Introduction to Gödel theory

by

Jens Hee
https://jenshee.dk

August 2025

Change log

17. May 2023

1. Document started.

31. January 2024

1. "Truth that cannot be proven" clarified

6. May 2025

1. "Truth that cannot be proven" changed once more

26. June 2025

1. Gödel number update.
2. Diagonalization Lemma explained.
3. Some of the content rewritten

6. August 2025

1. Diagonalization Lemma updated.
2. Prolog program improved

11. August 2025

1. Several errors corrected.
2. Section about The diagonalization lemma rewritten
3. Prolog code replace by c-like pseudo code

Contents

1 Introduction **2**

2 The gödel number **2**

3 Self Referencing Sentence **2**

4 Some definitions **3**
4.1 An example 3

5 The Diagonalization Lemma **3**
5.1 Programming approach 4

6 Truth that cannot be proven **5**

1 Introduction

In 1931 Kurt Gödel published his two inconsistency Theorems. Unfortunately it is often unclear what Gödel proved and how he did it. I shall in the following try to clarify some aspects of his works, which are often misinterpreted and incorrectly presented. I hope this paper can be useful to those, who wish to dig deeper into the works of Kurt Gödel and other aspects of Formal Logic.

2 The gödel number

A fact that was not so obvious 100 years ago is that a one to one relation exists between finite length sentences and numbers. We are to day used to store text on a hard drive, and expect that we load in the text again. On the hard drive the text is stored as a bit-pattern and thereby as an integer number. Thus, there is a one to one correspondence between pieces of text and integer numbers. If the text is based on an alphabet with L characters, then $P = ceil(log_2(L))$ bits is required for each character. If a text (sentence) has N characters then the sentence number g is given by:

$$g = \sum_{n=0}^{N-1} 2^{Pn} b_n$$

where b_n is the number corresponding to the n'th letter in the sentence.
Gödel had a different approach. If, as before, each letter in the alphabet is given a number b_n, the sentence number is given by:

$$s = 2^{b_1} 3^{b_2} 5^{b_3} 7^{b_4} \ . \ . \ . \ . \ p_N^{b_N}$$

where p_n is the n'th prime number and b_n is the integer for the n'th symbol in the sentence. Since prime number factoring is unique, the b_n are uniquely given by the sentence number in both cases. See D. Hofstader: Limits of Logic: The Gödel Legacy

There is thus an isomorphism between the integers and the set of sentences and since it doesn't matter which approach is used, the number identified with a sentence is just called the gödel number in the following.

3 Self Referencing Sentence

In Formal Logic it is required that a sentence is without self reference, as this would lead to infinite loops in a the argumentation. But Gödel found a backdoor to a kind of self reference, that lead to his astonishing conclusions.
An example of a self referencing sentence is:

This sentence is false

This leads to argumentation like: If it is false, it must be true, so it must be false etc. And these kind of sentences must be discarded.

4 Some definitions

A formula (statement) is required to be well formed. A set of rules defines if the formula is well formed. Among the rules is the requirements that a formula must not be self referencing.

Formulas can be divided into two groups: closed and open formulas. Closed formulas have no free variables, whereas open formulas may have one or more free variables (e.g. $\forall x, \exists y(x > y)$). In the following closed formulas are called sentences and since only open formulas with one free variable is considered they are just called formulas.

If a sentence can be proven true, i.e it is a logical consequence of the fundamental building blocks, it is called a theorem. The gödel number for such a sentence is then called a theorem number. Before Kurt Gödel published his theories it was assumed that all well formed sentences could be proven either true or false. E.g. the German mathematician David Hilbert said: "Wir Wollen Wissen", meaning that this would be possible. As we shall see there is also a third possibility.

4.1 An example

Consider the sentence:

The number 64 is not a prime number

It can be proven to be true and consequently it is a theorem.

5 The Diagonalization Lemma

The diagonalization lemma can be formulated as follows:

Given any formula $A(x)$ with one independent variable, it is possible to construct a sentence D biconditional with $A(\ulcorner D \urcorner)$

where $\ulcorner D \urcorner$ is the gödel number of D.
More formally we can write:

$$\mathbf{F} \vdash \mathbf{D} \leftrightarrow \mathbf{A}(\ulcorner \mathbf{D} \urcorner)$$

or

$\mathbf{D}$ can be proved if and only if $\mathbf{A}(\ulcorner \mathbf{D} \urcorner)$ can be proved

The usefulness of the lemma will become clear in section 6

In the following the construction of the sentence $\mathbf{D}$ and an informal proof of the lemma is given using a c-like programming example. The above formulation of the lemma follows by replacing $\phi(k)$ with $\mathbf{D}$ and ψ by $\mathbf{A}$.

5.1 Programming approach

ψ = "Some well formed formula (WFF) with one independent variable x"
ψ(n) = ψ.replace("x", n.toString()), n is some int value

// The text to harddisk approach is used;
int s = save(S) // i.e. s = $\ulcorner S \urcorner$
string S = load(s) // the sentence S has the gödel number s

// Definition of ϕ by giving a function for $\phi(n)$ for any int n
ϕ(n) = ψ(m) where subst(n, n, out m)

// ϕ becomes a WWF, since ψ is a WFF
// Definition of k
int k = save(ϕ)

// Now insert k into subst and we get:
m = save(load(k).replace("x", k.toString())) = save(ϕ(k))
ϕ(k) = ψ(save(ϕ(k)))

// Definition of g
g = save(ϕ(k))

ϕ(k) = ψ(g)

subst(p, v, out r){
if (WFF(load(p))
r = save(load(p).replace("x", v.toString()))
else
r = 0}
// Note that if subst(x1, x1, out y1) and subst(x2, x2, out y2) and x1 $\neq$ x2 then y1 $\neq$ y2

$\phi(k)$ can be proved if and only if ψ(g) can be proved or more formally;

F $\vdash \phi(k) \leftrightarrow \psi(\ulcorner \phi(k) \urcorner)$

6 Truth that cannot be proven

If we let:

A= "The number x is not a theorem number"

and construct D and g as above we have the sentence:

G: The number g is not a theorem number

where g is the gödel number of the sentence D constructed from A.

Let us now try to prove the sentence G:
Assume the sentence G is true, this means that g is not a theorem number and neither D nor G are theorems, this means that G is false in contradiction with the assumption.
Assume the sentence G is false, this means that g is a theorem number and D and thereby G are theorems and thereby true, this is also in contradiction with the assumption.
It is thus neither possible to prove the sentence true nor false and it must therefore be regarded undecidable. A sentence is on other hand either true or false, it is not false and must therefore be true, but it is not possible to prove it using the rules of logic.
We can thus conclude:

There are true sentences, that cannot be proven.

It is sometimes claimed that Gödel has smuggled self reference in through a backdoor like a Trojan Horse by making an isomorphism between sentences and integers. The sentence G only expresses something about a number and is not self referencing, but as the number g is a sentence number, we are in a way dealing with self referencing.
The statement "There are true sentences, that cannot be proven" may sound strange from a formal point of view, but if mathematics is regarded as something we do not invent but discover, then it may have a meaning, but this is more of a philosophical issue.