

Gödel's Incompleteness Theorem

*“There are more things in heaven and earth, Horatio,
Than are **proved** in your philosophy.”*
(W.Shakespeare, **with changes by me**)

Prologue

“Cretans, ever liars, wretched creatures, idle bellies”

(attr. to Epimenides, a Cretan)

Gödel: “This statement is unprovable in the formal system S ”

Assume false \Rightarrow provable \Rightarrow true. Contradiction!

Cantor: “Set $\mathcal{P}(X)$ of all subsets of X has greater cardinality than X ”

Russel: “Barber of Sevilla shaves everyone in Sevilla who doesn't shave himself. Who shaves the barber?”

The notion of proof

1. What are (little) proofs made of?

- (a) All things nice.
- (b) Sugar and spice.
- (c) Fancy words and symbols.
- (d) Concrete and steel.

2. How can you tell a proof from a novel?

- (a) It says "Proof" on top.
- (b) It can be verified (checked).
- (c) It can be published in Ann. Math.
- (d) You won't get the Booker prize for it.

3. What does a proof prove?

- (a) Usually something obvious.
- (b) That whoever wrote it is genius and everyone else is a fool.
- (c) That one can obfuscate even the simplest of ideas.
- (d) Something that can be extracted from the proof (usually the last statement).

Deductive system

Definition 1. A triple $\langle \Delta, D, \delta \rangle$ is called *deductive system*, where

- Δ is alphabet (of proofs).
- $D \subset \Delta^\infty$ is the set of all proofs and is decidable.
- δ is the function extracting the result of the proof.

Corollary 1. *There are countably many proofs.*

Fundamental couple

Definition 2. $\langle \Lambda, T \rangle$ is called *fundamental couple*, where

- Λ is alphabet (of statements).
- $T \subset \Lambda^\infty$ is set of all *true* statements.

Example 1. Statements “ $1 + 3 = 4$ ” and “ $\exists x: x + 1 = 2$ ” are true in arithmetics. But there is no universal notion of truth, e.g. a fundamental couple where “ $2 + 2 = 5$ ” is true is perfectly legitimate as well.

The function δ in a deductive system — the result extraction function — acts from D (proofs) into Λ^∞ (statements).

Definition 3. $\langle \Delta, D, \delta \rangle$ defined over $\langle \Lambda, T \rangle$ is called *consistent* if $\delta(D) \subset T$ and is called *complete* if $\delta(D) \supset T$.

“For sufficiently rich $\langle \Lambda, T \rangle$ there is no $\langle \Delta, D, \delta \rangle$ which is both consistent and complete”

Computable functions, enumerable and decidable sets

Computable function is the one you can express as a finite program on a computer with infinite memory.

Definition 4. If a program never stops for initial data n , we say the function is *undefined* for n .

Definition 5. Set A is *enumerable* if it is either empty or \exists computable f defined on whole \mathbb{N} , s.t. $f(\mathbb{N}) = A$.

Definition 6. Set A is *decidable* in B if $\chi_A: B \rightarrow \{0, 1\}$ is computable and defined on whole of B .

Example 2. \mathbb{N}^2 is enumerable.

Example 3. Δ^∞ is enumerable.

Example 4. The set of even numbers is decidable in \mathbb{N} .

Enumerable and decidable sets (cont)

Lemma 1. A set $S \subset \mathbb{N}$ is decidable in \mathbb{N} if and only if both S and $\mathbb{N} \setminus S$ are enumerable.

Proof. “ \Rightarrow ” is an exercise.

“ \Leftarrow ”: let $S = f(\mathbb{N})$ and $\mathbb{N} \setminus S = g(\mathbb{N})$. To pinpoint $n \in \mathbb{N}$, compute

$$f(0), g(0), f(1), g(1), f(2), g(2), \dots$$

until you get n . If it was produced by f , $n \in S$, otherwise $n \in \mathbb{N} \setminus S$. □

Lemma 2. $\delta(D)$ is enumerable.

Proof. If $\delta(D)$ is empty, it is enumerable. Otherwise $\exists d \in \delta(D)$. Let $\xi: \mathbb{N} \rightarrow \Delta^\infty$ enumerate Δ^∞ . Define $f: \mathbb{N} \rightarrow \Lambda^\infty$ by

$$f(k) = \begin{cases} \delta(\xi(k)) & \text{if } \chi_D(\xi(k)) = 1, \\ d & \text{otherwise.} \end{cases}$$

Then f enumerates $\delta(D)$. □

Criterion of incompleteness

Theorem 1. *If for a fundamental couple $\langle \Lambda, T \rangle$ we can find*

- *non-enumerable set $Q \subset \mathbb{N}$ and*
- *a computable function $\phi: \mathbb{N} \rightarrow \Lambda^\infty$ satisfying*
 $\phi(Q) \subset T$ *and* $\phi(\mathbb{N} \setminus Q) \subset \Lambda^\infty \setminus T,$

then it is impossible to define a deductive system on $\langle \Lambda, T \rangle$ which is both consistent and complete.

Example 5. $\phi(x) = "x = 3"$ describes the set $\{3\}$ and $\phi(x) = "\exists y: 2y = x"$ describes the set of even numbers.

Proof of theorem 1. Assume the contrary: $\delta(D) = T$.

Then T must be enumerable: $\exists \tau: \tau(\mathbb{N}) = T$.

Since Q is not enumerable, $\exists q \in Q$. Define $f: \mathbb{N}^2 \rightarrow \mathbb{N}$ by

$$f(k, n) = \begin{cases} k & \text{if } \phi(k) = \tau(n), \\ q & \text{otherwise.} \end{cases}$$

Then, if $\xi: \mathbb{N} \rightarrow \mathbb{N}^2$ enumerates \mathbb{N}^2 , $\phi(\xi)$ enumerates Q . □

Battle plan

To show incompleteness we need a non-enumerable set Q and a formula describing it.

Plan:

1. Construct an enumerable set S with non-enumerable complement.
2. Describe (extended) arithmetic language.
3. Show that EAL can describe the set of results of any computable function \Rightarrow can describe any enumerable set \Rightarrow can describe above S .
4. Notice that if formula F describes S , then formula $\neg F$ describes $\mathbb{N} \setminus S$.
5. Bingo!

Programming Language

$R(a)$ denotes memory register number a (there are infinitely many registers). Language commands are:

- $R(a) \leftarrow b$
- $R(a) \leftarrow R(b)$
- $R(a) \leftarrow R(b) + R(c)$
- $R(a) \leftarrow R(b)R(c)$
- GOTO n
- IF $R(a) = R(b)$ GOTO n
- STOP

All registers are initialised to zero. The argument of the function is placed into $R(0)$, the result will be placed there as well.

Programming Language (cont)

Example 6. What does the following program compute?

1. IF $R(1) = R(0)$ GOTO 7
2. $R(3) \leftarrow 1$
3. $R(2) \leftarrow R(1) + R(3)$
4. IF $R(2) = R(0)$ GOTO 7
5. $R(1) \leftarrow R(2)$
6. IF $R(0) = R(0)$ GOTO 2
7. $R(0) \leftarrow R(1)$
8. STOP

Example 7. Simplified program from previous example.

1. IF $R(1) = R(0)$ RETURN $R(1)$
2. $R(2) \leftarrow R(1) + 1$
3. IF $R(2) = R(0)$ RETURN $R(1)$
4. $R(1) \leftarrow R(2)$
5. GOTO 2

Exercise 1. Design machine code Π for the above PL.

Exercise 2. Design syntax checker for PL.

Exercise 3. Design emulator for PL in PL.

Exercise 4. Change PL so that previous exercise is actually doable.

Universal function

Lemma 3. There is a program Φ that, given (k, n) ,

1. Generates k -th element of Π^∞ .
2. Checks if it is a valid program.
3. Runs it on input n .

Definition 7. A program such as the above is called *universal*: $\forall g, \exists k \in \mathbb{N}$ s.t. $g(\cdot) = \Phi(k, \cdot)$.

Define f by $f(n) = \Phi(n, n)$ (it might not be defined for some values of n).

There is no computable function g which is everywhere different from f !

Wouldn't $g(n) = f(n) + 1$ be different from f ?

No, unless f is everywhere defined!

Function with unresolvable domain

Theorem 2. *There is a computable function which does not admit everywhere defined extensions. Therefore its domain is not decidable.*

Proof. $g = f + 1$ is such a function: an everywhere defined extension would be everywhere different from f , which is impossible.

If $\text{Dom}(g)$ were decidable, then

$$\tilde{g}(n) = \begin{cases} g(n) & \text{if } n \in \text{Dom}(g), \\ 0 & \text{otherwise} \end{cases}$$

would be everywhere defined computable extension of g .

□

Domain of a computable function

Lemma 4. *Domain of every computable function g is enumerable.*

Corollary 2. *Therefore $S = \text{Dom}(g)$ is enumerable (but not decidable) $\Rightarrow \mathbb{N} \setminus S$ is not enumerable.*

Proof of Lemma 4. If $\text{Dom}(g) = \emptyset$, it is enumerable. Otherwise $q \in \text{Dom}(g)$. Define $R: \mathbb{N}^2 \rightarrow \mathbb{N}$ by

for given (m, n) run g on value n for m steps of the program. If g stopped, return n , otherwise return q .

Then if ξ enumerates \mathbb{N}^2 , $R(\xi)$ enumerates $\text{Dom}(g)$. □