Gödel's Proof

3. Incompleteness Theorems

S M Srivastava

After presenting the technique developed by Gödel to examine questions about a theory such as Number Theory, Set Theory by the theory itself, we present the surprising Incompleteness Theorems of Gödel.

S M Srivastava is with the Indian Statistical Institute, Kolkata. His current interests lie in descriptive set theory, mathematical logic and topology. He loves teaching mathematics.

1. Preliminaries to The Incompleteness Theorem

In the last two parts of this article, we presented an overview of the First Order Logic. We now proceed to state and give an outline of the proof of Gödel's famous incompleteness theorem. Whereas it is not very hard to give the main ideas involved in the proof, giving the complete proof is quite long drawn and a tedious job. However, after understanding all the concepts involved, it is quite possible to appreciate the beauty, depth and impact of the result. Further, it only requires patience to understand the concepts and results presented below. For easy reading, we have divided this section into various subsections. In each subsection we present a main concept and state some relevant results.

[1] Part 1. An Introduction to Mathematical Logic, *Resonance*, Vol.12, No.2, pp.59–70, 2007.
Part 2. Semantics of First Order Theory and Hilbert's Program, *Resonance*, Vol.12, No.3, pp.35–46, 2007.

1.1 *Recursive Functions*

Let $\mathbb{N}$ denote the set of all natural numbers $0, 1, 2, \ldots$. Intuitively, a function $f : \mathbb{N}^k \to \mathbb{N}^\ell$ is called *recursive* if there is an algorithm (a computer program in modern avatar) that computes f (in the sense that on every input $(n_1, \cdots, n_k)$, the algorithm terminates and outputs $f(n_1, \cdots, n_k)$). We proceed to give a precise definition of recursive functions now.

Keywords
Recursive functions, Church's thesis, Gödel numbers, representability, set theory.

Here is a list of completely trivial functions:

Successor Function: $S(n) = n + 1$;

The year 2006 marked the birth centenary of the great mathematician-logician Kurt Gödel. Gödel's incompleteness theorem has far reaching implications concerning the foundations of mathematics. This article by S M Srivastava can be viewed upon as a prelude for an understanding/appreciation of Gödel's incompleteness theorem. The article is in three parts. The first part will be accessible to anyone with mathematical preparation at the level of II Year BSc/BTech/BE. The second and the third parts presuppose a certain level of mathematical maturity; students at the master's level may find them useful.

The reader is also urged to look at the article 'Gödel's Explorations in Terra Incognita' by Vijay Chandru in the July 2001 issue of *Resonance*, featuring Gödel. One may also see the April 2006 issue of *Notices of the American Mathematical Society* devoted to Gödel.

Editors

Constant Functions: For any $k \geq 1$ and any $p \in \mathbb{N}$,

$$C_p^k(n_1, \cdots, n_k) = p\ ;$$

Projection Functions: For any $k \geq 1$ and $1 \leq i \leq k$,

$$\pi_i^k(n_1, \cdots, n_k) = n_i\ .$$

These functions are called *initial functions*. Intuitively, it is quite easy to see that initial functions are computable: To compute the successor function, represent each integer by its binary expansion. Clearly, binary addition by 1 is quite a mechanical procedure. To compute a constant map C_p^k, output p on all inputs. To compute π_i^k given an input $(n_1, \cdots, n_k)$, output n_i.

Now we give some 'constructive' schemes for defining a function f with integer arguments from given functions.

Composition: Given $h(n_1, \cdots, n_m)$ and $g_i(\ell_1, \cdots, \ell_k)$, $1 \leq i \leq m$, define

$$f(\ell_1, \cdots, \ell_k) = h(g_1(\ell_1, \cdots, \ell_k), \cdots, g_m(\ell_1, \cdots, \ell_k)).$$

Note that if there exist 'mechanical procedures' to compute h and $g_1, \cdots, g_k$, then there is a 'mechanical procedure' to compute f.

Primitive Recursion: Given a m-ary function g and a $(m+2)$-ary function h, we define a $(m+1)$-ary function f by

$$f(0, n_1, \cdots, n_m) = g(n_1, \cdots, n_m),$$

$$f(k+1, n_1, \cdots, n_m) = h(f(k, n_1, \cdots, n_m), k, n_1, \cdots, n_m).$$

The scheme of primitive recursion is a general form of definition of functions by induction. It should be noted that m may be 0 and that a 0-ary function is nothing but a constant. Thus given a natural number p and a 2-ary function h, this procedure defines a sequence $\{x_k\}$ by induction: Set $x_0 = p$ and $x_{k+1} = h(x_k, k)$. Intuitively it should be obvious that if there are 'mechanical procedures' to compute g and h, there is a 'mechanical procedure' to compute f.

Intuitively it should be obvious that if there are 'mechanical procedures' to compute *g* and *h*, there is a 'mechanical procedure' to compute *f*.

Minimalization: Given a function g of $(m+1)$ variables such that for every $(n_1, \cdots, n_m)$, there is a k such that $g(k, n_1, \cdots, n_m) = 0$. We then define f by

$$f(n_1, \cdots, n_m) = \text{smallest } k \text{ such that } g(k, n_1, \cdots, n_m) = 0.$$

It is not very hard to see that if there is a 'mechanical procedure' to compute g, then f can be computed mechanically: On any input $(n_1, \cdots, n_m)$, set $k = 0$ and compute $a = g(k, n_1, \cdots, n_m)$. If $a = 0$, output k and stop. Else, increase k by 1 and do the same. By our hypothesis, in a finite number of steps one arrives at a k such that $a = g(k, n_1, \cdots, n_m) = 0$. Then output k and stop.

Finally, a function f is called *recursive* if it can be defined by successive applications of composition, primitive recursion and minimalization starting with initial functions. More precisely, *the set of recursive functions is the smallest collection of number theoretic functions which contains all initial functions and which is closed under composition, primitive recursion and minimalization.* It is now accepted that this is precisely the collection of all those functions which are intuitively computable. This assertion is known as *Church's Thesis.*

The set of recursive functions is precisely the collection of all those functions which are intuitively computable. This assertion is known as *Church's Thesis*.

It should be mentioned that there are alternate (and in some cases more intuitive) models of computations such as λ-calculus, Turing Machines and Random Access Machines. Since all the known models of computations give the same class of computable functions and predicates, it supports the belief in Church's Thesis.

A systematic development will show that common arithmetical functions like addition, subtraction, multiplication, division, exponentiation, etc., and some common predicates like <, ≤ are recursive.

Example. Consider the following inductive definition of $m + n$:

$$m + 0 = m,$$
$$m + (n + 1) = (m + n) + 1.$$

It can be easily seen that the addition $m + n$ is a binary recursive function. A systematic development will show that common arithmetical functions like addition, subtraction, multiplication, division, exponentiation, etc., and some common predicates like $<$, $\leq$ are recursive. It is also clear that the sets of recursive functions and recursive predicates are countable. So, there are non-recursive functions and predicates. However, usually it is hard to prove that a function is not recursive and many of the surprising results including Gödel's incompleteness theorem, Hilbert's tenth problem hinge on proving that certain functions and predicates are not recursive.

We say that $P \subset \mathbb{N}^k$ is *recursive* if its characteristic function is recursive. Thus, by Church's Thesis, recursive sets are precisely those sets $P \subset \mathbb{N}^k$ corresponding to which there is an algorithm to decide whether a k-tuple of natural numbers $(n_1, \cdots, n_k)$ belongs to P or not.

Usually it is hard to prove that a function is not recursive and many of the surprising results including Gödel's incompleteness theorem, Hilbert's tenth problem hinge on proving that certain functions and predicates are not recursive.

A non-empty subset of $\mathbb{N}^k$ is called *semi-recursive* or *recursively enumerable* (r.e., in short) if it is the range of a recursive function $f : \mathbb{N} \to \mathbb{N}^k$. Empty set is defined to be r.e..

It is easy to see that the set of recursive (semi-recursive) subsets of $\mathbb{N}^k$ is closed under finite unions and finite intersections. Further, if $P \subset \mathbb{N}^k$ is recursive (semi-recursive) and if $f_i : \mathbb{N}^m \to \mathbb{N}$, $1 \leq i \leq k$, are recursive, then the m-ary predicate $Q \subset \mathbb{N}^m$ defined by

$$Q(\ell_1, \cdots, \ell_m) \Leftrightarrow P(f_1(\ell_1, \cdots, \ell_m), \cdots, f_k(\ell_1, \cdots, \ell_m))$$

is also recursive (respectively, semi-recursive). Finally, the class of recursive sets is closed under complementation.

Other important facts about these concepts are:

Every recursive set is semi-recursive;

A set $P \subset \mathbb{N}^k$ *is semi-recursive if and only if it is the projection of a recursive set* $Q \subset \mathbb{N}^k \times \mathbb{N}$*;*

A set $P \subset \mathbb{N}^k$ *is recursive if and only if both* P *and* $\mathbb{N}^k \setminus P$ *are semi-recursive;*

and

There are semi-recursive sets that are not recursive.

Using the second assertion above, it is not hard to see that if $P \subset \mathbb{N}^k$ is semi-recursive, then there is an algorithm that halts on an input $(n_1, \cdots, n_k)$ if and only if the input belongs to P. Using one of the most beautiful ideas of Gödel described in the next section, one can show that the converse of this assertion is also true.

1.2. *Gödel Numbers*

The next concept, called *Gödel number*, is a beautiful idea due to Gödel. It represents syntactical objects, e.g., symbols, terms, formulae, proofs, etc., of a theory by natural numbers. Consequently, statements about syntactical objects – formulae, theorems, etc., – are expressed in terms of numbers. Its importance and beauty cannot be overemphasized. It has the potential of converting a metamathematical statement into a number theoretic statement. Thus, the problem of whether a metamathematical statement is true or not is translated into a number theoretic problem. This idea also plays a significant role in the theory of computation. To elaborate a bit more, using the same idea, one can code each algorithm by an integer, or one can translate questions about algorithms into number theoretic problems.

The problem of whether a metamathematical statement is true or not is translated into a number theoretic problem by the concept of *Gödel number*. This idea also plays a significant role in the theory of computation.

The main idea is simple. For instance, we can talk of rational numbers within Peano arithmetic as follows: Use 2^n when we want to talk on n; use 3^n for $-n$; use $5^{3^5} \cdot 7^{2^6}$ for $-5/6$. In general, let p/q be a rational number with p and q relatively prime and $q > 0$. We represent p/q by $5^{2^p} \cdot 7^{2^q}$ if $p \geq 0$ and by $5^{3^p} \cdot 7^{2^q}$ if $p < 0$. It is easy to see that there is an algorithm that decides whether a natural number codes a rational number or not. Further, if a natural number n codes a rational number, the algorithm recovers the rational number that n codes.

To simplify the matter, we assume that T has only finitely many non-logical symbols.

In the first step we assign a *symbol number* to each symbol of $L(T)$.

Set $SN(x_i) = 2i$, $i \geq 0$; $SN(\neg) = 1$; $SN(\vee) = 3$; $SN(\exists) = 5$; $SN(=) = 7$; if α is the ith non-logical symbol, we set $SN(\alpha) = 7 + 2i$.

It is easy to see that the set of all symbol numbers is recursive. In other words, there is an algorithm to decide whether an integer is a symbol number or not. Further, intuitively it is easy to see that there is an algorithm such that given a symbol number n, the algorithm recovers the symbol whose symbol number is n.

Let $2 = p_0, p_1, \ldots,$ be the increasing enumeration of all prime numbers. It is easy to prove that the function $n \to p_n$ is recursive. Also, for any $k \geq 0$, the map

$$(n_0, \cdots, n_{k-1}) \to \langle n_0, \cdots, n_{k-1} \rangle = p_0^{n_0+1} \cdots p_{k-1}^{n_{k-1}+1}$$

is recursive. A number of the form $\langle n_0, \cdots, n_{k-1} \rangle$ is called a *sequence number*. The set of all sequence numbers is recursive. Thus, there is an algorithm to decide whether a number is a sequence number or not. Further, there is an algorithm such that given any sequence number n, it recovers the sequence that it codes.

Let t be a term and A a formula of T. We now define the *Gödel number* $\lceil t \rceil$ and $\lceil A \rceil$ of t and A respectively by induction as follows.

If t is a variable or a constant, set

$$\lceil t \rceil = \langle SN(t) \rangle.$$

If f is a n-ary function and $t_1, \cdots, t_n$ are terms whose Gödel numbers have been defined, we set

$$\lceil ft_1 \cdots t_n \rceil = \langle SN(f), \lceil t_1 \rceil, \cdots, \lceil t_n \rceil \rangle.$$

If A is an atomic formula $pt_1 \cdots t_n$, define

$$\lceil A \rceil = \langle SN(p), \lceil t_1 \rceil, \cdots \lceil t_n \rceil \rangle.$$

If A is $\neg B$, and if $\lceil B \rceil$ has been defined,

$$\lceil A \rceil = \langle SN(\neg), \lceil B \rceil \rangle.$$

If A is $\vee BC$,

$$\lceil A \rceil = \langle SN(\vee), \lceil B \rceil, \lceil C \rceil \rangle.$$

If A is $\exists x B$,

$$\lceil A \rceil = \langle SN(\exists), \lceil x \rceil, \lceil B \rceil \rangle.$$

Now we have associated a number (called *Gödel number*) to each term and each formula of T so that there is an algorithm which on any input n, decides whether it is the Gödel number of a term (or a formula) or not. Further we can mechanically recover the term (or the formula) of which n is the Gödel number.

We have

The unary predicate $LAx_T \subset \mathbb{N}$ consisting of Gödel numbers of all logical axioms of T is recursive.

This result is true in more generality. But if T has only finitely many non-logical symbols, the proof becomes easier. We call a theory T *axiomatized* if the set $NAx_T \subset$

We have associated a number (called *Gödel number*) to each term and each formula of T so that there is an algorithm which on any input n, decides whether it is the Gödel number of a term (or a formula) or not. Further we can mechanically recover the term (or the formula) of which n is the Gödel number.

$\mathbb{N}$ of Gödel numbers of all non-logical axioms of T is recursive. This easily implies that the set $Ax_T \subset \mathbb{N}$ of Gödel numbers of all logical and non-logical axioms of T is also recursive. Thus, if T is axiomatized, there is an algorithm to decide whether a formula is an axiom (logical or non-logical) of T or not.

Example. Since the Theory N has only finitely many axioms, it is axiomatized. This is because every finite set of natural numbers is recursive. The Peano arithmetic and ZF can be shown to be axiomatized.

We continue the idea further. If $A_1, \cdots, A_n$ is a sequence of formulae, the number

$$\langle \lceil A_1 \rceil, \cdots, \lceil A_n \rceil \rangle,$$

is called the Gödel number of the sequence. Then we have

If T is axiomatized, the set Pr_T of Gödel numbers of all proofs in T is recursive.

We can also show that

If T is axiomatized, the set $Prf_T \subset \mathbb{N} \times \mathbb{N}$ of all pairs of numbers (m, n) such that m is the Gödel number of a proof of a formula whose Gödel number is n is recursive.

Since a natural number n is the Gödel number of a theorem of T if and only if there is a natural number m such that $(m, n) \in Prf_T$, the set of Gödel numbers of theorems of T is the projection of a recursive set in $\mathbb{N} \times \mathbb{N}$.

Thus, we have the following result.

If T is axiomatized, then the set Thm_T of Gödel numbers of all theorems of T is semi-recursive.

It can be shown that a formula A is a theorem if and only if its closure is a theorem. Further, assume that the theory T is complete. Then, if a sentence A is not a theorem of T, $\neg A$ is a theorem of T. Now there is a recursive function $g : \mathbb{N} \to \mathbb{N}$ such that if n is the Gödel number of a formula, $g(n)$ is the Gödel number of its closure; also, there is a recursive function $f : \mathbb{N} \to \mathbb{N}$ such that whenever m is the Gödel number of a formula A, $f(m)$ is the Gödel number of $\neg A$: Simply take

$$f(m) = \langle SN(\neg), m\rangle.$$

From these observations, it easily follows that $\mathbb{N} \setminus Thm_T$ is semi-recursive.

Theorem. *If T is an axiomatized, complete theory, then Thm_T is recursive.*

Proof. We have already shown that $Thm_T \subset \mathbb{N}$ is r.e.. Our proof will be complete, if we show that $\mathbb{N} \setminus Thm_T$ is also r.e. This follows from the following equivalence: For any $n \in \mathbb{N}$,

$$n \notin Thm_T \iff \neg For_T(n) \vee Thm_T(f(g(n))),$$

i.e., n is not the Gödel number of a theorem if and only if n is the Gödel number of a formula B implies that the closure of the negation B is a theorem of T. Since For_T is recursive, $\neg For_T$ is recursive. Hence, $\neg For_T$ is semi-recursive. By the closure properties of r.e. sets, it follows that $\mathbb{N} \setminus Thm_T$ is r.e..

This is a very important theorem. It says that if T is axiomatized and Thm_T is not recursive, then T is not complete. Gödel makes an elegant use of this result to establish his incompleteness theorem.

1.3 *Representability*

Now we present yet another beautiful concept, called *representability*, introduced by Gödel which shows that, in some sense, recursive functions and predicates can be represented by formulae of the theory N defined in Section 2 of Part 1 (see *Resonance*, February 2007).

Definition. We say that $P \subset \mathbb{N}^p$ is *representable in* N if there is a formula $A[v_1, \cdots, v_p]$ of N, with free variables $v_1, \cdots, v_p$ distinct, such that for every sequence of numbers $n_1, \cdots, n_p$,

$$(n_1, \cdots, n_p) \in P \Rightarrow N \vdash A_{v_1,\cdots,v_p}[k_{n_1}, \cdots, k_{n_p}],$$

and

$$(n_1, \cdots, n_p) \notin P \Rightarrow N \vdash \neg A_{v_1,\cdots,v_p}[k_{n_1}, \cdots, k_{n_p}].$$

We have

Theorem. *Every recursive predicate is representable in* N.

This is a remarkable idea. To make statements about numbers, first one develops a formal language (for instance, the language of N) and expresses statements about numbers syntactically in N and can be examined by N or a suitable extension of N; then using the idea of Gödel numbers, one expresses statements about syntactical objects using numbers themselves and, finally, using the notion of representability, one represents certain class S of statements about numbers, namely those S for which the set

$$\{n \in \mathbb{N} : n \text{ is the Gödel number of a sentence in } S\}$$

is recursive, by a formula of N. This technique enables one to hop into the theory from the meta-world and vice-versa. Thus many questions in the meta-world are expressed by a formula. Sometimes even a proof in the meta-world is converted into a proof inside the theory. This is how Gödel shows that *mathematics (= ZF or Peano arithmetic, say) can't prove its own consistency.*

This idea has also been quite precisely compared with the paintings of Escher and the music of Bach. (See [2].)

Gödel showed that mathematics (= ZF or Peano arithmetic, say) can't prove its own consistency. This idea has also been quite precisely compared with the paintings of Escher and the music of Bach.

2. Incompleteness Theorems

Theorem. *Every axiomatized, consistent extension of N is incomplete.*

Proof. Let T be an axiomatized, consistent extension of N. Since T is axiomatized, the set Thm_T of all Gödel numbers of theorems of T is semi-recursive. We have also observed that if T were complete, Thm_T would be recursive. So our proof will be complete if we show that Thm_T is not recursive. Towards arriving at a contradiction, assume that Thm_T is recursive. Fix a variable v. Now there is a recursive function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that if m is the Gödel number of a formula B of T, then $f(m, n)$ is the Gödel number of $B_v[k_n]$. Then the binary predicate $U \subset \mathbb{N} \times \mathbb{N}$ defined by

$$U(m, n) \Leftrightarrow Thm_T(f(m, n))$$

is recursive. So, the predicate

$$P(m) \Leftrightarrow \neg U(m, m)$$

is recursive.

By representability theorem, there is a formula A of N such that A with v represents P. This means that for every $n \in \mathbb{N}$,

$$n \in P \Rightarrow N \vdash A_v[k_n]$$

and

$$n \notin P \Rightarrow N \vdash \neg A_v[k_n].$$

Since T is an extension of N, A is a formula of T and for every $n \in \mathbb{N}$,

$$n \in P \Rightarrow T \vdash A_v[k_n] \quad (1)$$

and

$$n \notin P \Rightarrow T \vdash \neg A_v[k_n]. \quad (2)$$

Now let m be the the Gödel number of A.

The statement saying that "this formula is not true", is similar to the statement "I am lying" of the Liar's paradox.

Suppose $m \in P$. Then, by (1), $T \vdash A_v[k_m]$, i.e., $U(m, m)$ holds by the definition of U. Hence, $m \notin P$ by the definition of P. This is a contradiction.

On the other hand, suppose $m \notin P$. Then, by (2), $T \vdash \neg A_v[k_m]$. Since T is consistent, this implies that $T \nvdash A_v[k_m]$. Therefore, by the definition of U, $\neg U(m, m)$ holds. But then $m \in P$ by the definition of P. We have arrived at a contradiction again.

The proof of the theorem is complete.

Comparison with Liar's paradox. Since T is consistent, note that m is the Gödel number of the formula A with free variable v such that

$$P(m) \Leftrightarrow \neg Thm_T(f(m, m)).$$

So, m is the Gödel number of A such that $A_v[k_m]$ is not a theorem. In other words, the statement $\neg Thm_T(f(m, m))$ is saying that "this formula is not true", which is similar to the statement "I am lying" of the Liar's paradox.

This result can be easily generalized to

Theorem. (**First Incompleteness Theorem.**) *If T is a consistent, axiomatized theory such that N has an interpretation in an extension by definition of T, then T is incomplete.*

This result is true because T is essentially an extension by definition of N. However, the complete proof is quite a routine argument in logic. Since N has an interpretation in an extension by definition of ZF, it follows that, if ZF is inconsistent, it is incomplete.

Remark. Clearly in any axiomatization, the set of axioms should be decidable. Thus the above theorem says that there is absolutely no hope of axiomatizing most of the theories so that all the statements in the theory are decidable.

A very interesting question arises: *Is set theory consistent?* What does it mean? Note that just as there have to be initial concepts, initial theorems, etc., there has to be an initial theory too. Till now set theory is regarded as such a theory. So, another question arises: *Can set theory prove its own consistency?* In one of the greatest intellectual achievements, Gödel formulates this question precisely, and shows that *set theory can't prove its own consistency.* How does Gödel prove this? Using the concept of Gödel numbers and representability, we can represent the metamathematical statement '*ZF is consistent*' by a formula of ZF (actually of an extension of definition of ZF in which the Peano arithmetic has an interpretation).

We have noticed that the predicate $Prf_{ZF}(m, n)$ saying that m is the Gödel number of a proof in ZF of a formula whose Gödel number is n is recursive. So it is representable. We shall denote a formula representing it by $Prf_{ZF}(x, y)$ itself. Now denote the formula $\exists x Prf_{ZF}(x, y)$ by $Thm_{ZF}(y)$. This means that 'y is a theorem of ZF'. We have observed that a theory is inconsistent if and only if every formula is a theorem. We have also observed that the predicate $For_{ZF}(n)$ consisting of Gödel numbers of all formulae of ZF is recursive. So it is representable by a formula of ZF. We shall denote a formula that represents it by $For_{ZF}(y)$ itself. Let Con_{ZF} abbreviate the formula

$$\neg\forall y(For_{ZF}(y) \Rightarrow Thm_{ZF}(y)).$$

It says that not all formule y of ZF is a theorem of ZF, i.e., ZF is consistent.

Gödel shows:

Theorem. (**Second Incompleteness Theorem.**) *Con_{ZF} is not a theorem of ZF. That is, set theory can't prove its own consistency.*

Indeed such a result is true for all those axiomatized theory T such that the Peano arithmetic has an interpretation in an extension by definition of T.

The second incompleteness theorem is proved essentially by translating the proof of the first incompleteness theorem inside ZF. This is not surprising because our informal meta-world certainly 'contains' ZF. Often we have used induction in our meta-world. But since the Peano arithmetic has an interpretation in an extension by definition of ZF, arguments involving induction can also be formalized inside ZF. The rest is taken care of by the Representability theorem.

Address for Correspondence
S M Srivastava
Stat-Math Unit
Indian Statistical Institute
203, BT Road
Kolkata 700 108, India.
Email:smohan@isical.ac.in

Suggested Reading

[1] Martin Davis, Elaine Weyuker and Ron Segal, *Computability, Complexity and Languages: Fundamentals of Computer Science*, Academic Press, 1994.

[2] Douglas R Hofstadter, *Gödel, Escher, Bach: An Eternal Golden Braid*, Vintage Books, 1989.

[3] K Kunnen, *Set Theory – An Introduction to Independence Proofs*, North-Holland, 1980.

[4] J Shoenfield, *Mathematical Logic*, Addison-Wesley, 1967.