_1, \ldots, e_k$ such that $n = p_1^{e_1} * p_2^{e_2} * \ldots * p_{k-1}^{e_{k-1}} * p_k^{e_k}$. Hence, for any natural number $n > 1$, we can associate it with its unique ordered tuple of exponents $\langle e_1, \ldots, e_k \rangle$.

We will now use the Fundamental Theorem of Arithmetic to show how we can encode a tuple of numbers $\langle n_1, n_2, n_3, \ldots, n_k \rangle$ into a number m. Before we do this, however, let us consider a method that would not work. The failure of the following method will partly explain why we do our encoding the way we do it. Suppose we suggest that a tuple of numbers $\langle n_1, \ldots, n_k \rangle$ should be encoded as $n_1 * n_2 * n_3 * \ldots * n_k$. That would mean that $\langle 2, 3, 3 \rangle$ would be encoded as $2 * 3 * 3 = 18$. Now, if you are given the code of 18, can you find out which tuple it is encoding? Absolutely not. Firstly, $18 = 2 * 9 = 2 * 3 * 3$, so you would not know if 18 was encoding a tuple with two elements or a tuple with three elements. Secondly, suppose somehow that you knew that 18 was encoding a tuple of three elements, you could not know if it was the tuple $\langle 2, 3, 3 \rangle$, $\langle 3, 2, 3 \rangle$ or $\langle 3, 3, 2 \rangle$ because $2 * 3 * 3 = 3 * 2 * 3 = 3 * 3 * 2 = 18$. Remember that the order of the tuple matters a lot, especially when we know that each number in the tuple is the code for a symbol. If we swap the position of two numbers in the tuple, then we swap the position of two symbols in our expressions. Hence, we need a way of encoding a tuple of numbers such that there is only that one tuple that will be encoded to the given number. In more technical jargon, we want our encoding to be *injective*.

We will first show how we will do this generally, then consider some examples. Let π_i denote the i-th prime number. That is $\pi_1 = 2$, $\pi_2 = 3$, $\pi_3 = 5$, $\pi_4 = 7$ and so on. If we have the tuple $\langle n_1, n_2, n_3, \ldots, n_k \rangle$ of non-zero natural numbers, then we let $\pi_1^{n_1} * \pi_2^{n_2} * \pi_3^{n_3} * \ldots * \pi_k^{n_k}$ be the code of $\langle n_1, n_2, n_3, \ldots, n_k \rangle$. By the Fundamental Theorem of Arithmetic, every number has exactly one prime composition. This means that if $m = \pi_1^{n_1} * \pi_2^{n_2} * \pi_3^{n_3} * \ldots * \pi_k^{n_k}$, then there is no other way of finding prime numbers $p_1, \ldots, p_l$ and non-zero exponents $e_1, \ldots, e_l$ such that $m = p_1^{e_1} * p_2^{e_2} * p_3^{e_3} \ldots p_l^{e_l}$. So, any number m encodes exactly one tuple in this way, because it has exactly one prime composition[3].

Let us return to the example from earlier, and calculate the Gödel code of the sentence $0 = 0$. We recall that $\ulcorner 0 \urcorner = 1$ and $\ulcorner = \urcorner = 21$, so it is only a matter of finding the encoding of the tuple $\langle 1, 21, 1 \rangle$ that remains. The first three prime numbers are $2, 3$ and 5. So we encode the tuple $\langle 1, 21, 1 \rangle$ by $2^1 * 3^{21} * 5^1 = 2 * 3^{21} * 5 = 104603532030$. By the Fundamental Theorem of Arithmetic, 104603532030 only has the prime composition of $2^1 * 3^{21} * 5^1$, and so from 104603532030 we can uniquely read off the tuple $\langle 1, 21, 1 \rangle$. Hence, we say that $\ulcorner 0 = 0 \urcorner = 104603532030$.

We will consider one more example before we move on to the last thing we will cover in this lecture, namely how to encode a proof. Let us consider $\ulcorner \forall v_5 (\neg S(v_5) = 0) \urcorner$. We start by writing down the associated tuple $\langle \ulcorner \forall \urcorner, \ulcorner v_5 \urcorner, \ulcorner (\urcorner, \ulcorner \neg \urcorner, \ulcorner S \urcorner, \ulcorner (\urcorner, \ulcorner v_5 \urcorner, \ulcorner) \urcorner, \ulcorner = \urcorner, \ulcorner 0 \urcorner, \ulcorner) \urcorner \rangle = \langle 19, 12, 11, 15, 3, 11, 12, 13, 21, 1, 13 \rangle$. There are 11 numbers in this tuple, so we need the first 11 primes to write out the code of the tuple. They are $2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31$. This means that $\ulcorner \forall v_5 (\neg S(v_5) = 0) \urcorner = 2^{19} * 3^{12} * 5^{11} * 7^{15} * 11^3 * 13^{11} * 17^{12} * 19^{13} * 23^{21} * 29^1 * 31^{13}$. Finishing the calculation is also left as an exercise to the reader, preferably by hand.

We now move on to the last part of the lecture, namely how to write down a Gödel encode for a proof. We have not yet formally defined a proof, which we will do in the next lecture, but there is only one thing we need to know about a proof in order to write down its Gödel code. A proof consists of a finite sequence of $\mathcal{L}$-sentences. Hence, if we have a proof given by $\varphi_1, \varphi_2, \ldots, \varphi_k$ then we write it down as the long expression $\#\varphi_1\#\varphi_2\#\varphi_3\# \ldots \#\varphi_{k-1}\#\varphi_k\#$. This is another expression in $\mathcal{L}$, and so we can assign it

3 There is a small caveat here, namely that none of the numbers in our tuple can be 0. To keep things simple, we will not expand more on why we have this restriction. We therefore do what mathematicians are always very fond of doing, and leave the question why none of the numbers can be 0 as an exercise to you, dear reader.

a Gödel code in our usual way. The reason this works is because the symbol # will never occur in a sentence, so it will never be in any of the $\varphi_1, \ldots, \varphi_k$. Sadly, here we will not give you any examples, as it would take way too long to write down.

References

Peter Smith. *An Introduction to Gödel's Theorems.* Logic Matters: https://www.logicmatters.net/resources/pdfs/godelbook/GodelBookLM.pdf, 2020.