*compiled on 1 December 2025*

# Diagonalisation - The Road to Infinities, Truth and Gödel

Mahin Hossain and Rasmus Bakken

1 December 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:

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

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 $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. 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

$$x = 0.d_1d_2d_3\ldots, \qquad d_k \in \{0, 1, \ldots, 9\}.$$

A few numbers have two forms (e.g. $0.5000\ldots = 0.4999\ldots$). 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 $x_1, x_2, x_3, \ldots$; i.e. they are *uncountable*.

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

$$x_1,\ x_2,\ x_3,\ \ldots$$

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

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

**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 $\underline{a_{22}}$, ..., $b_6$ 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 $S(S(S(S(S(0)))))$ 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 $\sigma$ and $\tau$ 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 $\varphi$ and $\psi$ 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 $\pi_i$ 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 $\mathcal{L}$-formulas. 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 formula, 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.

## Lecture III: Truth and the Liar

Summary: We introduce provability inside a formal system, prove Gödel's diagonal lemma, and use it to show the undefinability of truth in a formal logical setting.

> *All Cretans are liars.*
>
> – Epimenides, a Cretan

In this lecture, we will see our first instances of self-reference in formal mathematics. Using the Gödel-numbering scheme that we introduced in the last lecture, we will construct self-referential sentences. To remove all doubt of whether our instance of self-reference is 'genuine', we will define a formal system $PA$ using the language $\mathcal{L}$ constructed in the last lecture. Within the formal system $PA$ we will have sentences that, when examined metatheoretically, will say, e.g., "I am unprovable".

The trick to doing this all will be using Gödel's Diagonal Lemma, which essentially says that for any formula $\varphi(v_i)$, there is a sentence $\psi$ such that $PA$ proves $\psi \leftrightarrow \varphi(\ulcorner \psi \urcorner)$. The intuition is that any formula $\varphi$ with one free variable essentially has a *slot* that a number can fit into. When you place a number in the variable slot, you obtain a sentence. For example, the formula $\mathsf{Even(x)}$ has a slot for a number — you could slot in $\mathsf{Even}(\overline{32})$ to obtain a true sentence or $\mathsf{Even}(\overline{31})$ to obtain a false sentence. *Now, note that the Gödel code numbers of formulas/sentences/sets of sentences are numbers too.* Therefore, these code numbers can also be slotted into some formula with a free variable.

By setting up a very cunning one-place formula, we can make it such that when *that formula's* Gödel number is placed in the one open slot, we end up with a sentence which "talks about itself".

This lecture will contain three sections. Firstly, in section III.1, we will introduce the formal system $PA$ and show what it means for $PA$ to prove a formula $\varphi$. We will employ a Hilbert-style calculus, which is difficult to do proofs in, but makes it easy to reason *about* proofs. We demonstrate by ending that section with a proof in $PA$ of $1 + 1 = 2$, which takes 15 lines to do.

In section III.2, we will prove Gödel's Diagonal Lemma. This is the key lemma that allows us to construct self-referential sentences. To prove the Diagonal Lemma, we will introduce the notion of substitution, before helping ourselves to a function $sub(v_1, v_2, v_3)$ within $PA$ that allows us to perform substitution on Gödel codes of formulae. In this course there is no time to go through the full construction of this function, so we will have to be content with assuming that it works as needed.

Lastly, in section III.3 we show Tarski's famous Undefinability of Truth Theorem. It essentially say that there is no $\mathcal{L}$-formula $\varphi(v_i)$ that captures the intuitive notion of truth. More concretely, that means that there is no $\varphi$ such that $PA \vdash \varphi(\ulcorner\psi\urcorner) \leftrightarrow \psi$ for all $\mathcal{L}$-sentences $\psi$. The Diagonal Lemma and the liar sentence will play a key role in the proof.

## III.1: What is a Proof

In this section, we will have a look at what a proof in a formal system is. There are many different equivalent ways of defining a proof in a formal system, each with advantages and disadvantages. We will use a Hilbert-style calculus, which has the advantage that it is very easy to reason about but the disadvantage that it is very difficult to actually use. Luckily, we are not planning on using it very much, so the disadvantage will not hinder us terribly.

A Hilbert-style calculus is given by **a set of axioms** and **rules of inference**. Our axioms will come in two flavours: the logical axioms and the non-logical axioms. The logical axioms are the ones characterising the underlying logic we will be using, which in this case will be classical logic. An example of a logical axiom is $\varphi \to (\psi \to \varphi)$ for two formulae $\varphi$ and $\psi$. The non-logical axioms are the axioms pertain to our subject matter, which in this case is a theory of the natural numbers (i.e. arithmetic). Hence, a non-logical axiom would be something true of the natural numbers. For example, we have the axiom $\forall v_i(S(v_i) \neq 0)$, which essentially says that 0 is not the successor of any number.[4]

The rules of inference, meanwhile, allow us to proceed from premises to conclusions. In this case we proceed from our axioms to *theorems*, that is, claims that follow logically from the axioms. One famous rule of inference is modus ponens, which allow us to infer $\psi$ from the set of premises $\{\varphi \to \psi, \varphi\}$.

We will write down a fixed list of logical axioms and a list of rules of inference. (There are actually a few different viable lists of axioms and rules, and which specific list we take is of little importance.) The axioms are supposed to characterise first-order predicate logic with equality, which is the logic you end up with at the end of most introductory courses in logic.[5] This might be a slightly different presentation of something you most likely have seen before.

4 Negative numbers are a foreign concept to us at the moment.
5 See, for instance, Halbach's Logic Manual.

Definition III.1 (First-order Predicate Logic)
Let $\varphi$, $\psi$ and $\xi$ be $\mathcal{L}$ formula, then the following are axioms of first-order logic:

C1 $\varphi \to (\psi \to \varphi)$.
C2 $(\varphi \to (\psi \to \xi)) \to ((\varphi \to \psi) \to (\varphi \to \xi))$.
C3 $(\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$.
C4 $\forall v_i \varphi(v_i) \to \varphi(t)$, where $t$ is a term of $\mathcal{L}$ such that it does not contain a variable $v_j$ where $v_i$ occurs free in the scope of $\forall v_j$ in $\varphi$.
C5 $\forall v_i(\varphi \to \psi) \to (\varphi \to \forall v_i \psi)$, provided $v_i$ does not occur free in $\psi$.
C6 $\forall v_i(v_i = v_i)$.
C7 If $F$ and $G$ are atomic formulae, where $G$ results from replacing some but not necessarily all of $v_i$ in $F$ by $v_j$, then $\forall v_i \forall v_j(v_i = v_j \to (F \to G))$ is an axiom.

The rules of inference are the following:

MP From $\varphi$ and $\varphi \to \psi$ you may infer $\psi$.
Gen From $\varphi$ you may infer $\forall v_i \varphi$.

The important thing is that this set of axioms sufficiently characterises first-order predicate logic, and so any theorem of first-order predicate logic is provable from them. A different set of axioms might also suffice.

Let us for a moment forget about the non-logical axioms, and consider what a proof might look like in the Hilbert-style calculus of pure first-order predicate logic.

A proof of a formula $\varphi$ is **a finite sequence of formulae** $\varphi_1, \ldots, \varphi_k$ such that the last formula in the sequence $\varphi_k = \varphi$ and for each $i \leq k$, either $\varphi_i$ is an axiom or it follows from earlier members of the sequence by application of some rule of inference. The intuition is that we write down some axioms, and reason with our rules of inference, until we end up with the formula that we want to prove. In practise, it is common to write a proof down as a list. Say that we want to prove that $\varphi \to \varphi$. A proof might look something like this.

| | | |
|---|---|---|
| **1** | $(\varphi \to ((\varphi \to \varphi) \to \varphi)) \to ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi))$ | C2 |
| **2** | $\varphi \to ((\varphi \to \varphi) \to \varphi)$ | C1 |
| **3** | $((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi))$ | MP with line 1 and line 2 |
| **4** | $\varphi \to (\varphi \to \varphi)$ | C1 |
| **5** | $\varphi \to \varphi$ | MP with line 3 and 4 |

As mentioned earlier, using a Hilbert-style calculus to prove things is not a very pleasant endeavour. It is not very intuitive why we were using axioms C1 and C2 in the way we did. However, it is easy to understand what is going on conceptually, when looking at an already-written proof. We invoke a couple axioms, then use one of our rules of inference, invoke another axiom, use a rule of inference and we end up with the 5th line being $\varphi \to \varphi$ as required.

We will conclude this section by writing down the non-logical axioms that we need, and giving an example of how to use them in a proof. Our theory of the natural numbers is called Peano Arithmetic, or $PA$ for short. Again, when writing down a formal proof in $PA$ nothing changes except that you are allowed to use more axioms in your proof. We usually write $PA \vdash \varphi$ to express that $\varphi$ is provable using the axioms of $PA$ and the axioms of first-order logic.

We conclude this section by writing down the axioms of $PA$ and showing that $PA \vdash \overline{1} + \overline{1} = \overline{2}$. Recall that $\overline{1}$ is shorthand for $S(0)$ and $\overline{2}$ is shorthand for $S(S(0))$.

> Definition III.2 (Peano Arithmetic)
> Peano Arithmetic, or $PA$ for short, is the theory given by the following set of axioms:
>
> PA1 $\forall v_i(S(v_i) \neq 0)$.
> PA2 $\forall v_i \forall v_j(S(v_i) = S(v_j) \to v_i = v_j)$.
> PA3 $\forall v_i(v_i + 0 = v_i)$.
> PA4 $\forall v_i \forall v_j(v_i + S(v_j) = S(v_i + v_j))$.
> PA5 $\forall v_i(v_i * 0 = 0)$.
> PA6 $\forall v_i \forall v_j(v_i * S(v_j) = v_i * v_j + v_i)$.
> PA7 $\forall v_i(exp(v_i, 0) = \overline{1})$.
> PA8 $\forall v_i \forall v_j(exp(v_i, S(v_j)) = exp(v_i, v_j) * v_i)$.
> $Ind_{\mathcal{L}}$ For any $\mathcal{L}$ formula with exactly one free variable $\varphi(v_i)$, $(\varphi(0) \wedge \forall v_i(\varphi(v_i) \to \varphi(S(v_i)))) \to \forall v_i \varphi(v_i)$ is an axiom.

We can now prove that $1 + 1 = 2$. This took Russell and Whitehead quite a few pages to do in their Principia Mathematica.

> Lemma III.3
> $PA \vdash \overline{1} + \overline{1} = \overline{2}$.

| | | |
|---|---|---|
| **1** | $\forall v_i \forall v_j (v_i + S(v_j) = S(v_i + v_j))$ | PA4 |
| **2** | $\forall v_i \forall v_j (v_i + S(v_j) = S(v_i + v_j)) \rightarrow \forall v_j (S(0) + S(v_j) = S(S(0) + v_j))$ | C4 |
| **3** | $\forall v_j (S(0) + S(v_j) = S(S(0) + v_j))$ | MP with line 1 and line 2 |
| **4** | $\forall v_j (S(0) + S(v_j) = S(S(0) + v_j)) \rightarrow S(0) + S(0) = S(S(0) + 0)$ | C4 |
| **5** | $S(0) + S(0) = S(S(0) + 0)$ | MP with line 3 and 4 |
| **6** | $\forall v_i (v_i + 0 = v_i)$ | PA3 |
| **7** | $\forall v_i (v_i + 0 = v_i) \rightarrow S(0) + 0 = S(0)$ | C4 |
| **8** | $S(0) + 0 = S(0)$ | MP with line 6 and line 7 |
| **9** | $\forall v_i \forall v_j (v_i = v_j \rightarrow (S(0) + S(0) = S(v_i) \rightarrow S(0) + S(0) = S(v_j))$ | C7 |
| **10** | $\forall v_i \forall v_j (v_i = v_j \rightarrow (S(0) + S(0) = S(v_i) \rightarrow S(0) + S(0) = S(v_j)) \rightarrow \forall v_j (S(0) + 0 = v_j \rightarrow (S(0) + S(0) = S(S(0) + 0) \rightarrow S(0) + S(0) = S(v_j))$ | C4 |
| **11** | $\forall v_j (S(0) + 0 = v_j \rightarrow (S(0) + S(0) = S(S(0) + 0) \rightarrow S(0) + S(0) = S(v_j))$ | MP line 9 and line 10 |
| **12** | $\forall v_j (S(0) + 0 = v_j \rightarrow (S(0) + S(0) = S(S(0) + 0) \rightarrow S(0) + S(0) = S(v_j)) \rightarrow S(0) + 0 = S(0) \rightarrow (S(0) + S(0) = S(S(0)) \rightarrow S(0) + S(0) = S(S(0))$ | C4 |
| **13** | $S(0) + 0 = S(0) \rightarrow (S(0) + S(0) = S(S(0) + 0) \rightarrow S(0) + S(0) = S(S(0))$ | MP line 11 and line 12 |
| **14** | $(S(0) + S(0) = S(S(0) + 0) \rightarrow S(0) + S(0) = S(S(0))$ | MP line 8 and line 13 |
| **15** | $S(0) + S(0) = S(S(0))$ | MP line 5 and line 14 |

We did warn you that doing a proof in a Hilbert-style calculus is no fun. But now you have a good explanation for *how it really is the case that* '$1 + 1 = 2$', if you ever get pestered by some critical theorist at a party.

## III.2: The Diagonal Lemma

In this section, we will prove the diagonal lemma. After Gödel-numbering, this is the second ingenious device that Gödel developed in order to prove his famous incompleteness theorems. Let us first state the lemma.

Theorem III.4 (Diagonal Lemma)
If $\varphi(v_n)$ is a $\mathcal{L}$-formula with one free variable then there is a $\mathcal{L}$-sentence $\gamma$ such that $PA \vdash \gamma \leftrightarrow \varphi(\overline{\ulcorner \gamma \urcorner})$.

The goal of this section is to prove the diagonal lemma. The key to proving the diagonal lemma will indeed be some sort of diagonalisation.

However, before we can diagonalise, we have to talk about substitution. Let $\varphi$ be the formula $S(v_5) = S(S(0))$. It has only one free variable, namely $v_5$. Now, to substitute $v_i$ in $\varphi$ by a term $t$ means to uniformly replace each instance of $v_i$ in $\varphi$ by $t$. Hence, if we substitute $v_5$ by $S(0)$ in $\varphi$ then we get $S(S(0)) = S(S(0))$. We denote the result of such substitution by the expression $\varphi(S(0)/v_5)$. So, $\varphi(0/v_5)$ is the (false) sentence $S(0) = S(S(0))$ whilst $\varphi(v_3/v_5)$ is the formula $S(v_3) = S(S(0))$. On the other hand, $\varphi(S(0)/v_4)$ is still $S(v_5) = S(S(0))$ as $v_4$ does not occur in $\varphi$.

There are always some complications when dealing with substitutions. Let $\varphi$ be the formula $\forall v_4(S(v_4) = v_3)$. Then we see that $\varphi(v_5/v_3)$ becomes $\forall v_4(S(v_4) = v_5)$ whilst $\varphi(v_4/v_3)$ becomes $\forall v_4(S(v_4) = v_4)$. $\varphi(v_5/v_3)$ and $\varphi(v_4/v_3)$ are very different, the former is a formula whilst the latter is a sentence. Furthermore, the former expresses something that could be true, whilst the latter expresses something that has to be false. Intuitively, this does not feel quite right. Substituting in different variables in the same formula should not change anything. What happened with $\varphi(v_4/v_3)$ was that we were, in some sense, unlucky enough that $\forall v_4$ was a part of $\varphi$. Hence, our newly substituted $v_4$ ended up getting *captured* by the quantifier $\forall v_4$ in $\varphi$. We do not want this, for the reason we just saw.

One way to get around this problem is by considering the formula $\varphi'$, which we define as $\forall v_6(S(v_6) = v_3)$. We see here that $\varphi'(v_4/v_3)$ becomes $\forall v_6(S(v_6) = v_4)$, which is a lot closer to $\varphi(v_5/v_3)$. What we have done is renamed the bound variable, i.e. the variable in the quantifier, from $v_4$ in $\varphi$ to $v_6$ in $\varphi'$. This is a safe move since $\varphi$ and $\varphi'$ are two equivalent formulae. So, to avoid our variables being captured when substituting in new terms, we always posit the caveat that we can rename bound variables to avoid this type of capture. We will not spend more time on this now, but as a matter of logical hygiene it needs to be mentioned.

We can now start to prove the diagonal lemma. The proof is inspired by Halbach and Leigh [Halbach and Leigh 2024]. As time is scarce, we will have to cheat slightly[6]. We will state, without proof, the following lemma.

6 This is okay, most mathematicians cheat a bit with their proofs.

LEMMA III.5
There is a function $sub(v_1, v_2, v_3)$ in $\mathcal{L}$ such that $PA \vdash \overline{\ulcorner \varphi \urcorner} = sub(\overline{\ulcorner \psi \urcorner}, \overline{n}, \overline{\ulcorner t \urcorner})$ just in case $\varphi = \psi(t/v_n)$.

The lemma says, in other words, that there is a function in $\mathcal{L}$ that allows us do perform substitution on the Gödel code of formulae. We will use that substitution operator to define our diagonal operator, which is what will allow us to prove the diagonal lemma.

DEFINITION III.6 (THE DIAGONAL OPERATOR)
Let $\varphi(v_n)$ be a formula where the only free variable is $v_n$, then we define $dia(\overline{\ulcorner \varphi(v_n) \urcorner}) = sub(\overline{\ulcorner \varphi \urcorner}, \overline{n}, \overline{\ulcorner \overline{\ulcorner \varphi \urcorner} \urcorner})$.

We recall that $\overline{\ulcorner \varphi \urcorner}$ is the numeral of the Gödel code of $\varphi$, and so $\overline{\ulcorner \overline{\ulcorner \varphi \urcorner} \urcorner}$ is the numeral of the Gödel code of the numeral of the Gödel code of $\varphi$.

Say, for instance, that the Gödel code of $\varphi$ was three
then $\ulcorner \varphi \urcorner = 3$
and so $\overline{\ulcorner \varphi \urcorner} = \overline{3} = S(S(S(0)))$.

Now, $S(S(S(0)))$ has the Gödel code $2^3 * 3^{11} * 5^3 * 7^{11} * 11^3 * 13^{11} * 17^1 * 19^{13} * 23^{13} * 29^{13}$.

Hence, $\overline{\ulcorner S(S(S(0))) \urcorner}$ is the numeral that has $2^3 * 3^{11} * 5^3 * 7^{11} * 11^3 * 13^{11} * 17^1 * 19^{13} * 23^{13} * 29^{13}$ many instances of $S$ appended to 0. In other words, if $\ulcorner \varphi \urcorner = 3$ then $\ulcorner \overline{\ulcorner \varphi \urcorner} \urcorner = 2^3 * 3^{11} * 5^3 * 7^{11} * 11^3 * 13^{11} * 17^1 * 19^{13} * 23^{13} * 29^{13}$.

We first prove a helpful lemma about the diagonal function that we will use in the proof of the diagonal theorem.

LEMMA III.7
$PA \vdash dia(\overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x)) \urcorner}) = \overline{\ulcorner \exists x(x = dia(\overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x)) \urcorner}) \wedge \varphi(x)) \urcorner}$
for any $\mathcal{L}$-formula $\varphi$ with a free variable $v_n$.

PROOF $dia(\overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x)) \urcorner}) = sub(\overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x)) \urcorner}, n, \overline{\ulcorner \overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x)) \urcorner} \urcorner}) =$
$\overline{\ulcorner \exists x(x = dia(\overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x)) \urcorner}) \wedge \varphi(x)) \urcorner}$.

Now we can prove Gödel's diagonal lemma.

> THEOREM III.4 (GÖDEL'S DIAGONAL LEMMA)
> If $\varphi(v_n)$ is a $\mathcal{L}$-formula with one free variable then there is a $\mathcal{L}$-sentence $\gamma$ such that $PA \vdash \gamma \leftrightarrow \varphi(\overline{\ulcorner \gamma \urcorner})$.

PROOF Let $\gamma = \exists x(x = dia(\overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x))\urcorner}) \wedge \varphi(x))$, then $PA \vdash dia(\overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x))\urcorner}) = \overline{\ulcorner \exists x(x = dia(\overline{\ulcorner \exists x(x = dia(v_n) \wedge \varphi(x))\urcorner}) \wedge \varphi(x))\urcorner} = \overline{\ulcorner \gamma \urcorner}$ by Lemma III.7. Hence, $PA \vdash \gamma \leftrightarrow \exists x(x = \overline{\ulcorner \gamma \urcorner} \wedge \varphi(x)) \leftrightarrow \varphi(\overline{\ulcorner \gamma \urcorner})$.

## III.3: The Undefinability of Truth

There is a vast literature about truth in mathematics, both what it means for something to be mathematical true *simpliciter* and what it would mean for something to formalise a notion of mathematical truth. For an excellent introduction to the latter notion, see Halbach's *Axiomatic Theories of Truth* [Halbach 2014].

Suppose that $\varphi(v_i)$ is a $\mathcal{L}$-formula formula meant to express truth.

There are many things we might want from this formula $\varphi$. Firstly, we might want that $\varphi(v_i)$ capture our intuition that a disjunction is true if and only if one of the disjuncts are true. In formal terms, we would then want $PA \vdash \varphi(\overline{\ulcorner \psi_1 \vee \psi_2 \urcorner}) \leftrightarrow \varphi(\overline{\ulcorner \psi_1 \urcorner}) \vee \varphi(\overline{\ulcorner \psi_2 \urcorner})$ for any formulae $\psi_1$ and $\psi_2$. However, this might not be enough. To really *use* the our truth formula $\varphi$. we might want it to commute with the disjunction in a more general sense. There is a function, usually denoted by $\dot{\vee}$, such that if $x = \ulcorner \psi_1 \urcorner$ and $x_2 = \ulcorner \psi_2 \urcorner$, then $x_1 \dot{\vee} x_2 = \ulcorner \psi_1 \vee \psi_2 \urcorner$. Then, we might want $\varphi(v_i)$ to satisfy the stronger requirement that $PA \vdash \forall x \forall y(\varphi(x \dot{\vee} y) \leftrightarrow \varphi(x) \vee \varphi(y))$. This is a stronger claim because it is saying that $\varphi$ commutes with $\dot{\vee}$ in a general sense, not just with any particular formulae. Hence, we are allowed to use this fact when using $\varphi$ in an inductive proof. The difference between whether permitting the more general case or the special case, and whether we accept instances of such a truth formula in our inductive scheme is of great importance, especially in the deflationism debate about truth. We will not go much further into it here, but again to read more about it see Halbach's book.

It seems that one *minimal*, absolutely necessary requirement for some formula to define truth is to satisfy the *Tarski-Biconditionals*. That means that if $\varphi$ is a formalised truth-formula, then it has to be the case that $PA \vdash \varphi(\overline{\ulcorner \psi \urcorner}) \leftrightarrow \psi$ for any $\mathcal{L}$-sentence $\psi$.

Tarski's Undefinability of Truth Theorem shows that there cannot be any $\mathcal{L}$-formula $\varphi$ satisfying the Tarski-Biconditionals.

THEOREM III.8 (TARSKI'S UNDEFINABILITY THEOREM)
There are no formula $\varphi(v_i)$ such that for all sentences $\psi$, $PA \vdash \varphi(\overline{\ulcorner \psi \urcorner}) \leftrightarrow \psi$.

PROOF Suppose for contradiction that we have such a formula $\varphi(v_i)$. By the Diagonal Lemma, there is a sentence $\lambda$ such that $PA \vdash \lambda \leftrightarrow \neg\varphi(\overline{\ulcorner \lambda \urcorner})$. By assumption, we have that $PA \vdash \lambda \leftrightarrow \varphi(\overline{\ulcorner \lambda \urcorner})$, which means that $PA \vdash \neg\lambda \leftrightarrow \neg\varphi(\overline{\ulcorner \lambda \urcorner})$. Hence, we get that $PA \vdash \lambda \leftrightarrow \neg\lambda$, a contradiction.

The $\lambda$ we constructed in the proof of Theorem III.8 is called the *liar sentence*, because it essentially says that it is false. That is because $\lambda$ is equivalent to $\neg\varphi(\overline{\ulcorner \lambda \urcorner})$.

No formula that is definable using the expressive resources of $PA$ can serve as an adequate representative of the predicate "....is true". This is remarkable, and by no means obvious; after all there are all sorts of (fairly complex) predicates that *are* definable using just the expressive resources of $PA$, e.g. ".... is even" and ".... is provable". But somehow the Tarski Biconditionals are beyond reach.

One response to this result is to introduce a *primitive* unary truth-predicate $T$ to our language and use that to define truth in $PA$. More formally, one would let $\mathcal{L}_T = \mathcal{L} \cup \{T\}$, and write down some axioms governing the use of $T$. There are many consistent ways of doing this, but the simplest is letting $TB = PA \cup \{T\overline{\ulcorner \psi \urcorner} \leftrightarrow \psi\}$ for any $\mathcal{L}$-sentence $\psi$. We can still define the liar sentence $\lambda$ as the result of applying the Diagonal Lemma to $\neg Tv_1$, however we will not have $TB \vdash T\overline{\ulcorner \lambda \urcorner} \leftrightarrow \lambda$ as $\lambda$ is not an $\mathcal{L}$-sentence. It contains the truth-predicate after all.

## Lecture IV: Gödel

Summary: We situate Hilbert's program historically, introduce the provability predicate, prove Gödel's First Incompleteness Theorem (noting the second), and discuss their genuine and alleged philosophical consequences.

> *We must not believe those, who today, with philosophical bearing and deliberative tone, prophesy the fall of culture and accept the ignorabimus. For us there is no ignorabimus, and in my opinion none whatever in natural science. In opposition to the foolish ignorabimus our slogan shall be: We must know. We will know.*
>
> – David Hilbert

### IV.1: Hilbert's programme

At the turn of the twentieth century, set-theoretic paradoxes (most famously Russell's) unsettled the emerging edifice of modern mathematics. David Hilbert proposed a corrective vision: make mathematical reasoning completely *explicit* by casting it into formal systems—fixed languages with precise axioms and mechanical rules of inference—and then vindicate this practice by proofs that use only elementary, "finitary" methods.

There were three strands to Hilbert's ambition. Mathematics should be demonstrably *consistent* (no contradictions), *complete* (every definite statement settled one way or the other), and ideally *decidable* by a uniform mechanical test. The spirit was optimistic: if proofs become transparent symbolic objects, perhaps we can check them automatically and certify the overall enterprise from outside.

Hilbert's programme requires us to reason, in a disciplined way, about what can be proven inside formal systems. If you want to guarantee, for a fixed theory $T$, that no contradiction can be derived, you are really trying to establish:

"There is *no* $T$-proof of a contradiction."

This is already a statement *about proofs*. To treat such a claim within the same mathematical framework that $T$ itself inhabits (arithmetic), one needs a way to *talk inside arithmetic* about proofs and provability. Concretely, two ideas meet here.

First, once a language and proof system are fixed, proofs are finite symbolic objects—strings from a small alphabet, arranged according to explicit rules. This means

they can be *encoded* as natural numbers (a Gödel numbering). Second, if strings and derivations can be encoded, then the key syntactic relations ("$p$ is a code of a well-formed formula", "$p$ is the code of a correct $T$-proof of the formula coded by $q$") become *arithmetical* relations between numbers.

With this in place, one can introduce, *inside the language of arithmetic*, a formula that expresses "provability in $T$". Informally we may write:

$$Prov_T(\ulcorner \varphi \urcorner) \quad \text{for "}\varphi \text{ has a } T\text{-proof".}$$

Here $\ulcorner \varphi \urcorner$ denotes the numeral naming the Gödel code of $\varphi$. The English-language claim "$\varphi$ is provable in $T$" is mirrored by a concrete arithmetical predicate $Prov_T$ that talks *about numbers*. In exactly the same way, the consistency claim for $T$ can be formulated as the single arithmetical sentence

$$Con(T) \;\equiv\; \neg Prov_T(\ulcorner 0 = 1 \urcorner),$$

which says: "there is no $T$-proof of $0 = 1$".

This move—to a definable arithmetical predicate of provability—opens the door to a striking new possibility: since arithmetic can now speak about its *own* proofs, sentences in arithmetic can be made to talk about their *own* provability status. That feedback loop will be central below.

**A short chronological orientation.** In 1930, Gödel proved the *Completeness Theorem* for first-order logic: every semantically valid inference has a formal proof. This was an important vindication of Hilbert's outlook at the level of pure logic. In 1931, however, Gödel discovered the limits that arise once we move from logic to a fixed, effectively axiomatized theory rich enough to express elementary arithmetic (such as Peano Arithmetic, or even much weaker systems). His *First Incompleteness Theorem* shows that any such consistent theory $T$ leaves some true arithmetical sentences unprovable in $T$. His *Second Incompleteness Theorem* shows that $T$ cannot, on pain of inconsistency, prove its own arithmetically formulated consistency statement $Con(T)$. A few years later (1936–37), Church and Turing gave precise accounts of "effective calculation" and showed there is no general decision procedure for first-order validity—the *Entscheidungsproblem* has a negative answer.

These results do not declare mathematics incoherent. But they do dash the hopes of the Hilbert programme; it turns out that there are sharp limits on what any *one* fixed, effectively axiomatized, consistent arithmetical theory can achieve. The First Incompleteness Theorem blocks the hope that a single such theory might settle every

arithmetical question: there will always be statements true in the intended natural numbers that escape its proofs. The Second Incompleteness Theorem blocks the hope that the theory might *internally* certify its own safety in the Hilbertian sense: the formalised consistency statement $Con(T)$ is not derivable in $T$ itself (provided $T$ really is consistent).

What endures is a moderated programme: proof theory flourishes in the study of *relative* consistency and strength comparisons between theories, and logic retains its completeness as a consequence relation. But the original expectation—that formalisation would yield, from within a single arithmetical framework, both complete capture of arithmetical truth and an internal finitary proof of consistency—was decisively dashed by Gödel's insight. In what follows, we fix a particular theory $T$ and a Gödel numbering, construct the provability predicate $Prov_T$ carefully, and then walk—slowly—through the proof of the First Incompleteness Theorem.

## IV.2: From arithmetisation to the provability predicate

We now move from informal talk *about* proofs to a way of *speaking inside arithmetic* about proofs. Throughout, fix once and for all a particular effectively axiomatized, consistent theory $T$ in the language of arithmetic. This may, for instance, be $PA$ as defined in Lecture III.

Recall that we write $\ulcorner \sigma \urcorner$ for the code (i.e. number) of a syntactic object $\sigma$ and $\overline{\ulcorner \sigma \urcorner}$ with the corresponding *numeral* of that number in the language of arithmetic, so that it can be substituted into formulas of arithmetic.

**Two levels to keep separate.**

- *Meta-level* (informal): "$q$ is the code of a $T$-proof of the formula coded by $p$."
- *Object-level* (inside arithmetic): a *single formula of arithmetic* that is true of exactly those pairs $(p, q)$ with that meta-property.

The point of arithmetisation is that these meta-statements about strings can be mirrored by number-theoretic statements about their codes.

**The various syntactic objects (at a glance):**

- Formulas and terms are finite strings; their codes are numbers.
- A *proof in* $T$ is a finite sequence of formulas $\varphi_1, \ldots, \varphi_k$ where each line is either an instance of an axiom-scheme of $T$ or follows from earlier lines by one of the fixed inference rules. The Gödel code of the proof is given by $\ulcorner \#\varphi_1 \#\varphi_2 \# \ldots \#\varphi_k \# \urcorner$.

- Basic syntactic relations ("is a variable", "is a well-formed formula", "line $k$ follows by Modus Ponens from lines $i$ and $j$", "this line is an instance of axiom-scheme $\mathsf{Ax}_r$", etc.) are all *effective* checks on finite strings, and therefore become definite arithmetical relations on their codes.

**The proof predicate.** Using the fixed coding and the fact that each step of a correct derivation can be checked by a mechanical test, we define a two-place arithmetical predicate

$$Proof_T(p, q)$$

that is intended to say: "$q$ is the code of a $T$-proof whose last line is the formula coded by $p$." Unpacking this within arithmetic means:

- $q = \ulcorner \#\varphi_1 \# \ldots \#\varphi_n \# \urcorner$ where $\ulcorner \varphi_i \urcorner$ is the Gödel-code of a $\mathcal{L}$-formulas;
- for each $k \leq n$, either $\ulcorner \varphi_k \urcorner$ is the code of an instance of an axiom of $T$, or there exist $i, j < k$ witnessing that $\varphi_k$ follows from $\varphi_i, \varphi_l$ by one of the fixed inference rules;
- and $\ulcorner \varphi_n \urcorner = p$.

All of these clauses are expressible by formulas of arithmetic because each amounts to a bounded, mechanical check on codes of finite strings. We list the following two key results about the $Proof_T$ predicate.

> Lemma IV.1
> For any $\mathcal{L}$-formula $\varphi$ and any numeral $\overline{k}$ we have that
>
> $T \vdash Proof_T(\overline{\ulcorner \varphi \urcorner}, \overline{k})$ just in case $k$ is the Gödel code of a $T$-proof of $\varphi$.

> Lemma IV.2
> For any $\mathcal{L}$-formula $\varphi$ and any numeral $\overline{k}$ we have that
>
> $T \vdash \neg Proof_T(\overline{\ulcorner \varphi \urcorner}, \overline{k})$ just in case $k$ is not the Gödel code of a $T$-proof of $\varphi$.

**The provability predicate.** From $Proof_T$ we obtain the unary *provability* predicate by simply adding an existential quantifier:

$$Prov_T(p) \quad \equiv \quad \exists q \, Proof_T(p, q).$$

Intuitively: "there exists a number $p$ that codes a correct $T$-proof of the formula coded by $q$." In particular, for any sentence $\varphi$,

$$Prov_T(\overline{\ulcorner \varphi \urcorner})$$

is an arithmetical sentence that (under our fixed coding) says exactly that $\varphi$ is provable in $T$. This allows us to *internalise* key metamathematical assertions. For example, the formalised *consistency* statement of $T$ is the single arithmetical sentence

$$Con(T) \quad \equiv \quad \neg Prov_T(\overline{\ulcorner 0 = 1 \urcorner}),$$

asserting that there is *no* code of a $T$-proof of a contradiction.

When we write $Prov_T(\overline{\ulcorner \varphi \urcorner})$, keep in mind:

- $Prov_T(\overline{\ulcorner \varphi \urcorner})$ is a *formula of arithmetic*; it talks about numbers.
- But if the provability predicate is doing its job, then the arithmetical formula $Prov_T(\overline{\ulcorner \varphi \urcorner})$ is designed to capture the (meta-level/English language) claim "$\varphi$ is provable in $T$".
- Different (reasonable) Gödel numberings yield extensionally equivalent provability predicates: the particular coding does not matter for any of our forthcoming arguments.

By giving arithmetic the resources to speak about *its own* proofs, we can form sentences that make claims about their *own* provability status.

The next step is to deploy the diagonal (fixed-point) idea: for any one-variable formula $\phi(x)$, we can produce a sentence $G$ that, in effect, says "I have property $\phi$." Specialising to $\phi(x) \equiv \neg Prov_T(x)$ yields a sentence that asserts of itself that it is *not* provable in $T$. This will be the engine of the First Incompleteness Theorem.

## IV.3: Gödel's First Incompleteness Theorem: a slow proof

We now put the pieces together. Recall that for our fixed theory $T$ and coding, there is a sentence $G_T$ with

$$G_T \quad \longleftrightarrow \quad \neg Prov_T(\overline{\ulcorner G_T \urcorner}).$$

In a plain reading: $G_T$ says of itself that it is not provable in $T$. Our task is to see—gently and methodically—what follows from the modest assumption that $T$ is *consistent*.

**A standing reminder about levels.** The symbol $Prov_T(\cdot)$ lives *inside arithmetic*.

When we gloss $Prov_T(\overline{\ulcorner \sigma \urcorner})$ as "$\sigma$ is provable in $T$", that is us speaking at the meta-level. The diagonal biconditional lets these two registers speak to one another in a controlled way.

### IV.3.1: Why $T$ does not prove $G_T$ (consistency suffices)

Suppose, for a moment, that $T$ did prove $G_T$. Then there would be an actual finite derivation whose code $p$ witnesses this. Because $Proof_T(p, q)$ expresses a *mechanical check* on a finite object, $T$ can verify any *particular* successful check: if $q$ really is the code of a correct $T$-proof of $G_T$, then $T$ proves $Proof_T(\overline{\ulcorner G_T \urcorner}, \overline{q})$, and hence also proves $Prov_T(\ulcorner G_T \urcorner)$.[7]

But $T$ also proves the diagonal instance $G_T \leftrightarrow \neg Prov_T(\overline{\ulcorner G_T \urcorner})$, and by elementary logic, from a proof of $G_T$ it can infer a proof of $\neg Prov_T(\overline{\ulcorner G_T \urcorner})$. Thus, on our supposition, $T$ would prove both $Prov_T(\overline{\ulcorner G_T \urcorner})$ *and* $\neg Prov_T(\overline{\ulcorner G_T \urcorner})$—a contradiction. Therefore a consistent $T$ does *not* prove $G_T$.

**Interim conclusion.** If $T$ is consistent, then $T \nvdash G_T$.

The moral is already instructive: the sentence that says "I am not provable in $T$" is indeed not provable in $T$. So far, nothing paradoxical—only a careful alignment of what the sentence says with what the theory can and cannot do.

### IV.3.2: Why $T$ does not prove $\neg G_T$ (two routes)

To reach *incompleteness* we must also see that $T$ does not prove the negation $\neg G_T$. There are two well-travelled paths.

*Route A: Gödel's original assumption ($\omega$-consistency)*[8]*.* Assume $T$ is $\omega$-consistent: it never proves an existential statement $\exists n\, R(n)$ while also proving, for each numeral $\overline{k}$, the corresponding denial $\neg R(\overline{k})$.[9]

7 You do not need the inner workings here. The only fact in play is that "being a correct next line of a proof" is a bounded, algorithmic property of codes; $T$ is designed to reason about such properties line-by-line. In standard expositions this is packaged as: if $Proof_T(n, m)$ is *true*, then $T \vdash Proof_T(\overline{n}, \overline{m})$.

8 Technically speaking, you only need to assume 1-consistency for this to work, but we are not going to define that here.

9 Intuitively, $\omega$-consistency rules out a "phantom witness": the theory may not assert "some number has property $R$" while simultaneously denying $R(0), R(1), R(2), \ldots$ one by one. Ordinary consistency only forbids a *direct* contradiction; $\omega$-consistency forbids this infinite pattern of near-misses.

Suppose toward a contradiction that $T \vdash \neg G_T$. By the diagonal biconditional this is equivalent to $T \vdash Prov_T(\overline{\ulcorner G_T \urcorner})$; so $T$ proves the existential statement "there exists a proof-code of $G_T$". Now fix any particular numeral $\overline{k}$. If $T$ proved $Proof_T(\overline{\ulcorner G_T \urcorner}, \overline{k})$, then $T$ could read of the proof $\overline{k}$ of $G_T$, and hence we would have that $T \vdash G$, a contradiction. Hence, for each $k$, $T$ proves $\neg Proof_T(\overline{\ulcorner G_T \urcorner}, \overline{k})$[10].

We are now in the forbidden configuration: $T$ proves the existential claim $\exists q\, Proof_T(\overline{\ulcorner G_T \urcorner}, q)$ but also, one by one, proves the denial of each instance $Proof_T(\overline{\ulcorner G_T \urcorner}, \overline{k})$. By $\omega$-consistency, this cannot happen. Therefore $T \nvdash \neg G_T$.

*Route B: Rosser's refinement (only consistency).* There is a variant sentence $R_T$ (Rosser's sentence) that balances "there is a proof of me" against "there is a shorter refutation of me" in such a way that mere *consistency* of $T$ suffices to show $T$ proves neither $R_T$ nor $\neg R_T$. The construction is a little more intricate, but the lesson is the same: under the weakest natural hypothesis (consistency), a suitably crafted self-referential arithmetical sentence defeats both proof and refutation.[11]

> **Second interim conclusion.** If $T$ is $\omega$-consistent (Gödel's route), then $T \nvdash \neg G_T$. Alternatively, using Rosser's refinement, plain consistency of $T$ already yields $T \nvdash \neg R_T$ for a closely related sentence $R_T$.

### IV.3.3: What have we proved, and what is the "truth status" of $G_T$?

Putting the two halves together, we have reached:

> **First Incompleteness (friendly form).** Let $T$ be effectively axiomatized, sufficiently strong to express elementary arithmetic, and consistent. Then $T$ is *incomplete*: there is a sentence (for instance $G_T$ under $\omega$-consistency[12], or Rosser's $R_T$ under mere consistency) that $T$ neither proves nor refutes.

A natural final question is: what is the *truth* value of $G_T$ in the intended natural numbers $\mathbb{N}$? The informal reading of $G_T$ says: "I do not have a $T$-proof." And indeed—externally,

10 We have defined the provability predicate such that $T \vdash proof_T(\overline{\ulcorner \varphi \urcorner}, \overline{q})$ just in case $q$ is the Gödel code of a proof of $\varphi$ and $T \vdash \neg proof_T(\overline{\ulcorner \varphi \urcorner}, \overline{q})$ just in case $q$ is not the Gödel code of a proof of $\varphi$, this is what allows us to infer $T \vdash \neg Proof_T(\overline{\ulcorner G_T \urcorner}, \overline{k})$ from the fact that $T \nvdash Proof_T(\overline{\ulcorner G_T \urcorner}, \overline{k})$.

11 Rosser replaces $\neg Prov_T(x)$ by a predicate that compares the codes of purported proofs and refutations, ensuring that one cannot "get ahead" of the other without violating plain consistency. We do not need the details here; what matters is that $\omega$-consistency can be dispensed with.

12 We used $\omega$-consistency to show that $T \nvdash \neg G_T$ and consistency to show that $T \nvdash G_T$. Luckily, $\omega$-consistency entails consistency, and so we only have to assume the former.

from our vantage point—there is no such proof, for if there were, $T$ would be inconsistent (by the argument of the first subsection). So $G_T$ is *true in* $\mathbb{N}$ but unprovable in $T$.

**Two morals to carry forward.**

- *Truth vs. provability.* Even for the humblest facts of arithmetic, what is true in $\mathbb{N}$ may outrun what a given effective, consistent theory $T$ can prove.
- *No single, final capture.* Any one $T$ that meets our mild conditions will leave something out. If you strengthen $T$ to catch the missing sentence, a new gap opens elsewhere.

In the next section we will see a further limitation: the very consistency statement $Con(T)$ that Hilbert hoped to secure by finitary means cannot—on pain of inconsistency—be proved *within* $T$ itself.

## IV.4: Gödel's Second Incompleteness Theorem: A Sketch

Let us take stock. We have shown that for any sufficiently strong recursively axiomatisation of arithmetic $T$, there is a sentence $G_T$ such that $T \nvdash G_T$ and $T \nvdash \neg G_T$. This means that one of Hilbert's hopes will never come to fruition, as there will be no recursively axiomatisation of arithmetic that can prove all and only the true sentences of $\mathbb{N}$.

However, Hilbert had more hopes than just that. He was a finitist, and so he thought that only finitary mathematics was meaningful. When we talked about infinitary objects, he took those to be ideal elements. That means that they do not actually *mean* anything, but they can be useful to prove things about finitary objects.

A half-good comparison is how mathematicians viewed the use of imaginary numbers when finding real solutions to polynomials. They did not thing that the imaginary numbers were real in any sense, but they were a useful tool to find say something meaningful about the real solution of a polynomial.

Similarly, Hilbert wanted to use infinitary mathematics, which he did not consider real, to say meaningful things about finitary mathematics. This raises a question: if these infinitary objects are not real, how do we know that it is safe to introduce them? Naively introducing objects before had lead to paradoxes, and so Hilbert required that the infinitary extension of the finitary mathematics had to be consistent. His idea was something like this: let $T$ be an axiomatisation of a finitary theory, and let $T \subset T'$ be the extensions which introduces infinitary objects. As long as $T$ can prove the consistency of $T'$, then anything $T'$ manage to prove about $T$ would be safe.

Gödel's Second Incompleteness Theorem essentially say that no recursively axiomatic theory of arithmetic with its own Gödel sentence can prove its own consistency. So, if $T'$ cannot prove it's own consistency, $T$ could not do it either. A second dream of Hilbert is therefore stopped by Gödel. Proving the Second Incompleteness Theorem is outside the scope of this lecture series, but we will state it and show how we get the conclusion that $T \nvdash Con(T)$ for any theory $T$ that satisfies the premises of the First Incompleteness Theorem.

> Theorem IV.3 (Gödel's Second Incompleteness Theorem)
> Let $T$ be a consistent effectively axiomatized theory of Arithmetic with its own Gödel sentence $G_T$, then for any $\mathcal{L}$-sentence $\varphi$, $T \vdash \neg Prov_T(\overline{\ulcorner \varphi \urcorner}) \rightarrow \neg Prov_T(\overline{\ulcorner G_T \urcorner})$.

Suppose now for contradiction that $T \vdash Con(T)$, that means that $T \vdash \neg Prov_T(\overline{\ulcorner 0 = 1 \urcorner})$. By Theorem IV.3 we get that $T \vdash \neg Prov_T(\overline{\ulcorner G_T \urcorner})$ and so $T \vdash G_T$ by $T \vdash G_T \leftrightarrow \neg Prov_T(\overline{\ulcorner G_T \urcorner})$. This contradict Gödel's First Incompleteness Theorem, and so we get that $T \nvdash Con(T)$. Hence, Hilbert's second hope has also been stopped.

## IV.5: Conclusion

Let us conclude here. In these four lectures, we have tried to show you what diagonalisation can do. It allowed us to prove that there are more real numbers than natural numbers, and it allowed us to show that there cannot be a set of all sets. Both of these uses of diagonalisation were somewhat informal, as they did not use the Diagonal Lemma from Lecture III. Nevertheless, they have the same spirit as the Diagonal Lemma.

We then turned our eye from informal mathematics to more formal mathematics in the sense that we defined a formal language of arithmetic and subsequently used that formal language to give arithmetic a recursive axiomatisation. Then we saw how the Diagonal Lemma allowed us to do similar diagonal argument *inside* our formal system as we had done with Cantor's diagonal argument or Russell's Paradox in the first lecture. We made two such diagonal arguments, firstly showing Tarski's Undefinability of Truth Theorem and then showing Gödel's Incompleteness Theorems. Both of these theorems show the limitation of what formal systems can do.

There is still one unanswered question; are there true unprovable statements about arithmetic? As we have seen, the theory $PA$ cannot prove $G_PA$, although $G_PA$ is a true statement in $\mathbb{N}$. To this you might say, how about the theory $PA + G_{PA}$? It can prove $G_{PA}$ and it is consistent as long as $PA$ is consistent. Let us call that theory $T_1$, then

$T_1$ has a Gödel sentence $G_{T_1}$ which is true in $\mathbb{N}$ but cannot be proven nor refuted by $T_1$. You might see where we are going here, we can define $T_2$ as $T_1 + G_{T_1}$, and so on. It turns out that this process can terminate, and we can end up with a theory $T$ that proves all the true sentences of Arithmetic. However, $T$ is not recursively enumerable, and so we cannot write down this system in any efficient way. After all, if we just wanted an axiomatic system that proves all the true sentences of arithmetic, why not just consider $\{\varphi : \mathbb{N} \models \varphi\}$?

The answer is, perhaps, that we want to show that all true sentences of arithmetic are provable for humans, in some idealised sense. To this question, we have a partial answer. If the idealised mathematician is a recursively enumerable axiomatic system, then the answer is no. Another way of putting this is to say, if the mind is a Turing machine, then there are true sentences about arithmetic we cannot prove. This is called Gödel's disjunction, as it forces us to accept that either the mind is not a (Turing) machine, or there are true sentences about arithmetic we cannot prove. If you manage to find out which of these disjuncts are true, then please do not hesitate to send us an email.

## References


Volker Halbach. *Axiomatic Theories of Truth*. Cambridge University Press, 2014.

Volker Halbach and Graham E. Leigh. *The Road to Paradox: A Guide to Syntax, Truth and Modality*. Cambridge University Press, 2024.

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