Gödel's incompleteness theorems

Ted Sider February 3, 2021

In 1931 Kurt Gödel proved a pair of remarkable and creative theorems in metamathematics that doomed Hilbert's program. Gödel proved two things:

First incompleteness theorem Any "minimally strong" axiomatic system is incomplete: for some sentence, neither it nor its negation can be proven

Second incompleteness theorem No "slightly more than minimally strong" axiomatic system can prove its own consistency

We'll need to explain what these mean; but here, in a preliminary way, is why they were so damaging to Hilbert's program and formalism.

The first theorem implies that there are serious limits to what the formalist is trying to do, namely, write down axioms for a given domain from which everything will follow.

Maybe formalists could live with this. But not the second. For, although this isn't obvious, it implies that it's just not possible to construct finitary proofs even for Arithmetic, let alone analysis or set theory.

Gödel's reasoning is very complicated, and we can't go through it in complete detail here. However, if we skip a few of the nuts-and-bolts details, the core ideas are astonishingly creative and interesting, and relatively easy to explain.

1. Gödel numbering

The first main idea is that of Gödel numbering, or coding: that linguistic objects in formal languages can be associated with natural numbers. (We already encountered something like this, where we showed that there is a way of associating a natural number with each finite string in a finite vocabulary.) The number associated with a formula can be thought of as a "code" of the formula. For instance, consider the language of arithmetic, in which we have a symbol $'$ for successor, a symbol for $\mathsf{0}$, a symbol for $+$, and a symbol for $\times$.

Here are some formulas:

$$\forall n\, 0 \neq n'$$
$$0 + 0'' = 0''' \times 0'''''$$
$$\forall x \forall y\ x + y = y + x$$

Each of these formulas would have a code. Maybe the code of the first is 5, the code of the second is 47, and the code of the third is 7,000,000.

Gödel also showed how to assign codes to *sequences* of formulas. This isn't very different from assigning codes to formulas, since a sequence of formulas, like an individual formula, is just a finite string of formulas, which are themselves finite strings of expressions in the alphabet. For example, here are two sequences of formulas:

1. $0 + 0'' = 0''' \times 0'''''$
2. $0 + 0 = 0$
3. $\forall x \exists y\ x = y$

1. $\forall n\, 0 \neq n'$
2. $0 \neq 0'''$

Maybe the first sequence has a code of 67 and the second has a code of 4,865,215.

2. Representing metalogic in arithmetic

Once we have codes for formulas, we can take a further step: coming up with arithmetic counterparts to statements about logic.

For example, consider this property: *containing at least one quantifier*. Some formulas have the property, such as $\forall x \forall y\ x + y = 0$. Others don't, such as $0' + 0'' = 0'''$. Given a method of coding, there is a corresponding property of natural numbers: the natural numbers that are codes of the formulas with the property. For example, maybe the codes of formulas with at least one quantifier turn out to be exactly the even numbers. Then *evenness* is the counterpart of *having at least one quantifier*.

For another example, consider the property of *being a sequence of formulas that counts as a legal proof in T*, for some axiomatic system of arithmetic T. This property also has a mathematical counterpart: the property of numbers that are the codes of sequences of formulas that are legal proofs in T. Thus, return to the second of the example sequences above:

1. $\forall n\, 0 \neq n'$
2. $0 \neq 0'''$

Maybe this is such a proof, if the first sentence is one of T's axioms, since the second sentence follows from the first by the logical rule of universal instantiation. So if the code of this sequence is 4,865,215, that number has the arithmetical counterpart of the property of being a legal proof in T.

3. Formalizing metalogic in the language of arithmetic

Now we can take an even further step. Suppose we have some metalogical property, and its arithmetic counterpart. It may be that we can come up with a formula in the language of arithmetic that *expresses* the arithmetic counterpart. For example, suppose that evenness is the arithmetic counterpart of being a formula with at least one quantifier; then the corresponding formula in the language of arithmetic would be:

$$\exists y\; x = y + y$$

This formula gives us a way to use the language of arithmetic to "talk about itself"! For consider this sentence:

$$\exists x \exists y\; x = y + y$$

This of course can be seen as a hum-drum statement about numbers: that there exists at least one even number. But via the coding, it also can be seen as saying something about the language of arithmetic: that there is at least one formula containing a quantifier. That is, this sentence "formalizes in the language of arithmetic" the metalogical claim that there is at least one formula in the language of arithmetic containing a quantifier.

For a more interesting example, return to the arithmetic property of the numbers that are codes of sequences of formulas that count as legal proofs in T. There is also an arithmetic property of numbers that are codes of sequences of formulas that contain some contradiction. Suppose these arithmetic properties are formalized, respectively, by two formulas in the language of arithmetic: T-Proof(x) and Contains-contradiction(x). (This will probably be extremely long formulas!) Then consider this sentence:

$$\sim\exists x(\text{T-Proof}(x) \wedge \text{Contains-contradiction}(x))$$

In addition to being some straightforward claim about numbers, this sentence is, given the coding, also a way of making a metalogical statement: namely, the statement that there does not exist any sequence that is both a legal proof in T and also contains a contradiction. That is, this sentence is the formalization of the claim "T is consistent"!

4. First incompleteness theorem

Let T be some "minimally strong" formal system. Here's what that means:

- The language of T includes, at least, the language of arithmetic
- The axioms of T are "effectively decidable"—that is, there is a mechanical procedure for telling what is an axiom. (For example, you can't say: the axioms of T are defined as all the true sentences of arithmetic.)
- A certain minimal amount of arithmetic can be proven from the axioms of T. (This includes certain general laws, such as $\forall x\, x + 0 = x$, but the theory need not contain a schema of induction.)

Gödel then shows how to formalize *in the theory* T the metalogical property of being provable-in-T. Here's what that means.

Let A be any formula. This formula has a code, n. Now consider this term in the language of T:

$$0\overbrace{''\dots'}^{n \text{ of these}}$$

That is: "0" followed by a bunch of $'$ signs—namely, n of them. This term is, so to speak, a name of the formula A in the language of arithmetic. Let's abbreviate this term by writing: "$\ulcorner A \urcorner$".

What Gödel did, then, is come up with a formula Provable(x) which "says in T" that x is provable in the sense that these two statements are true:

1. if A is provable in T then Provable($\ulcorner A \urcorner$) is provable in T
2. if A is not provable in T then $\sim$Provable($\ulcorner A \urcorner$) is provable in T

Then—and this is the most ingenious part—Gödel shows how to construct a sentence, G, that "says" *of itself* that it is not provable. That is, this sentence is a theorem of T:

$$G \leftrightarrow \sim\text{Provable}(\ulcorner G \urcorner) \tag{*}$$

Now we can prove the first incompleteness theorem, which says that if T is consistent, some sentence is such that neither it nor its negation is provable in T. (If T is inconsistent then it is trivially complete, since from a contradiction, *every* sentence is provable.) That sentence is G. I'll just give half of the proof:

> Suppose that T is consistent and that G *is* provable. Then by (P), Provable($\ulcorner G \urcorner$) is provable. But also, since (*) is provable, $\sim$Provable($\ulcorner G \urcorner$) is also provable, which contradicts the fact that T is consistent.

The other half is a little more complicated, but it shows that if T is consistent[1], then the supposition that $\sim G$ is provable leads to a contradiction.

Thus neither G nor $\sim G$ is provable, and so T is incomplete. Since T is required only to be able to prove a minimal amount of arithmetic, it follows that pretty much any mathematical theory of interest is not going to be capable of being fully axiomatized.

5. Second incompleteness theorem

In the previous section we showed this conditional statement to be true:

> If T is consistent then G is not provable

The next thing Gödel does is to show that, if the theory T is just a bit stronger than before—in particular, if certain arguments by induction can be done in T—then this argument can be *formalized in* T. Recall above that the sentence $\sim\exists x(\text{Proof}(x) \wedge \text{Contains-contradiction}(x))$ formalizes the claim that there exists no proof of a contradiction. Call this sentence CON. Gödel was able to show that this sentence is provable in T:

$$\text{CON} \rightarrow \sim\text{Provable}(\ulcorner G \urcorner) \tag{**}$$

Basically, he does this by running through an analogous argument to the one of the previous section, but within the theory T.

[1]Technically, "ω-consistent".

We can now argue as follows

i) Suppose for reductio that CON is provable.

ii) Then since (**) is provable, so is $\sim$Provable($\ulcorner G \urcorner$)

iii) Then, since (*) is provable, so is G

iv) But we earlier showed that G is *not* provable. Therefore, CON is not provable. That is, T cannot "prove its own consistency"

But T was an arbitrarily chosen theory that was "slightly more than minimally strong". Thus this shows that no minimally strong theory can prove its own consistency.

This is fatal to Hilbert's program. His finitary methods of proof can all be formalized in a theory T that is only slightly more than minimally strong. So if his methods succeeded in proving the consistency of T, one could prove the consistency of T within T. Since that's impossible, it follows that Hilbert's methods cannot prove the consistency of T. And so they certainly can't prove the consistency of arithmetic (since a proof of the consistency of arithmetic would ipso facto be a proof of T, which is a part of arithmetic), let alone stronger theories like analysis and set theory.