GÖDEL'S INCOMPLETENESS THEOREM

RYAN MORENO

1 Background

1.1 Overview

Gödel's Incompleteness Theorem proves that every consistent mathematic system has statements which are true but cannot be proven within the system. It is also shown that the consistency of any mathematic system cannot be proven within that system. In order to understand the nuances of Gödel's proof, it is best to have the outline in mind.

First, Gödel creates a one-to-one mapping between natural numbers and mathematical expressions, providing a unique identifier for each mathematical expression. It is important to note that the identifier for each mathematical expression is an integer that mathematical expressions can act on. A key point in Gödel's paper is the ability to mirror statements about mathematical expressions (such as "the sequence of formulas $F_1, F_2 \dots F_{n-1}$ is a proof schema for the formula F_n ") into the arithmetic system.

The heart of Gödel's argument is defining a formula G that states "formula G is not provable." It is shown that G is provable iff $\neg G$ is provable. Of course, if both G and $\neg G$ were provable, this would imply the inconsistency of the arithmetic system. So, if the arithmetic system is consistent, then G cannot be proven true or false, meaning that G is undecidable. Despite being unprovable within the arithmetic system, G is true. This follows directly from the definition of G, which states "formula G is not provable." This means that if the arithmetic system is consistent, then there is a true statement which cannot be proven within the system. Finally, Gödel shows that the statement "the arithmetic system is consistent" is not provable within the arithmetic system.

Gödel extends his findings to every mathematic system, proving that every consistent mathematic system has true statements which cannot be proven within the system and that the consistency of the mathematic system is unprovable.

1.2 Definitions

1.2.1 Formal Axiomatic Systems

A formal axiomatic system is a formally defined set of derivable theorems. The system consists of a set of basic signs, a set of axioms, and a set of rules of inference. The basic signs make up the alphabet of the system. The set of axioms are the statements which are assumed to be universally true. There can be infinite axioms if a schema is used, such as "Every formula derived by substituting any formula for p in the following: $p \lor p \implies p$." The rules of inference are the operations that manipulate axioms and theorems to create new theorems. The rules of inference are assumed to be truth-preserving. In this way, the formal axiomatic system consists of the axioms and all of the theorems derivable using the rules of inference.

1.2.2 Incompleteness

A formal axiomatic system is said to be complete if every true statement definable in the language of the system is a theorem of the system. This means that every true statement can be derived from the axioms of the system by the rules of inference. An incomplete system is one in which there is a true statement that can be defined in the language of the system but is not a theorem of the system, meaning that the statement cannot be derived from the axioms, and therefore cannot be proven within the system.

1.2.3 ω -consistency

A simply consistent system is one in which there are no explicit contradictions. This means that there is no formula H for which both H and \neg H can be proven within the system. Simple consistency ignores the logical contradiction derived through induction if $\forall x$ (the formula H(x) is provable) and the formula $\neg(\forall x H(x))$ is provable. If this type of contradiction is present in a system, it is ω -inconsistent. An ω -consistent system is a simply consistent system that is not ω -inconsistent.

1.2.4 Metamathematics

Mathematical statements are ones that can be written in a mathematic system. Some simple examples are F_1 : $x = 1, F_2$: y = x + 1, and F_3 : F = 2. Metamathematical statements are English statements about these mathematical statements. For example, the following statement is metamathematical, "the sequence of formulas F_1, F_2 is a proof schema for the formula F_3 ." A useful analogy is that mathematical statements are like the positions of chess pieces on a board while metamathematical statements are claims based on inferences from the current positions, such as the number of opening moves in a chess game or which player is currently check-mated. Note that these inferences are made by assigning a specific meaning to the chess positions that aren't defined within the positions themselves. Likewise, metamathematical statements consider a string and assign it meaning by interpreting it as a mathematical formula.

1.2.5 Primitive Recursive Formulas

Primitive recursive formulas are formulas that are exclusively defined by simple arithmetical relationships, logical relationships, and other primitive recursive formulas. If there is a search involved in the formula, it must be bounded. The following is an example of a primitive recursive formula.

$$\operatorname{Div}(x, y) \equiv \exists z [(z \le x) \land (x = y \cdot z)]$$

Div(x, y) is a primitive recursive formula because it is only defined by simple arithmetical relationships (·) and logical relationships (=, \land , and \leq). Additionally, the search ($\exists z$) is bounded by value of x.

Primitive recursive formulas are crucial to Gödel's proof. The arithmetic system P (defined in the next section) contains the basic arithmetical and logical relationships within its axioms. So, as long as a formula can be defined only in these simple relationships, it has a corresponding arithmetic function that can be defined within P. By their recursive definition, primitive recursive formulas can be derived from the basic arithmetical and logical relationships in a finite number of steps. Additionally, the rules of inference of P contain the logical relationships necessary to derive new formulas. This means that all primitive recursive formulas have a corresponding arithmetic function that is definable within P. Further, because every primitive recursive formula can be defined within P in a finite series of steps, these same steps can be used to evaluate the corresponding arithmetic function for a given input, serving to prove the formula either true or false for a specific input. Therefore, not only does every primitive recursive formula have a corresponding arithmetic function that is definable within P, but also, when its variables are given specific values, it is either provably true or provably false.

2 The Arithmetic System P

Gödel begins his paper by defining a specific arithmetic system he calls P, which is heavily based on Principia Mathematica, a simple but sufficient arithmetic system. The system P also uses the Peano Axioms, widely accepted as the formal definition of natural numbers.

2.1 Basic Signs of P

P uses only 7 primitive elements in its alphabet: $0, \neg, \lor, \forall, (,)$, and f. These all have their usual meaning, with f being the successor function, meaning one more than its argument. In this way, f and 0 are used together to define the natural numbers 0, 1, 2, 3... as 0, f0, ff0, fff0... Note that other logical symbols, such as $\land, \forall, \Longrightarrow$, and = can be defined in terms of these primitive elements. For clarity, the other logical symbols are employed directly.

The other basic signs are variables, represented by $v_1, v_2, v_3 \dots$

2.2 Axioms ofP

Gödel sets forth the axioms of p, a subset of which are presented here. First, he directly uses several of the Peano Axioms to define the natural numbers:

• $\neg(fv_1=0)$

There are no strings for which the successor is 0, meaning that there are no negative numbers.

• $fv_1 = fv_2 \implies v_1 = v_2$

If the successor of two numbers are equal, so are the two numbers.

• $[\operatorname{contains}(v_1, 0) \land \forall v_2(\operatorname{contains}(v_1, v_2) \Longrightarrow \operatorname{contains}(v_1, fv_2))] \Longrightarrow \forall v_2(\operatorname{contains}(v_1, v_2))$

This is the principle of induction. If v_1 contains 0 and v_1 containing an integer implies v_1 contains the next integer, then v_1 contains all natural numbers.

The axioms also include a set of logical, truth-preserving statements such as $p \wedge p \implies p$. Further, the axioms include the schemas necessary to turn these statements into infinite axioms by substitution, in this case substituting any axiom or theorem for p.

2.3 Rules of Inference of P

There are two rules of inference in P. The class of provable formula is the class containing all of the axioms of P and all formulas derivable through these rules of inference.

- The formula c can be derived from the formulas a and b if $a = (\neg b \lor c)$.
- The formula $\forall v_i(a)$, where v_i is any variable, can be derived from the formula a.

3 Gödel Numbering

Gödel creates a one-to-one mapping between natural numbers and mathematical expressions. This way, mathematical expressions can be identified by a unique integer that mathematical expressions can act on. Because the mapping is one-to-one, there exists a function $\phi(s) = g$ that takes a mathematical expression (a string in the language P) and returns its Gödel number. Likewise, there exists a function $\phi'(g) = s$ that takes a Gödel number and, using prime factorization, returns the mathematical expression it corresponds to.

In Gödel's proof, the distinction between a formula and its Gödel number is important because the formula is an object of the metamathematical realm while the Gödel number is an object within P. To make the distinction clear, the Gödel number of a formula F will be denoted \mathbf{F} .

3.1 **Primitive Elements**

The primitive elements are assigned to integers less than 14 as follows:

'0'	1	'∀'	9
ʻf'	3	"('	11
·'	5	')'	13
'∨'	7		

3.2 Variables

Variables are assigned prime numbers greater than 13. For example $v_1 = 17$, $v_2 = 19$, $v_3 = 23$ These assignments are all unique and do not share any prime factors.

3.3 Formulas

A formula in P is a finite series of basic signs. By translating each basic sign into its Gödel number, as previously described, a formula can be represented by a finite series of natural numbers. In turn, this finite series of natural numbers can be mapped to a single natural number. Given this finite series of natural numbers $n_1, n_2, n_3, \ldots n_k$, we raise the *i*th prime number to the power of the *i*th number in the series and multiply the results together, getting $2^{n_1} \cdot 3^{n_2} \cdot 5^{n_3} \ldots p_k^{n_k}$.

These assignments are all unique. The Gödel number of a formula can be distinguished from that of a basic sign because it will have multiple prime factors whereas a basic sign has only one prime factor. A formula is also distinguishable from other formulas because the index of each basic sign in a formula is uniquely identified by the prime number it is assigned within the value $p_i^{n_i}$. Further, which basic sign exists at a given index is uniquely identifiable by its Gödel number.

3.4 Sequences of Formulas

It is crucial to Gödel's proof that sequences of formulas can be assigned a Gödel number. This way, proof schemas can be assigned a Gödel number and be referenced within P. The approach to mapping a sequence of formulas to a natural number is the same as mapping a sequence of basic signs within a formula to a natural number. By translating each formula into its Gödel number, as previously described, a sequence of formulas can be represented by a finite series of natural numbers. Given this finite series of natural numbers $n_1, n_2, n_3, \ldots n_k$, we raise the *i*th prime number to the power of the *i*th number in the series and multiply the results together, getting $2^{n_1} \cdot 3^{n_2} \cdot 5^{n_3} \ldots p_k^{n_k}$. Later on, this series will be referred to as the series of numbers that make up this Gödel number, and an individual $p_i^{n_i}$ will be referred to as an element of the series of numbers that make up the Gödel number.

For the same reasons as above, the Gödel number of a given sequence of formulas can be distinguished from that of any other sequence of formulas. Further, a series of formulas is distinguishable from a formula, because when we factor out one of the prime numbers p_i to get a specific element of the sequence $p_i^{n_i}$, n_i will have multiple prime factors. This is unlike a formula, in which n_i represents a basic sign and only has one prime factor.

4 The Arithmetization of Metamathematics

A key point in Gödel's paper is the ability to mirror metamathematics in P. By translating mathematical expressions into their Gödel numbers and translating their metamathematical relations into arithmetic relations, metamathematical statements can be translated into P. The following is an example of a simple metamathematical statement translated into an arithmetic statement. Div(x, y) is the metamathematical statement "x is divisible by y."

$$\operatorname{Div}(x,y) \equiv \exists z [(z \le x) \land (x = y \cdot z)] \tag{1}$$

This arithmetic statement can be expressed in P because it is a primitive recursive formula, consisting only of basic arithmetical and logical relationships. Further, since this arithmetic statement can be expressed as a formula in P, it can be assigned a unique Gödel number.

In order to describe metamathematical statements between two formulas, there needs to be a corresponding arithmetical relation between their two Gödel number's. The following is an example of an arithmetical relation that assists in breaking down a sequence of formulas into their individual formulas, or a sequence of basic signs into their individual basic signs. The formula Pr(n, x) is the *n*th prime factor, in order of magnitude, of the number *x*. Because the Gödel numbering system is dependent on prime factorization, Pr(n, x) is useful in interpreting a Gödel number.

$$\Pr(0, x) \equiv 0$$

$$\Pr(n+1, x) \equiv \epsilon y[(y \le x) \land (\Pr(y)) \land (\operatorname{Div}(x, y)) \land (y > \Pr(n, x))]$$
(2)

The formula $\epsilon y \operatorname{H}(y)$ represents the smallest number for which $\operatorname{H}(y)$ holds and 0 if there is no such number. $\operatorname{Prim}(y) \equiv y$ is a prime number. This is also shown to be primitive recursive. Both formulas are shown to be primitive recursive. The formula $\operatorname{Pr}(n, x)$ asserts that y is the smallest prime factor of x that is larger than the the nth prime factor of x. The initial assertion $y \leq x$ bounds the search for y. Given the base case that the 0th prime factor of x is 0, the formula uses recursion to find the n + 1th prime factor of x. Because this formula is primitive recursive in form and only employs other relations that are primitive recursive, $\operatorname{Pr}(n, x)$ is expressible in P.

Gödel's paper provides 46 such formulas representing metamathematical statements, each of which is primitive recursive. Thus, these metamathematical statements can be defined within P. The final two formulas Gödel presents are key to his proof of incompleteness. $Prov(\mathbf{x}, \mathbf{y})$ means that x is a sequence of formulas that proves the formula y. The formal definition requires several other functions that Gödel provides. $Ps(\mathbf{x})$ means that x is a proof-schema, a finite series of formulas such that each formula is either an axiom or deducible from previous formulas using the rules of inference of P. $L(\mathbf{x})$ is the length of the series of numbers that make up the Gödel number \mathbf{x} . $Nel(n, \mathbf{x})$ is the nth element of the series of numbers that make up the Gödel number x.

$$\operatorname{Prov}(\mathbf{x}, \mathbf{y}) \equiv (\operatorname{Ps}(\mathbf{x})) \land (\operatorname{Nel}(\operatorname{L}(\mathbf{x}), \mathbf{x})) = \mathbf{y}$$
(3)

 $Prov(\mathbf{x}, \mathbf{y})$ should require that x is a proof-schema and that the final formula in that proof-schema is y. Equation (3) asserts exactly that. The second half of the equation states that the last element in the series of numbers that make up \mathbf{x} is \mathbf{y} . Because the x is a proof schema, the last element in the series of numbers that make up \mathbf{x} is the Gödel number for the formula that x is supposed to prove.

The final equation that Gödel provides is $Provable(\mathbf{y})$ which asserts that y is a formula which is provable in P. This follows directly from $Prov(\mathbf{x}, \mathbf{y})$. It turns out that $Provable(\mathbf{y})$ is not primitive recursive because the search $\exists x$ is

unbounded. However, this is the only formula that Gödel's proof does not require to be primitive recursive, so this is not a problem.

$$Provable(\mathbf{y}) \equiv \exists \mathbf{x} Prov(\mathbf{x}, \mathbf{y})$$
(4)

In the next section, it will also be important to know that Gödel provides a primitive recursive definition of substitution. $Sub(\mathbf{x}, [v_1 : \mathbf{y_1}, v_2 : \mathbf{y_2}, ...v_n : \mathbf{y_n}])$ returns the Gödel number for the formula derived by taking x and substituting every occurrence of the variable v_i with the Gödel number $\mathbf{y_i}$. For simplicity, Sub is not explicitly referred to. However, it is important that the process of substitution within primitive recursive formulas yields primitive recursive formulas.

5 Gödel's Proof

5.1 P is Incomplete

After defining the equations necessary, Gödel shows that P is able to deduce a formula H(x) for which it cannot be proven $\forall x H(x)$ nor $\neg \forall x H(x)$. Thus, $U = \forall x H(x)$ is an undecidable formula within this system. This proof is detailed below.

Variables in P are each assigned a specific prime number. In this way, two formulas can refer to the same variable, which can be important when a formula is substituted into another formula. For example, consider the formula $F = \forall v_i[G]$ where G is a variable for a formula. If G is replaced by a formula with v_k as a variable, this is very different from G being replaced by a formula with v_i as a variable, since F quantifies G with v_i . For clarity, throughout the proof, the variables that are shared between formulas are referred to as v_1 and v_2 , while the variables that are not shared are referred to as a and b.

First, Gödel defines a new formula $Nprov(\mathbf{a}, \mathbf{b})$.

$$Nprov(\mathbf{a}, \mathbf{b}) \equiv \neg Prov(\mathbf{a}, \mathbf{b}\{\mathbf{v}_2 : \mathbf{b}\})$$
(5)

Nprov(\mathbf{a}, \mathbf{b}) asserts that the series of formulas *a* does not prove the formula derived by substituting every occurence of the variable v_2 in *b* with \mathbf{b} . Note that $b\{v_2 : \mathbf{b}\}$ is self-referential. Because Nprov(\mathbf{a}, \mathbf{b}) is a primitive recursive function, if the formula holds for specific values, it can be proven within P that it holds for those values.

$$Nprov(\mathbf{a}, \mathbf{b}) \implies Provable(Nprov(\mathbf{a}, \mathbf{b}))$$
 (6)

Similarly, it can be asserted that if $Nprov(\mathbf{a}, \mathbf{b})$ doesn't hold for specific values, it can be proven within P that it doesn't hold for those values.

$$\neg \operatorname{Nprov}(\mathbf{a}, \mathbf{b}) \Longrightarrow \operatorname{Provable}(\neg \operatorname{Nprov}(\mathbf{a}, \mathbf{b}))$$
 (7)

Gödel then presents two more formulas. At this point in the proof, it becomes difficult to translate the formulas into English, so it is best to consider them in a formal, abstract sense. $Xform(v_2)$ is the statement that no matter what is substituted for a, $Nprov(a, v_2)$ holds.

$$Xform(v_2) \equiv \forall v_1 N prov(v_1, v_2)$$
(8)

Now consider the Gödel number for the formula $X form(v_2)$, $X form(v_2)$. $Y form(v_1)$ is defined as the formula derived from Nprov(\mathbf{a}, \mathbf{b}) where v_1 is substituted for a and $\mathbf{Xform}(\mathbf{v_2})$ is substituted for b.

$$Yform(v_1) \equiv Nprov(v_1, Xform(v_2))$$
(9)

Next Gödel substitutes \mathbf{X} form (\mathbf{v}_2) into the formula X form (v_2) . The work is shown below.

$$\begin{aligned} \operatorname{Xform}(\mathbf{Xform}(\mathbf{v_2})) &= \forall v_1 \operatorname{Nprov}(v_1, \mathbf{Xform}(\mathbf{v_2})) \\ &= \forall v_1 \operatorname{Yform}(v_1) \qquad \text{by (9)} \end{aligned} \tag{10}$$

Finally, Gödel returns to equation 6.

$$Nprov(\mathbf{a}, \mathbf{b}) \implies Provable[\mathbf{Nprov}(\mathbf{a}, \mathbf{b})]$$

$$\neg Prov[\mathbf{a}, \mathbf{b}\{\mathbf{v}_{2}: \mathbf{b}\}] \implies Provable[\mathbf{Nprov}(\mathbf{a}, \mathbf{b})] \qquad by (5)$$

$$\neg Prov[\mathbf{a}, \mathbf{X} form(\mathbf{X} form(\mathbf{v}_{2}))] \implies Provable[\mathbf{Nprov}(\mathbf{a}, \mathbf{X} form(\mathbf{v}_{2}))] \qquad sub. \mathbf{b} = \mathbf{X} form(\mathbf{v}_{2})$$

$$\neg Prov[\mathbf{a}, \forall \mathbf{v}_{1} \mathbf{Y} form(\mathbf{v}_{1})] \implies Provable[\mathbf{Y} form(\mathbf{a})] \qquad by (10), (9)$$

$$(11)$$

The same steps apply to equation 7, resulting in the following.

$$\operatorname{Prov}[\mathbf{a}, \forall \mathbf{v_1} \mathbf{Y} \mathbf{form}(\mathbf{v_1})] \implies \operatorname{Provable}[\neg \mathbf{Y} \mathbf{form}(\mathbf{a})]$$
(12)

Consider the following equation.

$$\mathbf{U} \equiv \forall v_1 \mathrm{Yform}(v_1) \tag{13}$$

U is unprovable. For the sake of contradiction, let's assume it is provable. Then there would exist some series of formulas *a* that proves it, meaning $Prov(\mathbf{a}, \forall \mathbf{v_1} \mathbf{Y} \mathbf{form}(\mathbf{v_1}))$ would hold. By (12), this would imply $Provable(\neg \mathbf{Y} \mathbf{form}(\mathbf{a}))$, meaning that there is an a such that it's provable that \neg Yform(a). But we assumed that $\forall v_1$ Yform(v_1) is provable, meaning that Y form(a) is provable! Thus, the assumption that U is provable leads to an inconsistency.

It is also the case that $\neg U$ is unprovable. As was just shown, $\forall v_1 Y \text{form}(v_1)$ is unprovable. This means that there is some series of formulas a that doesn't prove it, meaning $\neg Prov(\mathbf{a}, \forall \mathbf{v_1} \mathbf{Y} \mathbf{form}(\mathbf{v_1}))$. By (11), this implies Provable(**Yform**(**a**)), meaning that there is an *a* such that it's provable that $Yform(\mathbf{a})$. If $\neg \forall v_1 Yform(v_1)$ was provable, this would cause an ω -inconsistency because we know there is an a such that it is provable that Yform(a). So, $\neg U$ is also unprovable.

If P is ω -consistent, then both U and $\neg U$ are unprovable. Thus, U is an undecidable formula if P is ω -consistent.

5.2 U is an Unprovable but True Formula

Although U is undecidable within P, from the metamathematical realm we can see that U is true.

г

$$U \equiv \forall v_1 \text{Yform}(v_1) \qquad \qquad \text{by (13)}$$

$$= \forall v_1 \operatorname{Nprov}[v_1, \mathbf{Xform}(\mathbf{v_2})] \qquad \text{by (9)}$$

$$= \forall v_1 \operatorname{Nprov}[v_1, \forall \mathbf{v_1} \operatorname{Nprov}(\mathbf{v_1}, \mathbf{v_2})] \quad \text{by (8)}$$

>1

(14)

From the perspective of P, The equation $U \equiv \forall v_1 \operatorname{Nprov}[\mathbf{v_1}, \forall \mathbf{v_1} \operatorname{Nprov}(\mathbf{v_1}, \mathbf{v_2})]$ is a meaningless string. However, from the metamathematical perspective, there is an attributed meaning. Remember that $\operatorname{Nprov}(\mathbf{a}, \mathbf{b})$ means the series of formulas *a* does not prove the formula derived by substituting every occurence of the variable v_2 in *b* with **b**.

 $U \equiv \forall v_1 \operatorname{Nprov}[\mathbf{v_1}, \forall \mathbf{v_1} \operatorname{Nprov}(\mathbf{v_1}, \mathbf{v_2})]$ means that there is no proof of $\forall v_1 \operatorname{Nprov}(v_1, v_2)$ when it's plugged into its own v_2 , meaning there is no proof of $\forall v_1 \operatorname{Nprov}[\mathbf{v_1}, \forall \mathbf{v_1} \operatorname{Nprov}(\mathbf{v_1}, \mathbf{v_2})]$. This is the original statement U! So, U asserts its own unprovability. As shown in the previous section, U is in fact unprovable, meaning U is true. Thus, although it's impossible to prove U from within the system P, it can be shown from the metamathematical realm that U is true.

5.3 The Consistency of P is Unprovable

Using the undecidability of U, Gödel demonstrates that the consistency of P is unprovable. The previous proof of incompleteness showed that if P is ω -consistent, U is unprovable. This is represented in the first line of the following proof. The term Pconst means that P is ω -consistent.

$Pconst \implies \neg Provable[\mathbf{U}]$	
$\operatorname{Pconst} \implies \forall v_1 \neg \operatorname{Prov}[\mathbf{v_1}, \mathbf{U})]$	by (4)
$Pconst \implies \forall v_1 \neg Prov[\mathbf{v_1}, \forall \mathbf{v_1} Y \mathbf{form}(\mathbf{v_1})]$	by (13)
$Pconst \implies \forall v_1 \neg Prov[\mathbf{v_1}, \forall \mathbf{v_1} \mathbf{X} \mathbf{form}(\mathbf{X} \mathbf{form}(\mathbf{v_2}))]$	by (10)
$Pconst \implies \forall v_1 N prov[\mathbf{v_1}, \mathbf{Xform}(\mathbf{v_2})]$	by (5)
$Pconst \implies \forall v_1 N prov[\mathbf{v_1}, \forall \mathbf{v_1} \mathbf{N prov}(\mathbf{v_1}, \mathbf{v_2})]$	by (8)
$Pconst \implies U$	by (14)

Using only the equations expressible within P, this proves that if P is ω -consistent, then U is true. So, if it can be proven that P is ω -consistent within P, then U can be proven within P. However, we know from the previous proof of incompleteness that U is unprovable within P. So, it must be the case that the ω -consistency of P cannot be proven within P.

6 Extension to other Mathematical Systems

Gödel's proof of incompleteness only relies on primitive recursive formulas. Because of this, the same idea can be applied to any mathematic system for which all primitive recursive relations are definable. As was discussed in the background, despite its simplicity, P contains all primitive recursive relations. Because P is a very simple mathematic system that contains only the basic arithmetic of natural numbers, every useful mathematic system is at least as powerful as P. To be as powerful, these systems must contain the axioms and rules of inference of P, meaning that they can define all primitive recursive formulas and Gödel's findings translate to them. Therefore, Gödel proves that every nontrivial mathematic system is necessarily incomplete and cannot be proven ω -consistent within the system.

References

- C. Mingzhong, "The Hitchhiker?s Guide to the Incompleteness Theorem," Department of Mathematics, Cornell University, Jan 2009. [Online.] Available: http://pi.math.cornell.edu/ mec/2008-2009/MingzhongCai/hgit.pdf
- [2] E. Nagel and J. R. Newman, Gödel's Proof, Revised Edition. New York University Press, 2001.
- [3] J. R. Meyer, "A Step by Step Guide to Gödel's Incompleteness Proof," Logic and Language, Jan 2015. [Online.] Available: https://www.jamesrmeyer.com/ffgit/godel-guide-0.html
- [4] J. Steinmetz, "An Intuitively Complete Analysis of Gödel's Incompleteness," 2015. [Online.] Available: https://arxiv.org/pdf/1512.03667.pdf
- [5] K. Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I," Monatshefte für Mathematik Physik, Vol. 38: 173-198, 1931.
- [6] P. Raatikainen, "Gödel's Incompleteness Theorems," Stanford Encyclopedia of Philosophy, Jan 2015. [Online.] Available: https://plato.stanford.edu/entries/goedel-incompleteness/