A Step by Step Guide to Gödel's Incompleteness Proof

Introduction

This is a step by step walk-through guide for anyone attempting to follow Gödel's original proof of incompleteness, the paper entitled "*On Formally Undecidable Propositions Of Principia Mathematica And Related Systems*" (see http://www.jamesrmeyer.com/ffgit/godel-original-english.html online for an English translation of Gödel's original proof).

This is a PDF copy of the online guide which is available at http://www.jamesrmeyer.com/ffgit/godel-guide-0.html.
It is recommended that this online version be used if possible, since it is interactive and will be the latest version; also the page breaks in the PDF copy are not ideal.

Despite the vast amount of material written about Gödel's proof, I have not seen any good guide which actually takes the reader step by step through Gödel's proof. There are a huge number of purported 'explanations' of Gödel's proof, which claim to explain the proof, but do not actually follow Gödel's actual proof, but instead use their own method. And while Gödel's proof is constantly referred to with veneration, and has been called an "*amazing intellectual symphony*",[1] the dearth of detailed analysis of the actual proof itself is all the more surprising. It is rather like telling musical students that Beethoven's last String Quartet in F major is a masterpiece, and using a rewritten version to point out some facets of it, but at the same time refusing to actually analyze the actual details of the work itself to try and establish what makes it a masterpiece.

In addition, other authors seem to treat Gödel with an overwhelming reverence and treat Gödel's paper as sacrosanct and above all criticism, so that anomalies in the proof are not mentioned, never mind questioned. While such anomalies will be noted, this will be done in order to assist the reader. In most cases the anomalies are inconsequential to the argument, but they can create real difficulties for someone reading the paper for the first time.

For this reason I decided to create a guide that leads the reader thorough the intricacies of Gödel's proof, with the aim of being the best possible such guide to Gödel's proof, which deals with the paper in a logical manner while at the same time explaining the line of argument of the proof. There is, of course, always room for improvement; if you have any suggestions or criticisms, contact me, they will be used to improve this guide. If there is any difficulty in following any part of the proof, please contact me and I will try to help.

It is only fair at this point to mention to the reader that this analysis reveals a fatal error in Gödel's argument. However, this analysis is presented in an objective manner, and there is no attempt to mislead or deceive the reader, nor any attempt to conceal any aspect of the proof or to misrepresent any part of it. The intention is that by following this analysis, the reader will attain a better understanding of Gödel's proof. I would note here that of all the incompleteness proofs I have examined, I much prefer Gödel's, as its line of argument is much more subtle. And one cannot help but be impressed by Gödel's virtuoso demonstration of how to translate meta-statements about formulas into number-theoretic relations about numbers. But a chain is only as strong as its weakest link, and if one is inspecting a chain to see if it is fit for purpose, one must look at every link, not just the shiny ones.

In this guide, we will not go into every single detail, but we will concentrate on those details that one needs to have a knowledge of in order to understand the line of argument in the proof. Once you understand how the proof works, then of course, you may want to look again at some aspects of the proof in more depth.

This guide is intended to be read alongside the English translation of Gödel's original proof which can be viewed online at http://www.jamesrmeyer.com/ffgit/godel-original-english.html or as a PDF at http://www.jamesrmeyer.com/pdfs/godel-original-english.pdf"[2]. Confusingly, in some versions of the same translation, negation is indicated by horizontal bars over the negated entity; here we will use the ~ symbol for negation, which is the same as in the translations linked to above. Note: you should be aware that there are some minor errors in Meltzer's translation which are not in the original German text.

In this guide, a certain amount of basic knowledge of mathematical concepts is assumed. If your knowledge of these basic concepts is limited, then perhaps you should consider reading the simplified explanation of Gödel's proof first.

Parts of the Proof

Gödel's proof can be broken down into the following principal parts:

Part 1: Gödel's introduction

This is Gödel's introductory part of his paper (Part 1 of his paper) rather than the proof itself.

Part 2: The definition of the formal system

Gödel's paper defines in precise detail a formal system; the goal of Gödel's proof is to prove that this formal system is incomplete.

Part 3: The axioms and rules of inference

Here Gödel defines the axioms and the rules of inference of the formal system.

Part 4: The Gödel numbering system

This is a method for assigning a unique number to every combination of symbols of the formal system - these numbers are sometimes called Gödel numbers. The proof is interested in certain relationships between certain combinations of symbols of the formal system, such as whether a certain string of symbols constitutes a proof of another string of symbols - in which case the first string is a proof, and the other is a valid formula of the system. The aim of the proof is to define relations between the Gödel numbers that precisely correspond to the relationships between the corresponding strings of symbols of the formal system - so that if the relationship between the symbol strings applies, then the corresponding relation between the corresponding Gödel numbers also holds. If this is done correctly, then the relations between the Gödel numbers mirror precisely the relationships between the corresponding symbol strings.

Part 5: The definition of primitive recursion

Gödel's reason for introducing primitive recursion is that any number-theoretic relation that can be shown to be primitive recursive is a relation which can always be proved either correct or incorrect by a simple mechanical procedure. This property is used as a key part of the proof sketch for Gödel's Proposition V.

Part 6: Gödel's definitions of functions and relations 1-23

Gödel defines 46 key relations/functions in the proof. This part deals with the first 23 of these.

Part 7: Gödel's definitions of functions and relations 24-46

This part deals with the remainder of the key relations/functions used in the proof

Part 8: Gödel's Proposition V

Using the previously defined relations, this proposition asserts that a certain relationship exists between every primitive recursive relation and a Gödel number that corresponds to that relation. The proposition includes the claim that propositions of the formal system can make logically valid propositions about propositions of the formal system *itself*, by way of Gödel numbering.

Gödel's Proposition VI

This is the assertion that there is a certain formula of the formal system **P** that the formal system cannot prove to be correct or incorrect, and yet it is a statement that must be either 'true' or 'false'. This guide does not deal with this proposition nor any part of the paper past this point.

---

[1] This accolade may be found in the section '*The heart of Gödel's argument (v)*' in the book *Gödel's Proof* by E Nagel and J Newman. New York University Press, revised edition, 2001. ISBN: 0814758169

[2] Alternative versions of Meltzer's translation can also be found online, see http://www.geier.hu/GOEDEL/Godel_orig/godel3.htm. Other translations of "*Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme*" are available; one that is often referred to is one by Jean Van Heijenoort. It is not available online; it can be found in the book: *From Frege to Gödel: A Source Book in Mathematical Logic*, publisher: Harvard University Press. There are arguments as to which English translation is the 'best' and most faithful to the German original (it can be seen at https://metalab.at/wiki/images/0/0b/Goedel.pdf), but the reality is that the differences between this translation and Meltzer's are minor and inconsequential. Both of these use the same names for relations/functions as in Gödel's original paper. Another English translation by Martin Hirzel is available online (http://www.research.ibm.com/people/h/hirzel/papers/canon00-goedel.pdf), though this is not recommended for reading in conjunction with this guide, since in Meltzer's translation, van Heijenoort's translation, in the original German text and in this guide the names for the various relations and functions are all the same, but they are different in Hirzel's translation.

A Step by Step Guide to Gödel's Incompleteness Proof

Part 1 of Gödel's paper

This part of Gödel's paper is an introduction and is not intended to be completely rigorous, so it should not be read with such expectations, and should be read as expository material that lays out an overview of the paper.

In the paragraph beginning *"The development of mathematics…"*, Gödel sets out his claim that all formal systems are incomplete. Then, in the paragraph beginning *"Before going into details… "*, Gödel sets out the basic ideas underlying his proof; the principal ideas are as follows:

The definition of a formal system

A formal system consists of a definite set of symbols. The formal system includes definitions that define which combinations of the symbols are valid formulas of the system. A proof (or proof-schema, as Gödel puts it) in a formal system is simply a series of formulas, beginning with one or more axioms, where each formula in the series follows from one of more previous formulas by the rules of inference of the system - where the last formula is the formula that is proved by the proof-schema.

Note: Gödel refers to a formal system **PM** - this refers to Russell's system 'Principia Mathematica'.[1] The formal system **P** that Gödel actually uses in his proof includes typed classes in a similar fashion to that in 'Principia Mathematica' together with the Peano axioms.[2]

The mapping of the formulas of a formal system to numbers

The symbols of the formal system are mapped to natural numbers, so that for every symbol of the formal system, there is a corresponding unique number. Using this mapping, then a formula of the formal system becomes mapped to a series of natural numbers, and a proof-schema becomes mapped to a finite series of series of natural numbers. And although Gödel does not state it here, in fact, his proof includes a method by which each such series of numbers is transformed into a single number, so that for each formula, and for each proof/proof-schema, there can be a unique corresponding natural number.

Mapping of relationships between formulas to relationships between numbers

Since relationships between formulas of the formal system, or between proof-schemas of the formal system, or between formulas and proof-schemas of the formal system, are all precisely defined notions, there can be corresponding relationships between the natural numbers that correspond to the formulas and proof-schemas of the formal system. Furthermore, if the correspondence is defined correctly, then if any such relationship between formulas/proof-schemas applies, then the corresponding relationship between natural numbers also applies. Also the inverse - if any relationship between natural numbers applies, then the corresponding relationship between formulas/proof-schemas also applies (note that if a natural number does not correspond to some symbol string of the formal system, then there can be no corresponding relationship, since there is not a corresponding formula/proof-schema).

Outline of a proof

The rest of the Part 1 of Gödel's paper from the point *"We now obtain an undecidable proposition…"* is an outline description of a proof. But it is ***not*** an outline description of the method Gödel actually used in his proof, but of a different method. This has led to confusion for many people, because they are expecting the proof proper to follow that outline. For this reason, the reader is advised to ignore the rest of the introduction until one is fully cognizant with the actual proof that Gödel sets out in Part 2 of his paper.

---

[1] Russell, Bertrand, and Alfred North Whitehead, "*Principia Mathematica*"; 3 Volumes, 1910, 1912, and 1913, Cambridge University Press.
[2] The Peano axioms were formulated by the Italian mathematician Giuseppe Peano. They constitute a formal definition of the fundamental properties of natural numbers. See for example http://mathworld.wolfram.com/PeanosAxioms.html.

A Step by Step Guide to Gödel’s Incompleteness Proof

The definition of the formal system

To understand the guide from this point on, you will need to already have a good understanding of the fundamentals of what is called propositional logic and predicate logic (including the meaning of free variables and bound variables). A fairly good online guide that covers the basics can be viewed at http://www.cs.odu.edu/~toida/nerzic/level-a/web_course.html. Some information can also be found on the website http://www.jamesrmeyer.com.

> **NB**: It is important to bear in mind that, in Meltzer’s translation, the term ‘***sign***’ is used to refer (in general) to strings of symbols of the formal system. The proof also uses some distinctive terms (number-signs, type-n signs, relation-signs, class-signs, see below) to refer to certain kinds of strings of symbols.
> But note that a single symbol is *also* referred to as a ‘***sign***’; it is a string with just one symbol. The term ‘***basic sign***’ is used to refer to either a single symbol or a string of symbols that represents a variable, see below. When you are reading the proof you need to be familiar with this terminology.

When reading Gödel’s proof, one should remember that Gödel is a proof of incompleteness for one specific formal system which Gödel defines in detail and which he calls the system **P**. Bear in mind that Gödel’s method, which he applies to this one particular system **P**, can in principle be applied to any formal system (provided it contains a certain amount of basic arithmetic, such as defining natural numbers and basic operations on those numbers).

From that point of view, you might say that the precise details of the formal system that Gödel uses in his proof are not of overriding importance, since Gödel’s argument could be applied to any formal system. But, having said that, if you don’t spend the time to become familiar with the terminology of the details below (and the corresponding parts of Gödel’s paper), you may have difficulty following the subsequent argument.

Note that in the following, green text is a direct excerpt of Gödel's text.

Definition of the formal system P

In the paragraph beginning *"We proceed now to the rigorous development…"*. Here Gödel sets out the definition of the formal system he will use for his proof. He calls this the system **P**.

The symbols of the formal system P

Every formula of the system **P** is constructed using seven single symbols along with symbols for variables (see below). The seven symbols are:

> "The basic signs of the system P are the following:
> I. Constants: "~" (not), "∨" (or), "∀" (for all), "**0**" (nought), "***f***" (the successor of), "**(**", "**)**" (brackets)."

0 and ***f*** are used to represent the natural numbers 0, 1, 2, 3, … as **0**, ***f*0**, ***ff*0**, ***fff*0**, **…**

~ is used to represent 'not' (note: in modern notation this would be ¬)

∨ is used to represent 'or'

∀ is used to represent 'for all' (note: where **a** is some formula, **v∀(a)** represents *'for all* ***v, a****'*; whereas the common equivalent modern notation is **∀v (a)**)

'(' and **')'** are used as opening and closing brackets.

Note that the above symbols of the formal system **P** are sometimes referred to as ***'basic signs'*** (the variables of the system are also referred by the same terminology, so you should be aware of this).

Classes in the formal system P

The other symbols that the formal system **P** uses are for variables. Before we consider the symbols used for variables, we need to know that the formal system **P** is defined as including a hierarchy of ***'classes'***, where the lowest class is a collection of natural numbers, the next level of class is a class that contains classes of classes of natural numbers, and so on. The different levels of classes are referred to as different ***types*** of classes, and the lowest class is a type 1 class. These details of how the formal system manipulates classes, are for the most part peripheral to the main argument of the proof, so when trying to understand how the proof operates, one should not get bogged down by these details. However, you do need to know what Gödel is referring to when he uses certain terms.

Note that in the system **P** brackets can have two purposes

a. to separate parts of an expression, and
b. to indicate membership of a class; for example, **a(b)** means ***'b*** *is a member of the class* ***a'***.

Type 1 class:

The '**type 1**' class (also called '**first type**') refers to classes that contain natural numbers (symbol strings such as **0**, ***f*0**, ***ff*0**, ***fff*0**, **…**), or variables for natural numbers. For the detailed definition of what this class contains see Signs of type 1 below.

Type 2 class:
The ‘**type 2**’ class (also called ‘second type’) refers to classes that contain **type 1** classes.

Type 3 class:
The ‘**type 3**’ (also called ‘third type’) refers to classes that contain **type 2** classes.

And so on, giving, in general,

Type n class:
This class refers to classes that contain classes of the next lower **type** (that is, classes whose type is **n - 1**) - and these in turn contain classes of the next lower **type** (that is, classes whose type is **n - 2**) - and so on, until you reach the class that contains natural numbers.

We can now define the symbols that are used for the variables of the system **P**.

Variables in the formal system P

Variables of type 1 have the domain of symbol strings of the form **0**, ***f*0**, ***ff*0**, ***fff*0**, **…** .
Represented by $\mathbf{x_1}$, $\mathbf{y_1}$, $\mathbf{z_1}$, …
Note that **variables of type 1** are also called **type 1 variables** or **variables for natural numbers**.

“II. Variables of first type (for individuals, i.e. natural numbers including 0): ‘$\mathbf{x_1}$’, ‘$\mathbf{y_1}$’, ‘$\mathbf{z_1}$’,…
Variables of second type (for classes of individuals): ‘$\mathbf{x_2}$’, ‘$\mathbf{y_2}$’, ‘$\mathbf{z_2}$’,…
Variables of third type (for classes of classes of individuals): ‘$\mathbf{x_3}$’, ‘$\mathbf{y_3}$’, ‘$\mathbf{z_3}$’, … ”

Variables of type 2 have the domain of *classes that contain symbol strings of the form* **0**, ***f*0**, ***ff*0**, ***fff*0**, **…** .
Represented by $\mathbf{x_2}$, $\mathbf{y_2}$, $\mathbf{z_2}$, …

Variables of type 3 have the domain of *classes of classes of symbol strings of the form* **0**, ***f*0**, ***ff*0**, ***fff*0**, **…** .
Represented by $\mathbf{x_3}$, $\mathbf{y_3}$, $\mathbf{z_3}$, …

Variables of type n have the domain of *classes of classes of classes* … . (n-1 levels).
Represented by $\mathbf{x_n}$, $\mathbf{y_n}$, $\mathbf{z_n}$, … [1]

Note that variables in the formal system **P** are sometimes referred to as ***‘basic signs’*** (as are the single symbols of the system, see above).

Number-signs, Type-n signs, Relation-signs and Class-signs

number-sign

This is a string of symbols of the form:

0, ***f*0**, ***ff*0**, ***fff*0, …**

i.e., a number-sign is a string of symbols that represents a number.

sign of type 1 (also called **type 1 sign**, or a **sign of the first type**)

This is either a **number-sign** (as defined above) or a string of symbols of the form

$\mathbf{x_1}$, $\mathbf{\mathit{f}x_1}$, $\mathbf{\mathit{ff}x_1}$, $\mathbf{\mathit{fff}x_1}$, **…** or $\mathbf{y_1}$, $\mathbf{\mathit{f}y_1}$, $\mathbf{\mathit{ff}y_1}$, $\mathbf{\mathit{fff}y_1}$, **…**

where $\mathbf{x_1}$, $\mathbf{y_1}$, etc are variables for natural numbers.

i.e., A sign of type 1 is a string of symbols that can either represent a specific number, or a variable quantity.

> "By a sign of first type we understand a combination of signs of the form: **a**, ***f*a**, ***ffa***, ***fffa***, **…** etc where **a** is either **0** or a variable of first type. In the former case we call such a sign a **number-sign**."

sign of type n (also called **type n sign**, or a **sign of the $\mathbf{n^{th}}$ type**)

In Gödel's proof, a **sign of type n**, where **n > 1**, is the same as a **variable of type n**. Note that this means that in the system **P** there are no symbols that represent specific classes.

> "For **n>1** we understand by a **sign of n-th type** the same as a **variable of n-th type**."

n-place relation-sign

This is a formula with **n** different free variables (formulas are defined below). Note that the symbol for the *same* free variable may occur several times in a formula, but this does not determine the **n-place** of the formula. It is the number of *different* free variables that determines the **n-place** of the formula.

> "A formula with just **n** free individual variables (and otherwise no free variables) we call an **n-place relation-sign** and for **n=1** also a **class-sign**."

class-sign

A **class-sign** is a **1-place relation-sign**, that is, a formula with only one free variable. As indicated above, note that this same variable may occur several times in the formula.

Formulas of the formal system

Gödel defines what constitutes a formula in the formal system. This is required to ensure that a formula is a valid string of symbols of the formal system.

> "Combinations of signs of the form **a(b)**, where **b** is a sign of n-th and **a** a sign of (n+1)-th type, we call **elementary formulae**. The class of **formulae** we define as the smallest class containing all elementary formulae and, also, along with any **a** and **b** the following: **~(a)**, **(a) ∨ (b)**, **x∀(a)** (where **x** is any given variable). We term **(a) ∨ (b)** the **disjunction** of **a** and **b**, **~(a)** the **negation** and **x∀(a)** a generalization of **a**. A formula in which there is no free variable is called a **propositional formula** (**free variable** being defined in the usual way). A formula with just **n** free individual variables (and otherwise no free variables) we call an **n-place relation-sign** and for **n=1** also a **class-sign**."

Elementary formula

These are of the form **a(b)** (as indicated above, this signifies ***'b** is a member of the class **a'***)

where **b** is a **type n sign**

and **a** is the next higher type, i.e., a **type n+1 sign**. Elementary formulas may be thought of as the fundamental building blocks that all formulas are built of.

Formula

A formula can be one of the elementary formulas as defined above, or a formula of the form

~(p)	(which means 'not' **p**)
(p) ∨ (q)	(which means **p** 'or' **q**)
x∀(p)	(which means 'for all' **x**, **p**)

where **p** and **q** are any formulas, and **x** is any variable. Note that this is a recursive definition since a formula is defined in terms of formulas; this means that when it is stated that **p** is a formula, then **p** itself must be either an elementary formula or a formula of the form **~(p_2)**, **(p_2) ∨ (q_2)** or **x∀(p_2)**, where $\mathbf{p_2}$and $\mathbf{q_2}$ themselves must be either elementary formulas or a formula of the form **~(p_3)**, **(p_3) ∨ (q_3)** or **x∀(p_3)**, and so on, until every $\mathbf{p_n}$ or $\mathbf{q_n}$ is an elementary formula.

Note that '**n-place relation-signs**' and '**class-signs**' are particular types of formulas, see above.

Propositional formula

This is a formula with no free variable.

Substitution of a variable in a formula

Subst

Gödel introduces a term **Subst a(v|b)** to refer to the substitution of the variable **v** in the formula **a** by some symbol string of the formal system (represented here by **b**). By the definition, **b** must be of the same sign as the variable **v**.

Note that, depending on what version of the translation you are using, **Subst** may be represented in this format:

$$Subst\left(a \; \begin{matrix} v \\ b \end{matrix}\right)$$

which is the format using in Gödel's original paper. The same applies to the format of the related **Sb** function which we will encounter later.

> "By **Subst a(v|b)** (where **a** stands for a formula, **v** a variable and **b** a sign of the same type as **v**) we understand the formula derived from **a**, when we replace **v** in it, wherever it is free, by **b**. Where **v** does not occur in **a** as a free variable, we must put **Subst a(v|b) = a**. Note that '**Subst**' is a sign belonging to metamathematics."

Type-lift

Gödel says that given any valid formula, if we replace ***every*** variable of that formula by a variable that is one type higher, then we have created another valid formula. So if a formula contained only **type 1** and **type 2** variables, and we replace every **type 2** variable by a **type 3** variable, and we replace every **type 1** variable by a **type 2** variable, we then have another formula

> "We say that a formula **a** is a **type-lift** of another one **b**, if **a** derives from **b**, when we increase by the same amount the type of all variables appearing in **b**."

Note: The notion of type-lift is peripheral to the main argument of the proof.

---

[1] Note: It might be observed that there are a finite number of alphabetical symbols such as **x**, **y**, **z**, and that in the actual formal system, there are no symbols **1**, **2**, **3**. However, there are several ways that the variables can be represented using only the symbols **x**, ***f*** and **0**, for example: **x*f*0**, **x*f*0*f*0**, **x*f*0*f*0*f*0**, for $\mathbf{x}_1$, $\mathbf{y}_1$, $\mathbf{z}_1$, …, and **x*ff*0**, **x*ff*0*ff*0**, **x*ff*0*ff*0*ff*0**, for $\mathbf{x}_2$, $\mathbf{y}_2$, $\mathbf{z}_2$, …, and so on. Hence there is no limit on the number of possible variables.

A Step by Step Guide to Gödel's Incompleteness Proof

The axioms and rules of the formal system

As is the case for the formal system **P** in general, the precise details of the axioms of this system are not important. It might be wondered whether the system **P** has sufficient axioms and rules of inference - that perhaps some detail has been omitted which renders the system not powerful enough to be complete. But this would be to miss the point - which is that Gödel's proof can *in principle* be applied to any formal system (provided it contains a certain amount of basic arithmetic, such as defining natural numbers and basic operations on those numbers). So that even if there was some deficiency in the system **P,** the overall thrust of the argument would remain and be applicable to all formal systems which include a basic arithmetical core. That said, you still need to be familiar with the terminology that Gödel uses if you are to follow the principle of Gödel's argument.

Note that, while in most logical systems in common use today, every variable in an axiom is bound by a quantifier, in Gödel's system P, this is not the case - the axioms can have variables which are not bound by a quantifier. However, every such axiom can be converted to a formula in which all the variables are bound, as will be seen below.

The rules of inference of the system P

There are two rules of inference .

1. If the formula **(~(b))** ∨ **(c)** is an axiom or is a proven formula, and **b** is an axiom or is proved, then the formula **c** is proved.
2. If the formula **a** is an axiom or is a proven formula, then the formula **v∀(a)** is proved, where **v** is any variable.

> "A formula **c** is called an **immediate consequence** of **a** and **b**, if **a** is the formula **(~(b))** ∨**(c)**, and an **immediate consequence** of **a**, if **c** is the formula **v∀(a)**, where **v** denotes any given variable. The class of **provable formulae** is defined as the smallest class of formulae which contains the axioms and is closed with respect to the relation 'immediate consequence of'. "

Hence, given an axiom or proven formula whose variables are not bound by a quantifier, by repeated application of rule 2 we can obtain the same formula where each variable is bound by a quantifier.

The axioms of the system P

Gödel divides the axioms into five sections.

Note that the axioms **II**, **III**, and **V** are actually Axiom Schema; this means that each 'axiom' actually represents infinitely many axioms, which are given by replacing the appropriate variables in the schemas by specific values. For example, for the Axiom Schema II.1 below ($\mathbf{p \vee p \supset p}$), we could replace **p** by any formula of the formal system to give a single axiom.

Note: the axioms as given include some symbols which are not actually symbols of the formal system **P**. The reason for this is as follows:

The system **P** uses a very small set of symbols. There are several symbols in common usage which are effectively abbreviations for long combinations of the basic symbols of the system **P**. Gödel uses these symbols because the actual representation in the notation of the system **P** would be very lengthy and difficult to read. The symbols Gödel uses are listed is below, along with the equivalent representation by symbols of the formal system **P**:

⊃ 'implies'	$\mathbf{a \supset b}$ is equivalent to $\mathbf{(\sim a) \vee b}$
∃ 'there exists'	$\mathbf{(\exists u)b}$ is equivalent to $\mathbf{\sim(u\forall(\sim b))}$
≡ 'equivalence'	$\mathbf{a \equiv b}$ is equivalent to $\mathbf{(a \supset b) \wedge (b \supset a)}$
= 'equals'	$\mathbf{x_1 = y_1}$ is defined as $\mathbf{x_2\forall\ (x_2(x_1) \supset x_2(y_1))}$[1]
∧ 'and'	$\mathbf{c \wedge d}$ is equivalent to $\mathbf{\sim((\sim c) \vee (\sim d))}$

Note that Gödel also uses the symbol **.** in the axiom I.3; this is equivalent to the ∧ symbol.

Note that, for convenience Gödel follows convention in omitting some brackets that would be present in the actual corresponding formula of the formal system. For example, in the following, wherever the ⊃ symbol (the 'implies' symbol) occurs, all of the expression to the left of the symbol implies all of the expression to the right of the symbol, unless that part of the expression containing the ⊃ symbol is enclosed by brackets.

Axioms I:

These are elementary axioms about natural numbers.

1. $\mathbf{\sim(}f\mathbf{x_1 = 0)}$

No number (other than zero) can be equal to zero.

2. $f\mathbf{x_1} = f\mathbf{y_1} \supset\ \mathbf{x_1 = y_1}$

Given two numbers, if we add one to each number, and the resultant numbers are equal, then that implies (the ⊃ symbol) that the original numbers are both equal.

3. $\mathbf{x_2(0) \wedge x_1\forall(x_2(x_1) \supset x_2(}f\mathbf{x_1))\ \supset\ x_1\forall(x_2(x_1))}$

This is what is usually called an axiom of induction. Given any class of natural numbers, if **0** is a member of that class, and if, for every natural number $\mathbf{x_1}$, if $\mathbf{x_1}$ being a member of that class implies that $\mathbf{x_1 + 1}$ is also a member of that class, then every natural number is a member of that class.

Axioms II:

These are Axiom Schemas, based on axioms of classical propositional logic. In the Schemas below **p**, **q** and **r** can be any formula of the formal system.

1. **p ∨ p ⊃ p**
For any formula **p**, **p** or **p** implies **p**.

2. **p ⊃ p ∨ q**
For any formula **p**, **p** implies '**p** or any other formula **q**'.

3. **p ∨ q ⊃ q ∨ p**
For any formulas **p** and **q**, '**p** or **q**' implies '**q** or **p**'.

4. **(p ⊃ q) ⊃ (r ∨ p ⊃ r ∨ q)**
For any formulas **p**, **q** and **r**, '**p** implies **q**' implies that '(**r** or **p**) implies (**r** or **q**)'

Axioms III:

In these Axiom Schemas,
a is any formula,
v is any variable,
b is a formula and
c is a sign;
b and **c** are subject to certain conditions.

1. **v∀(a) ⊃ Subst a(v|c)**
This states that, if for all **v**, formula **a** applies, then every formula given by the substitution of a valid value **c** for **v** applies.

As Gödel notes, the **Subst** function is a function of the meta-language, not of the formal system **P**. So while there is no expression of the formal system that corresponds to the above expression when values are simply inserted for **a**, **b**, **c**, and **v**, there are formulas that correspond to the expressions given when appropriate values are inserted for **a**, **b**, **c**, and **v** and the ***value*** given by the meta-language function **Subst a(v|c)** (which is a symbol string of the system **P**) is substituted in place of **Subst a(v|c)**.

Note that normally we use the term substitution to refer to the substitution of a variable by a specific value which is a member of the domain of the variable. Here **Subst** allows variables to be substituted by variables, provided they are of the same type, subject to certain conditions; this is akin to changing the names of the variables in a formula.

2. **v∀(b ∨ a) ⊃ b ∨ v∀(a)**

This states that, ‘if for all **v**, **b** or **a**’ applies, then **‘b** or for all **v**, **a’** applies (if **v** is not a free variable in **b**) .

> “Every formula derived from the two schemata by making the following substitutions for **a**, **v**, **b**, **c** (and carrying out in I the operation denoted by “Subst”): for **a** any given formula, for **v** any variable, for **b** any formula in which **v** does not appear free, for **c** a sign of the same type as **v**, provided that **c** contains no variable which is bound in **a** at a place where **v** is free.
> **c** is therefore either a variable or 0 or a sign of the form ***f…fu*** where **u** is either **0** or a variable of type 1.”

Axiom IV:

In this Axiom Schema,
v and **u** are variables, where **u** is one **type** higher than **v**, and
a is formula which does not have **u** as free variable

1. **(∃u)(v∀(u(v)≡a))**
For every formula, there is a corresponding class whose members satisfy that formula.

Axiom V:

In this axiom we have type 1 and type 2 variables.

1. $\mathbf{x_1}$**∀**$\mathbf{(x_2(x_1) \equiv y_2(x_1)) \supset x_2 = y_2}$
If two classes have precisely the same members, then they are identical.

Further axioms are defined by this axiom by applying Type-lift to the basic formula above.

---

[1] For all type 2 classes $\mathbf{x_2}$, if $\mathbf{x_1}$ is a member of that class, then $\mathbf{y_1}$ is a member of that class. Since this applies to *all* type 2 classes, then if both $\mathbf{x_1}$ and $\mathbf{y_1}$ are a member of a class that has only one member, then it must be the case that $\mathbf{x_1}$ is identical to $\mathbf{y_1}$. Similarly for higher type classes.

A Step by Step Guide to Gödel's Incompleteness Proof

The Gödel numbering system

In his proof Gödel used a method so that any combination of symbols of the formal system can be represented as a natural number. This method has since been called the Gödel numbering system. In the paper it is given in the paragraph beginning *"The basic signs of the system **P**…"* .

Gödel's numbering system means that every formula of the formal system **P** can be represented as a number; any every proof-schema of the formal system **P** can also be represented as a number.

The Gödel numbering method is *intended* (see below) to be a one-to-one function, so that every string of symbols is matched to a unique number. No two strings can have the same number, and no two numbers can match to the same string. The numbering method can be considered as a two-step process. The first step is to match every symbol of the formal system to a specific number. The basic symbols **0**, ***f***, ~, ∨, ∀, **(**, and **)** are matched to the numbers **1**, **3**, **5**, **7**, **9**, **11** and **13** as below:

0	↔	**1**	***f***	↔	**3**	~	↔	**5**	∨	↔	**7**
∀	↔	**9**	**(**	↔	**11**	**)**	↔	**13**			

where ∨ is the symbol for 'or', and ∀ is the symbol for 'for all'.
For variables, the matching is done using prime numbers greater than **13**. For **type 1** variables, we use **17**, **19**, **23**, etc. For **type 2** variables, we use $\mathbf{17^2}$, $\mathbf{19^2}$, $\mathbf{23^2}$, etc. In general, for a **type n** variable, we use $\mathbf{17^n}$, $\mathbf{19^n}$, $\mathbf{23^n}$, etc.

> "variables of type **n** are given numbers of the form $\mathbf{p^n}$ (where **p** is a prime number > **13**)."

In this way, every basic symbol and every variable of the formal system has a unique corresponding natural number.

This is the definition of a function, and although Gödel does not give this function a name, it is useful to call this function **ψ**; for example, **ψ[**∀**] = 9**. We will need to refer to this function later on. The function **ψ** gives a corresponding series of natural numbers for every symbol string of the formal system.

> **NB**: It is important to bear in mind that while a variable of the formal system **P** is actually a string of symbols, for the purposes of the Gödel numbering function, every variable is considered in the same way as a single symbol. So, for convenience and to avoid undue verbosity, from this point forward, when the term 'symbol' is used, unless otherwise indicated, that means either a single basic symbol of the system **P** or a variable of the system **P**.

The next step is to convert such a series of natural numbers into a single natural number in a way that retains all the original information. This involves the use of prime numbers: **2**, **3**, **5**, **7**, **11**, … . The method is to raise the power of the $\mathbf{n}^{th}$ prime number to the value of the $\mathbf{n}^{th}$ number in the series.

> "to every finite series of basic signs … there corresponds, one-to-one, a finite series of natural numbers. These finite series of natural numbers we now map (again in a one-to-one correspondence) on to natural numbers, by letting the number 2^{n1}, 3^{n2} … p_k^{nk} correspond to the series n_1, n_2, … n_k, where p_k denotes the **k**-th prime number in order of magnitude."

For example, for the sequence **~(ff0)**, the corresponding number series is **5**, **11**, **3**, **3**, **1**, **13**. The single number corresponding to this series is $2^5 \cdot 3^{11} \cdot 5^3 \cdot 7^3 \cdot 11^1 \cdot 13^{13}$ (where · indicates multiplication).

The name Gödel gives to this function is **φ(a)**, where **a** is some combination of symbols of the formal system. Today it is commonly called the *'Gödel numbering function'*. The inverse of this function can be applied in order to retrieve the original combination of symbols.

Correspondence of relations

In the part beginning with *"Suppose now one is given a class or relation…"*, Gödel states that given a relationship between symbol strings of the formal system, the aim is to produce a corresponding relation between the corresponding numbers given by the function **φ**. And the aim is also that if the relationship between the symbol strings applies, then the corresponding relation between the corresponding numbers also holds. In that way, the relations between the corresponding numbers mirror precisely the relationships between the symbol strings.

Usage of Italics

NB: The following is a crucially important detail and the failure to observe the distinction between certain words in italics and not in italics has been a source of confusion to many people.

The Gödel numbering function **φ** gives for every symbol string a corresponding natural number. In Gödel's paper such *numbers* are referred to by ***italics***, so:[1]

formula	=	**φ**(formula)
axiom	=	**φ**(axiom)
provable formula	=	**φ**(provable formula)
propositional formula	=	**φ**(propositional formula)

It should also be noted that Gödel when refers to *variable*s in italics, he is not referring to the Gödel numbering function **φ**, but to the **ψ** function (see above), so:

variable	=	**ψ**(variable)

NB: Gödel also refers to numbers as a '*series of formulas*' or as a *'proof-schema'*, where a *'proof-schema'* is a special case of a '*series of formulas*'. These are referred to in the relations 22 and 44 which occur later in the proof. A '*series of formulas*' or a *'proof-schema'* is a number **x** with the value $\mathbf{2^{\varphi[\text{ formula1 }]} \cdot 3^{\varphi[\text{ formula2 }]} \cdot 5^{\varphi[\text{ formula3 }]} \cdot \ldots}$, in other words, the exponents of the prime factors of **x** are Gödel numbers, rather than **x** itself being a Gödel number. So it is important to note that the number **x** in this case does *not* correspond by Gödel numbering to a series of formulas or a proof schema of the formal system **P**, i.e.**:**

x ≠ **φ**(series of formulas of the formal system **P**),

x ≠ **φ**(proof-schema of the formal system **P**).

It is crucial to remember that the same word in Gödel's proof - in plain text - and in *italics* - represents two quite separate concepts. The failure to appreciate this distinction has led many people astray in their attempt to understand the proof. It is important to always bear in mind the distinction between a statement which ***actually*** states:

> "There exist propositional formulas **A** of the system **P** such that neither **A** nor the negation of **A** are provable by any proof-schema of the system **P**"

and a similar statement of Gödel's that is actually referring to natural numbers**,** such as the statement:

> "There exist *propositional formulae* **a** such that neither **a** nor the *negation* of **a** are *provable formulae*".

The above statement is not a statement that refers directly to formulas of the system **P**; it is a statement about numbers which correspond to expressions of the formal system, so that the statement is actually stating:

> "There exists a number **a**, where **a = φ[A]** and **A** is a formula, such that there does not exist any number **b**, where **b = φ[B]** and **B** is a proof-schema and where **B** is a proof of **A** or ~**A**."

Because this dual use of the same word can result in confusion, it will not be used in this guide, except to refer to the italicized words in Gödel's original paper.

---

[1] This is the same as in the original German; in van Heijenoort's translation, SMALL CAPITALS are used for the same purpose.

A Step by Step Guide to Gödel's Incompleteness Proof

Primitive Recursive Relations and Functions

In Gödel's section *'Recursive Relations'* , the first paragraph (beginning *"We now introduce a parenthetic consideration…"*) refers to recursive functions and relations of number-theory. Nowadays we call such functions and relations 'primitive recursive' functions and relations, rather than simply 'recursive' number-theoretic functions and relations. Primitive recursive functions and relations are all number-theoretic, that is, they are functions/relations where the entities that the function or relation refers to are all natural numbers, or variables for natural numbers.

Gödel had two principal reasons for using primitive recursive relations:

1. For given values of its free variables, every primitive recursive relation is provable, or else its negation is provable from the axioms of number theory, and
2. Every primitive recursive relation can be expressed in the formal system **P**.

Both of these properties are used in the proof of Gödel's Proposition V (which we will come to later, see Part 8: Gödel's Proposition V).

Primitive recursive relations (for given values of the free variables) are always provable from the axioms of number theory since a primitive recursive relation can always be calculated as being a correct or incorrect proposition, using the axioms of number theory, by a method that always has a finite number of precisely defined steps.[1]

When it is asserted that every primitive recursive relation can be expressed in the formal system, what is being asserted is that:

1. since a number-theoretic relation is a relation where the entities that the relation refers to are all natural numbers or variables for natural numbers, then there is a formula of the formal system that corresponds to that number-theoretic relation, and
2. given any primitive recursive relation with ***n*** free variables, there is always a corresponding formula in the formal system which also has ***n*** free variables - and it has the property that:
3. whenever the free variables of the primitive recursive relation are substituted by specific values, if the resulting relation is provable, then the corresponding formula of the formal system is also provable; if the negation of the resulting relation is provable, then the corresponding negation formula of the formal system is also provable.

In this guide we will not go into the details of Gödel's definitions of primitive recursive relations; at this point it is sufficient to understand the principles of the main properties of primitive recursive relations/functions that is important. Gödel's point is that the actual relations and functions 1-45 that he uses in his proof (and which you will find in Gödel's paper under the heading 'Relations 1-46') can be shown to be primitive recursive.

The Epsilon notation ε

Note: Gödel introduces the **ε** notation in this section, and the **ε** notation is used in subsequent sections:
εx R(x, y) is defined to be a function with one free variable **y**, and whose value is *the smallest value of* **x** *for which the relation* **R(x, y)** *holds*. If there is no such smallest number, then the value of **εx R(x, y)** is **0**. The function **εx R(x, y)** is a primitive recursive function if the relation **R(x, y)** is primitive recursive *and* provided the relation **R(x, y)** includes an upper bound on the value of **x**.

A brief overview of primitive recursion is given below. If you wish to go into the details of the definition and treatment of primitive recursion, there are plenty of sources of such information available; the reason it is not included here is that such details will not significantly assist your understanding of the thrust of Gödel's argument.[2]

Primitive Recursive *Functions*

A primitive recursive function can always be evaluated by a method that always has a finite number of precisely defined steps.

For example, the 'factorial' function is defined as:
0! = 1
(n+1)! = (n+1) · n!
where · indicates multiplication.

The intention of the factorial function **n!** is that, for any natural number **n**, it gives the value of **n** multiplied by all the natural numbers less than **n** (except zero). For example,
1! = 1
2! = 2 · 1
3! = 3 · 2 · 1
4! = 4 · 3 · 2 · 1, and so on.

So when we have **(n+1)! = (n+1) · n!**, then that indicates that **n! = n · (n-1)!**, providing **n-1** is greater than zero, so that:
(n+1)! = (n+1) · n! = (n+1) · n · (n-1)! (providing **n**, **n-1**, **n-2**, **n-3**, … are all greater than **0**)
or we might say that:
n! = n · (n-1)! = n · (n-1) · (n-2) · (n-3) · … (again, providing **n**, **n-1**, **n-2**, **n-3**, … are all greater than **0**)

For any value of **n**, the value of **n!** can be calculated by a simple method, like this:
1. Let **w = n**
2. Let **y = n**
3. **y = y - 1**
4. If **y > 1** then **w = y · w** and return to step 3
5. Otherwise the value of **n!** is **w**

Suppose we use this method to calculate **4!**

1. **w = 4**
2. **y = 4**
3. **y = 3**
4. **y > 1** so **w = 4 · 3**
3. **y = 2**
4. **y > 1** so **w = 4 · 3 · 2**
3. **y = 1**
4. Since **y = 1**, no return to step 3
5. **4! = w = 4 · 3 · 2**

So, given any value of **n**, the value of the function **n!** can always be determined in a finite number of steps of calculation. We know that the calculation must stop at some point, since we must always reach the point where **y = 1**, and so **n!** is a primitive recursive function. This method could be made into a computer program, and every primitive recursive function can be calculated by a simple computer program (but you should be aware that there are functions that evaluate as natural numbers by computer programs, but which are not primitive recursive functions).

Simple primitive recursive functions include the addition function **x + y**, the multiplier function **x · y** and the exponential function $\mathbf{x^y}$. Gödel's section *'Recursive Relations'* introduces more complex primitive recursive functions which he defines in terms of simpler primitive recursive functions.

Primitive Recursive *Relations*

Gödel gives a precise definition of primitive recursive relations in terms of primitive recursive functions. The definitions are such that, for every primitive recursive relation, there is always a method with a finite number of steps that can calculate if the relation or its negation can be proved to follow from the axioms of number theory. For example, using the simple factorial function above, we could have the relation
x = (y + 3)!

Given any **x** and any **y**, it can be calculated, in a finite number of steps, if the relation is correct or incorrect for those values of **x** and **y**.

Simple primitive recursive relations include the 'less than' relation **x < y** and the equality relation **x = y**. Gödel's section *'Recursive Relations'* introduces more complex primitive recursive relations which he defines in terms of simpler primitive recursive functions and relations.

---

[1] Note: There are relations and functions that can be calculated by a finite number of steps, but which do not fall within the strict definition of primitive recursion. Gödel did not use such relations or functions in his proof.
[2] You may read elsewhere that a comprehensive knowledge of the details of primitive recursion is essential to an understanding of Gödel's paper, but this is not the case. In fact, if you arrive at a full understanding of the paper up to and including Gödel's Proposition V, you will see that primitive recursion is a peripheral matter.

A Step by Step Guide to Gödel's Incompleteness Proof

Gödel's Relations 1- 45 of Natural Numbers

> Note again that in Gödel's paper the epsilon symbol **ε** is used in several definitions; the meaning of the term **ε** (see also Epsilon notation ε in the previous section) is given by:
>
> **εx R(x, y)** is a function with the free variable **y**, and whose value is *the smallest number* **x** *for which the relation* **R(x, y)** *holds*, and if there is no such smallest number, then the value of **εx R(x, y)** is **0**.

In the section entitled "*Relations 1-46*" Gödel defines various relations/functions. The relations/functions 1-45 are all primitive recursive; relation 46 is not defined as primitive recursive. Note that while in modern terminology, all variables of a relation are normally enclosed after the relation name in brackets (e.g. Rel(x, y)), Gödel often places variables before and after the relation name (e.g. x Rel y)). The names are mainly abbreviations of German words, see notes at the foot of this page for the words and the English translations.

Some of these functions (nos. 3, 4, 5, 16, 30) are recursively defined in terms of the value of the function itself. For example, the function 4 is the factorial function mentioned in the previous page.

For most of the relations/functions, Gödel adds comments describing the relationship between symbol strings of the formal system that corresponds (by the Gödel numbering function) to the given relation between natural numbers. In the following, further notes are included to assist the reader. In these notes variables for symbols/symbol strings of the formal system **P** are represented by colored capital letters, e.g., **X**, **Y**.[1]

It is crucial to remember that all these relations/functions are purely number-theoretic - that is, that they only refer to natural numbers and variables whose domain is natural numbers. It is important not to use intuition to jump to conclusions, nor to make assumptions about the corresponding relationships between symbol strings of the formal system. Any such correspondence must follow in a logical manner.

It should be noted that in some cases, a corresponding number is obtained not by Gödel numbering function **φ** but by the **ψ** function (as referred to previously, see here). This does not present any problems, since given a number obtained via the **ψ** function for a symbol, one can always obtain the Gödel number by applying the Gödel numbering function **φ** to that single symbol. For example $\psi(f) = 3$, and $\varphi[f] = 2^3$.

NB: when the term 'symbol' is used below, unless otherwise indicated, that can mean either a single basic symbol of the system **P** or a variable of the system **P**. Also note that some of the 'formulas' used in the examples are not proper formulas; this is so that the examples are not overly long.

Relations/functions 1-5:
Basic Arithmetical Concepts.

These are basic arithmetical concepts, principally concerning properties of prime numbers. Since the Gödel numbering method is based on the fact that every natural number has a unique factorization into its constituent prime factors, these relations/functions underpin the subsequent relations and functions. The details of the definitions are given below; clicking "show" will show the details for that definition.

1. **x/y**
This is a relation that asserts that **y** is divisible by **x** with no remainder.
e.g., the relation **72/6** applies, but **73/6** does not apply.

$x/y \equiv (\exists z)[z \leq x \ \& \ x = y \cdot z]$

2. **Prim(x)**
This is a relation that asserts that **x** is a prime number. i.e., **x** is one of **2**, **3**, **5**, **7**, **11**, **13**, **17**, **19**, **23**, **…**
e.g., the relation **Prim(7)** applies, but **Prim(8)** does not apply.

$Prim(x) \equiv \sim(\exists z)[z \leq x \ \& \ z \neq 1 \ \& \ z \neq x \ \& \ x/z] \ \& \ x > 1$

3. **n Pr x**
This is a function whose value is the $\mathbf{n^{th}}$ largest prime number that is a factor of **x**.
e.g., $\mathbf{4\ Pr\ [2^3 \cdot 3 \cdot 5^3 \cdot 7^4 \cdot 11 \cdot 13^3] = 7}$
(**7** is the $\mathbf{4^{th}}$ largest prime factor of $\mathbf{2^3 \cdot 3 \cdot 5^3 \cdot 7^4 \cdot 11 \cdot 13^3}$)

$0 \ Pr \ x \equiv 0$
$(n+1) \ Pr \ x \equiv \varepsilon y[y \leq x \ \& \ Prim(y) \ \& \ x/y \ \& \ y > n \ Pr \ x]$

4. **n!**
This is the factorial function as mentioned previously in this guide. Its value is given by multiplying together **n** and all the natural numbers that are smaller than **n** (except zero).
e.g., $\mathbf{4! = 4 \cdot 3 \cdot 2 \cdot 1}$

5. **Pr(n)**
This is a function whose value is the $\mathbf{n^{th}}$ largest prime number.
e.g., **Pr(5) = 11** (the $\mathbf{5^{th}}$ largest prime is **11**; the four smaller primes are **2**, **3**, **5** and **7**)

$Pr(0) \equiv 0$
$Pr(n+1) \equiv \varepsilon y[y \leq \{Pr(n)\}! + 1 \ \& \ Prim(y) \ \& \ y > Pr(n)]$

Relations/functions 6-23:
Relations/functions that correspond to the construction of symbol strings

We now move from basic arithmetical definitions to the definition of functions that correspond to operations on symbols and symbol strings of the formal system - and to the definition of relations that correspond to assertions regarding symbol strings of the formal system.

The formulas of the formal system are combinations of the symbols of the system that satisfy certain conditions. One of the goals of these relations and functions is to lead to the definition of relation 23, which corresponds to the assertion that a given symbol string is a formula of the formal system.

It will be noted that the functions/relations are presented here in a different order to that given by Gödel; here they are grouped according to their similarity and purpose.

Functions 6, 7, 9: Basic string operations

These are basic functions that correspond to:

6. **n Gl x**: the operation of obtaining the symbol at a particular position in a symbol string
7. **l(x)**: the operation of counting the number of symbols in a symbol string
9. **R(x)**: the operation of obtaining the Gödel number of a single symbol

6. **n Gl x**

This is a function which is defined in terms of the function **n Pr x** above, and gives the value of the smallest factor of the exponent of the $\mathbf{n^{th}}$ largest prime in the number **x**.

e.g., $\mathbf{5\ Gl\ [2^3 \cdot 3^3 \cdot 5^1 \cdot 7^{11} \cdot 11^{19} \cdot 13^3] = 19}$ (the $\mathbf{5^{th}}$ largest prime in **x** is **11**, and its exponent is **19**)

If **x** is the Gödel number of the formula **X**, i.e., $\mathbf{x = \varphi[X]}$, then **n Gl x** corresponds to:

the symbol at the $\mathbf{n^{th}}$ position in the symbol string **X**

$$\mathbf{n\ Gl\ x \equiv \varepsilon y[y \leq x\ \&\ x/(n\ Pr\ x)^y\ \&\ \neg(x/(n\ Pr\ x)^{y+1})\ ]}$$

"**n Gl x** is the $\mathbf{n^{th}}$ term of the series of numbers assigned to the number **x** (for $\mathbf{n > 0}$ and **n** not greater than the length of this series)."

e.g., if $\mathbf{X} = \boldsymbol{ff}\mathbf{0(y_1)}$, then $\mathbf{x = \varphi[X] = 2^3 \cdot 3^3 \cdot 5^1 \cdot 7^{11} \cdot 11^{19} \cdot 13^3}$

then $\mathbf{5\ Gl\ [2^3 \cdot 3^3 \cdot 5^1 \cdot 7^{11} \cdot 11^{19} \cdot 13^3] = 19}$ to correspond to:

the operation of obtaining the $\mathbf{5^{th}}$ symbol of the symbol string $\boldsymbol{ff}\mathbf{0(y_1)}$, which is $\mathbf{y_1}$

where $\mathbf{19 = \psi[y_1]}$.

7. **l(x)**

This is a function whose value is the number of unique prime factors in **x**.

e.g., $\mathbf{l(2^3 \cdot 3^3 \cdot 5^3 \cdot 7^1) = 4}$ (there are only four unique prime factors: **2**, **3**, **5**, **7**)

If **x** is the Gödel number of the formula **X**, i.e., **x = φ[X]**, then **l(x)** corresponds to:

the operation of counting the number of individual symbols/variables in the formula **X**

$$\mathbf{l(x) \equiv \varepsilon y[y \leq x \ \&\ y \ Pr \ x > 0 \ \&\ (y + 1) \ Pr \ x = 0]}$$

"**l(x)** is the length of the series of numbers assigned to **x**."

e.g., if **X** = ***fff*0**, then $\mathbf{x = \varphi[X] = 2^3 \cdot 3^3 \cdot 5^1 \cdot 7^1}$

then $\mathbf{l(2^3 \cdot 3^3 \cdot 5^1 \cdot 7^1) = 4}$ corresponds to:

the operation of counting the number of symbols in ***fff*0**, which is **4**.

9. **R(x)**

This is a function which raises **2** to the power of **x**, e.g., $\mathbf{R(19) = 2^{19}}$

So **R(x) = φ[X]**

where **x = ψ[X]** (as given by **ψ** - the Psi function - the function that assigns a unique number to a symbol **X** of the formal system), and where **X** is a single symbol or a variable of the formal system **P**.

"**R(x)** corresponds to the number-series consisting only of the number **x** (for **x > 0**)."

e.g., if **X** = **(**, then **x = ψ[( ] = 11**

and $\mathbf{R(x) = 2^{11}}$ corresponds to:

the operation of obtaining the Gödel number of the symbol **(**, which is $\mathbf{\varphi[\ (\ ] = R(\psi[\ (\ ]) = 2^{11}}$

Function 8: Defining numbers that correspond to a concatenation of two strings

x * y corresponds to the operation of concatenating (joining together) two symbol strings.

8. **x * y**

This function can be thought of as 'shifting' up the prime factors of the second number **y**. It is probably easiest to understand this function by an example.

e.g., $[2^3] * [2^3 \cdot 3^3 \cdot 5^3 \cdot 7^1] = 2^3 \cdot 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^1$

And **x * y** = **φ[XY]**, where **x** = **φ[X]** and **y** = **φ[Y]**

i.e., **φ[X] * φ[Y] = φ[XY]**

that is, if **x= φ[X]** and **x= φ[Y]**, then the function **x * y** corresponds to:

the operation of concatenating **X** and **Y**

where **X** and **Y** are symbol strings of the formal system **P**.

$$x * y \equiv \varepsilon z[z \leq [Pr\{l(x) + l(x)\}]^{x+y}$$
$$\& \ (n)\ [n \leq l(x) \Rightarrow n\ Gl\ z = n\ Gl\ x]$$
$$\& \ (n)\ [0 < n \leq l(y) \Rightarrow \{n + l(x)\}\ Gl\ z = n\ Gl\ y]]$$

"**x * y** corresponds to the operation of 'joining together' two finite series of numbers."

e.g., if **X** = ***f*** and **Y** = ***fff*0**, then **x** = **φ[X]** = 2^3, and **y** = **φ[Y]** = $2^3 \cdot 3^3 \cdot 5^3 \cdot 7^1$

then $2^3 * 2^3 \cdot 3^3 \cdot 5^3 \cdot 7^1 = 2^3 * 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^1$ corresponds to:

the operation of concatenating the symbol strings ***f*** and ***fff*0** to give ***ffff*0**.

Functions 16, 17: Defining numbers that correspond to simple strings

These are functions that correspond to:

16. **n N x**: the operation of of **n** repetitions of prefixing a symbol string by the symbol ***f***
17. **Z(n)**: the operation of **n** repetitions of prefixing the symbol **0** by the symbol ***f***

16. **n N x**

For this function,

0 N x = φ[X]
1 N x = φ[*f*X]
2 N x = φ[*ff*X]
…
…
n N x = φ[*fff*…*ff*X], where there are **n** occurences of ***f***
that is, if **x = φ[X]**, then the value of **n N x** corresponds to:
the operation of prefixing **n** repeated occurrences of the symbol ***f*** in front of the symbol string **X**

0 N x ≡ x
(n + 1) N x ≡ R(3) * n N x

"**n N x** corresponds to the operation: "**n**-fold prefixing of the sign '***f***' before **x**." "

e.g., if **X** is ***f*0**, and **φ[*f*0]** = $2^3 \cdot 3^1$
then **3 N** $(2^3 \cdot 3^1) = 2^3 \cdot 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^1$ corresponds to:
the operation of prefixing ***f*0** by three instances of ***f***, which gives ***ffff*0**.

17. **Z(n)**
For this function,

Z(n) = n N R(1), and

R(1) = 2^1, so

Z(n) = n N 2^1= φ[*fff…ff*0], where there are **n** occurences of ***f***, since by the Gödel numbering function, **2^1** corresponds to the formal symbol for zero, which is **0**. So the value of **Z(n)** corresponds to:

the operation of prefixing **n** repeated occurrences of the symbol ***f*** in front of the symbol **0**

"**Z(n)** is the *number-sign* for the number **n**."

e.g., **Z(5) = φ[*fffff*0] = $2^3 \cdot 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^3 \cdot 13^1$** corresponds to:

the operation of obtaining the Gödel number of ***fffff*0**

Note: Gödel is being rather disingenuous here in simply asserting that "**Z(n)** is the *number-sign* for the number **n**", without giving any further details. By that assertion he is asserting that **Z(n) = φ[n]**, i.e., that **Z(n)** is the same as the Gödel numbering function for the number **n**. But the domain of the variable of the Gödel numbering function **φ** is *all* symbol strings of the formal system, whereas the domain of the function **Z** is the domain of natural numbers. To make the assertion means that the domain of the common variable **n** would have to be restricted to symbol strings of the format ***0, f0, ff0, fff0, …*** , and it would not be allowable, for example to have an equality between **5** and ***fffff0***, since if such equality were permissible, we would have **φ[5]** which is not a valid expression. Furthermore, the symbols for zero and the successor symbol for the formal system could be any symbols, for example, **3** and **7**, which would rule out using the symbols **3** and **7** for three and seven, as is conventional for number-theoretic relations. Gödel's stipulation requires that his number-theoretic relations must necessarily use precisely the same format for natural numbers as the formal system in question - which begs the question as to why the two systems should be interdependent in this manner. We will return to this matter later when considering Gödel's Proposition V.

Functions 10, 13, 14, 15: Defining numbers that correspond to complex strings

These functions correspond to operations that build up more complex symbol strings from simpler ones. They correspond to:

10. **E(x)**: the operation of putting brackets around a symbol string
13. **Neg(x)**: the operation of creating the negation of a symbol string
14. **x Dis y**: the operation of joining two symbol strings by the 'or' symbol ∨
15. **x Gen y**: the operation of prefixing a symbol string by a variable and the 'for all' symbol ∀

10. **E(x)**

This function can be thought of as multiplying the number **x** by an 'initial' $\mathbf{2^{11}}$, and 'shifting' each of the exponents of the prime factors of the number **x** up to the next largest prime, and then multiplying the result by a 'final' next largest prime with an exponent of **13**. The exponent of the 'initial' **2** is **11**, since **11** is the matching number for the opening bracket (**11 = ψ[( ]**), and the exponent of the 'final' prime is **13**, since **13** is the matching number for the closing bracket (**13 = ψ[ )]**). It is probably easiest to understand this function by an example:

e.g., $\mathbf{E(2^3 \cdot 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^1) = 2^{11} \cdot 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^3 \cdot 13^1 \cdot 17^{13}}$

In this example **17** is the additional 'final' next largest prime (with the exponent of **13**).

Note that **E(x) = φ[(X)]**

where **x = φ[X]**, and where **X** is a symbol string of the formal system **P**.

See Function 8, which gives that

φ[(X)] = φ[( ] * φ[X] * φ[ )]

that is, if **x = φ[X]** then the value of **E(x)** corresponds to:

the operation of putting brackets on either side of the symbol string **X**

E(x) ≡ R(11) * x * R(13)

"**E(x)** corresponds to the operation of "bracketing" [11 and 13 are assigned to the basic signs '**(**' and '**)**']."

e.g., if **X** = ***ffff*0**, and $\mathbf{φ[\mathit{ffff}0] = 2^3 \cdot 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^1}$

then $\mathbf{E(x) = φ[(X)] = φ[(\mathit{ffff}0)] = 2^{11} \cdot 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^3 \cdot 13 \cdot 17^{13}}$ corresponds to:

the operation of enclosing the symbol string ***ffff*0** by brackets to give **(*ffff*0)**.

13. **Neg(x)**

This function can be thought of as multiplying the number **x** by an 'initial' $2^{5} \cdot 3^{11}$, and 'shifting' each of the exponents of the prime factors of the number **x** up to the next largest prime, and then multiplying the result by a 'final' next largest prime with an exponent of **13**. The exponent of the 'initial' **2** is **5**, since **5** is the matching number for the negation symbol (**5 = ψ[~]**), and the exponent of the 'initial' **3** is **11**, since **11** is the matching number for the opening bracket (**11 = ψ[( ]**); the exponent of the 'final' prime is **13**, since **13** is the matching number for the closing bracket (**13 = ψ[ )]**). It is probably easiest to understand this function by an example:

e.g., **Neg**$(2^{3} \cdot 3^{3} \cdot 5^{3} \cdot 7^{3} \cdot 11^{1}) = 2^{5} \cdot 3^{11} \cdot 5^{3} \cdot 7^{3} \cdot 11^{3} \cdot 13^{3} \cdot 17^{1} \cdot 19^{13}$

In this example **19** is the additional 'final' next largest prime (with the exponent of **13**).

And **Neg(x) = φ[~(X)]**

where **X** is a string of symbols of the formal system **P**, and ~, **(** and **)** are symbols of the formal system **P**.

Function 8 above gives that

φ[~] * φ[( ] * φ[X] * φ[ )]= φ[~(X)]

that is, if **x = φ[X]** then the value of **Neg(x)** corresponds to:

the operation of putting the symbol ~ in front of the symbol string **X**

i.e., creating the negation of the formula **X**

Neg(x) ≡ R(5) * E(x)

"**Neg(x)** is the *negation* of **x**."

e.g., if **X** = ***ffff*0**, and **φ[*ffff*0]** = $2^{3} \cdot 3^{3} \cdot 5^{3} \cdot 7^{3} \cdot 11^{1}$

then **Neg(x) = φ[~(*ffff*0)]** = $2^{5} \cdot 3^{11} \cdot 5^{3} \cdot 7^{3} \cdot 11^{3} \cdot 13^{3} \cdot 17^{1} \cdot 19^{13}$ corresponds to:

the operation of enclosing the symbol string ***ffff*0** by brackets and prefixing that by the negation symbol to give ~**(*ffff*0)**.

14. **x Dis y**

For this function,

x Dis y = φ[(X) ∨ (Y)]

where **X** and **Y** are symbol strings of the formal system **P**, and ∨ is a symbol of the formal system **P**.

Function 8 above gives that

φ[( ] * φ[X] * φ[ )] * φ[∨] * φ[( ] * φ[Y] * φ[ )] = φ[(X) ∨ (Y)]

that is, if **x = φ[X]** and **y = φ[Y]**, then the value of **x Dis y** corresponds to:

the operation of concatenating the symbol string (**X**), the symbol ∨, and the symbol string (**Y**).

i.e., creating a formula by joining two formulas by the 'or' symbol ∨

x Dis y ≡ E(x) * R(7) * E(y)

"**x Dis y** is the *disjunction* of **x** and **y**."

e.g., if **X** = ***f*0**, and **φ[*f*0]** = $2^3 \cdot 3^1$, and **Y** = ***ff*0**, and **φ[*ff*0]** = $2^3 \cdot 3^3 \cdot 5^1$, then

x Dis y = φ[(*f*0) ∨ (*ff*0)] = $2^{11} \cdot 3^3 \cdot 5^1 \cdot 7^{13} \cdot 11^7 \cdot 13^{11} \cdot 17^3 \cdot 19^3 \cdot 23^1 \cdot 29^{13}$ corresponds to:

the operation of enclosing the symbol strings ***f*0** and ***ff*0** by brackets and interposing the symbol ∨ between them to give **(*f*0) ∨ (*ff*0)**.

15. **x Gen y**

For this function,

x Gen y = φ[X ∀ (Y)]

where **X** and **Y** are symbol strings of the formal system **P**, and ∀ is a symbol of the formal system **P**.

Function 8 above gives that

φ[X] * φ[∀] * φ[( ] * φ[Y] * φ[ )] = φ[X ∀ (Y)]

that is, if **x = φ[X]** and **y = φ[Y]**, then the value of **x Gen y** corresponds to:

the operation of concatenating the symbol string **X**, the symbol ∀, and the symbol string (**Y**).

i.e., creating the generalization of a formula by prefixing it with a variable and the '*For all*' symbol (though note that in this function **X** is not restricted to being a variable - it can be any symbol string of the system **P**; the definition of what constitutes a variable is given by Relation 12).

x Gen y ≡ R(x) * R(9) * E(y)

"**x Gen y** is the *generalization* of **y** by means of the *variable* **x** (assuming **x** is a *variable*)."

e.g., if **X** is a **type 1 variable**, and **x** = **φ[z_1]**, and **Y** is some formula, where **y = φ[Y]**

then **x Gen y** = $2^{23} \cdot 3^9 \cdot \ldots = 2^{\psi[z_1]} \cdot 3^{\psi[\forall]} \cdot \ldots$ corresponds to:

the operation of bracketing the formula **Y** and prefixing it by **z_1∀**

Relations 11, 12: Assertions regarding variables of the formal system

Relations 11. **n Var x** and 12. **Var(x)** correspond to the assertion that a particular symbol is a variable (reminder: we refer to a variable of the formal system as a 'single' symbol, although it is actually composed of two or more symbols).

11. **n Var x**
This is a relation, which asserts that $\mathbf{x} = \mathbf{p}^{\mathbf{n}}$, where **p** is a prime number greater than **13**.
e.g., the relation **1 Var 17** applies; **2 Var 23^2** applies; **2 Var 19** does not apply (since the exponent of **19** is not **2**).

The relation **n Var x** is an assertion regarding natural numbers **n** and **x**, and corresponds to the assertion:
the symbol string **X** is a **type n** variable of the formal system **P**
where $\mathbf{x} = \psi[\mathbf{X}]$.

Note that **n Var x** defines a correspondence to a variable of the formal system by the use of the ψ function rather than using the φ function directly. The **n** defines the type of the corresponding variable **X**.

n Var x ≡ (∃z)[13 < z ≤ x & Prim(z) & x = z^n] & n ≠ 0

"**x** is a *variable* of ***n**^th^ type*."

e.g., if **X** is the variable $\mathbf{x_3}$ of the formal system , and $\psi[\mathbf{x_3}] = \mathbf{17}$
then **3 Var 17^3** corresponds to:
the assertion that the symbol string $\mathbf{x_3}$ is a **type 3 variable**.

12. **Var(x)**
This is a relation whose definition follows directly from the previous relation (**n Var x**), i.e., it asserts that $\mathbf{x} = \mathbf{p}^{\mathbf{n}}$, where **p** is a prime number greater than **13**, and **n** > **0**.

The relation **Var(x)** is an assertion regarding a natural number **x**, and corresponds to the assertion:
the symbol string **X** is a variable of the formal system **P**
where $\mathbf{x} = \psi[\mathbf{X}]$.

Similar to the definition for **n Var x**. **Var(x)** also defines a correspondence to a variable of the formal system by the use of the ψ function ()

Var(x) ≡ (∃n)[n ≤ x & n Var x]

"**x** is a *variable*"

e.g., if **X** is the variable $\mathbf{y_2}$ of the formal system, and $\psi[\mathbf{y_2}] = \mathbf{19^2}$
then **Var 19^2** corresponds to:
the assertion that the symbol string $\mathbf{y_2}$ is a **variable** of the formal system **P**.

Relations 18, 19: Assertions regarding Type n signs of the formal system

Relations 18. $\mathbf{Typ_1'(x)}$ and 19. $\mathbf{Typ_n(x)}$ correspond to the assertion that a particular combination of symbols is a sign of the formal system.

18. $\mathbf{Typ_1'(x)}$ 19. $\mathbf{Typ_n(x)}$

$\mathbf{Typ_1'(x)}$ is not used anywhere else, its only role is in the definition of the relation $\mathbf{Typ_n(x)}$.
$\mathbf{Typ_n(x)}$ asserts that either $\mathbf{x = 2^3 \cdot 3^3 \cdot \ldots \cdot p^q}$, or $\mathbf{x = 2^{p^n}}$, where **p**, **q** and **n** must satisfy certain conditions.

$\mathbf{Typ_1'(x)}$ asserts that either:
$\mathbf{x = 2^3 \cdot 3^3 \cdot \ldots \cdot p^1}$, where **p** is some prime number; this corresponds to :
X is a symbol string of the form **0**, ***f*0**, ***ff*0**, ***fff*0**, **…**,
where $\mathbf{x = \varphi[X]}$.

or

$\mathbf{x = 2^3 \cdot 3^3 \cdot \ldots \cdot p^q}$, where **p** is some prime number and **q** is some prime number greater than 13; this corresponds to:
X is a symbol string of the form $\mathbf{x_1}$, $\mathbf{\mathit{f}x_1}$, $\mathbf{\mathit{ff}x_1}$, $\mathbf{\mathit{fff}x_1}$, **…**, or $\mathbf{y_1}$, $\mathbf{\mathit{f}y_1}$, $\mathbf{\mathit{ff}y_1}$, $\mathbf{\mathit{fff}y_1}$, **…**, etc,
where $\mathbf{x = \varphi[X]}$.

For $\mathbf{Typ_n(x)}$, for the case when **n = 1**, the case is simply
$\mathbf{Typ_1(x) = Typ_1'(x)}$

when **n > 1**, $\mathbf{Typ_n(x)}$ asserts that:
$\mathbf{x = 2^{p^n}}$, or $\mathbf{x = (2^p)^n}$, where **p** is some prime number greater than **13**; this corresponds to:
X is a **sign of type n** (which is the same as a **variable of type n**)
where $\mathbf{x = \varphi[X]}$.
See here for the definition of a **type n sign/variable**.

"$\mathbf{Typ_1'(x)}$: **x** is a *sign of first type*. $\mathbf{Typ_n(x)}$: **x** is a *sign of* $\mathbf{\mathit{n}^{th}}$ *type*."

So, $\mathbf{Typ_n(x)}$ is an assertion regarding a natural number **x**, and corresponds to the assertion:
the symbol string **X** is a **type n sign** of the formal system **P**
where $\mathbf{x = \varphi[X]}$.

Relations 20 - 23: Assertions regarding formulas of the formal system

Relations 20. **Elf(x)** and 21. **Op(x,y,z)** correspond to the assertion that a particular combination of symbols is a particular type of formula of the formal system. **FR(x)** is a relation which corresponds, though not directly by Gödel numbering, to the notion of a series of formulas. These relations are not used elsewhere; their only purpose is for the definition of **Form(x)**.

Relation 23. **Form(x)** corresponds to the assertion that a particular combination of symbols is a formula of the formal system. It is defined in terms of **Elf(x)**, **Op(x,y,z)** and **FR(x)**.

20. **Elf(x)** 21. **Op(x,y,z)** 22. **FR(x)** 23. **Form(x)**

Elf(x) is a relation that corresponds to the assertion:
the symbol string **X** is an elementary formula of the formal system **P.**
where **x** = **φ[X]**. See elementary formula.

$$\mathbf{Elf(x) \equiv (\exists y,z,n)[y,z,n \leq x \ \& \ Typ_n(y) \ \& \ Typ_{n+1}(z) \ \& \ x = z * E(y)]}$$

Op(x, y, z) is a relation that corresponds to the assertion:
the symbol string **X** of the formal system **P** is of the form
~(Y), or
(Y) ∨ (Z), or
x∀(X),
where **x** = **φ[X]**, **y** = **φ[Y]** and **z** = **φ[Z]** and **Y** and **Z** are symbol strings of the formal system **P**.
See the formulas following the section elementary formula.

$$\mathbf{Op(x, y, z) \equiv x = Neg(y) \lor x = y \ Dis \ z \lor (\exists v)[v \leq x \ \& \ Var(v) \ \& \ x = v \ Gen \ y]}$$

FR(x) is a relation which asserts that

$\mathbf{x} = \mathbf{2}^{\varphi[X1]} \cdot \mathbf{3}^{\varphi[X2]} \cdot \mathbf{5}^{\varphi[X3]} \cdot \ldots \ \mathbf{p}^{\varphi[Xn]}$.

x corresponds to a series of formulas, in that the prime factors of **x** have exponents that are the Gödel numbers of formulas. Each formula is either an elementary formula or a formula built up from elementary formulas. See here for definition of a formula of the system **P**. Note that where the function **n Gl x** occurs in the definition of **FR(x)**, it does *not* correspond to giving a single symbol of the formal system, which was was the case in the previous relations - it corresponds to giving a symbol string of the formal system **P**.

NB: This is not a direct correspondence by Gödel numbering as in the previous relations, and there is no symbol string of the formal system **P** where **x = φ[X]**.

FR(x) ≡ (n){0 < n ≤ l(x) ⇒ Elf(n Gl x) ∨ (∃p,q)[0 < p,q < n & Op(n Gl x, p Gl x, q Gl x)]} & l(x) > 0

"**x** is a series of *formulae* which is either an *elementary formula* or arises from those preceding by the operations of *negation*, *disjunction* and *generalization*."

Form(x) is a relation which asserts that

the symbol string **X** is a formula of the formal system **P**

where **x = φ[X]**, and **x** is the exponent of the largest prime **p** in a number **n**, where:

$\mathbf{n} = \mathbf{2}^{\varphi[Y1]} \cdot \mathbf{3}^{\varphi[Y2]} \cdot \mathbf{5}^{\varphi[Y3]} \cdot \ldots \ \mathbf{p}^{\varphi[X]}$

See here for definition of a formula of the system **P**.

"**x** is a *formula* (i.e. last term of a *series of formulae* n)."

---

[1] Note that these variables are in a language that is a meta-language to the formal system.

Below is a list of names used for various relations in the text, which are mostly abbreviations of German words; translations are provided below:

A	Anzahl	= number
Aeq	Aequivalenz	= equivalence
Ax	Axiom	= axiom
B	Beweis	= proof
Bew	Beweisbar	= provable
Bw	Beweisfigur	= proof-schema
Con	Conjunktion	= conjunction
Dis	Disjunktion	= disjunction
E	Einklammern	= include in brackets
Elf	Elementarformel	= elementary formula
Ex	Existenz	= existence
Fl	unmittelbare Folge	= immediate consequence
Flg	Folgerungsmenge	= set of consequences
Form	Formel	= formula
Fr	frei	= free
FR	Reihe von Formeln	= series of formulae
Geb	gebunden	= bound
Gen	Generalisation	= generalization
Gl	Glied	= term
Imp	Implikation	= implication
l	Lange	= length
Neg	Negation	= negation
Op	Operation	= operation
Pr	Primzahl	= prime number
Prim	Primzahl	= prime number
R	Zahlenreihe	= number series
Sb	Substitution	= substitution
St	Stelle	= place
Su	Substitution	= substitution
Th	Typenerhohung	= type-lift
Typ	Typ	= type
Var	Variable	= variable
Wid	Widerspruchsfreiheit	= consistency
Z	Zahlzeichen	= number-symbol

A Step by Step Guide to Gödel's Incompleteness Proof

Gödel's Relations 24 - 46 of Natural Numbers

The number-theoretic relations are now becoming more complex as they correspond to more complex statements about formulas of the formal system **P**. It is not intended to cover every detail of these relations, but rather to concentrate on the main points and the thrust of the argument. It would be easy to get bogged down in such details and then fail to see the wood for the trees. As noted on the previous page, the names of the relations are mainly abbreviations of German words, see notes at the foot of this page for the words and the English translations.

Relations 24-26: Assertions regarding variables of the formal system

These define number-theoretic relations that correspond to assertions as to whether a symbol is a free variable or a bound variable within a given symbol string.

24. **v Geb n,x**
This is an assertion regarding the natural numbers **v**, **n** and **x**, and it corresponds to the assertion:

> at the $\mathbf{n^{th}}$ symbol in the formula **X**, if **V** is a variable, and if it were to be at that position, it would be a bound variable

provided that **v = ψ[V]** and **x = φ[X]**.

Suppose we have a formula of the form **X** = **A (**$\mathbf{x_1}$**∀ B) C**, where **A**, **B** and **C** are symbol strings of the formal system, and where $\mathbf{x_1}$ is a variable, and ∀ is the quantifier symbol for '*for all*'. Then the quantifier on the variable $\mathbf{x_1}$ applies everywhere in the string **B**, but that quantifier does not apply to the string **C**. So the relation is asserting that the quantifier applies throughout the string B, ***regardless*** of where the varaible V might be in that string.

v Geb n,x asserts that if **v = ψ[V]**, where **V** is a variable of the formal system, and **x = φ[X]**, where **X** is a formula, then if the $\mathbf{n^{th}}$ prime number in **x** is one of **p**, **q**, **r**, **s**, **…**, then the corresponding variable **V** is bound (by the quantifier ∀) at the corresponding positions in the formula **X**.

> "The *variable* **v** is *bound* at the $\mathbf{n^{th}}$ place in **x**."

25. **v Fr n,x**
This is a relation. **v Fr n,x** is an assertion regarding the natural numbers **v**, **n** and **x**, and

v Fr n,x corresponds to the assertion:

> **V** is a variable, and it occurs as the $\mathbf{n^{th}}$ symbol in the formula **X**, and it is free at that position

provided that **v = ψ[V]** and **x = φ[X]**

> "The *variable* **v** is *free* at the $\mathbf{n^{th}}$ place in **x**."

26. **v Fr x**
This is a relation. **v Fr x** is an assertion regarding the natural numbers **v** and **x**, and
v Fr x corresponds to the assertion:
V is a variable, and it occurs as a free variable in the formula **X**
provided that **v = ψ[V]** and **x = φ[X]**.

"**v** occurs in **x** as a *free variable*."

Functions 27-31: Defining numbers that correspond to substitution in the formal system

The functions 27-30 lead up the function 31 which corresponds to the concept of the substitution of a free variable by a symbol or symbol string of the formal system.

27. **Su x(n|y)** 28. **k St v,x** 29. **A(v,x)** 30. **Sb_k(x v|y)** 31. **Sb(x v|y)**
Sb(x v|y) is a function which is defined in terms of **Su x(n|y)**, **k St v,x**, **A(v,x)** and **Sb_k(x v|y)**; these functions are not used anywhere else.

if **x = φ[X]**, **v = ψ[V]** and **y = φ[Y]**, then **Sb(x v|y)** corresponds to:
the operation of substituting the symbol **V** where it occurs as a free variable within the symbol string **X**, by the symbol string **Y**.

As an example, suppose that we have a formula **X** with only one free variable **x_1** at only one position in the formula. Then the Gödel number corresponding to that formula **X** will be like this :
$\varphi[X] = 2^a \cdot 3^b \cdot 5^c \ldots \cdot p^{17} \cdot j^u \cdot k^w \cdot l^y \cdot m^z \cdot \ldots$
where **p**, **j**, **k**, **l**, **m**, **q**, **r**, **s** and **t** are all prime numbers in order of size, and the values of the exponents **a**, **b**, **c**, **…**, **17**, **u**, **w**, **y**, **z**, **…** are given by the function **ψ** on the individual symbols/variables of the formula **X**, where **17 = ψ[x_1]**.

If the symbol string which will be substituted is say, ***ffff*0**, then
$\varphi[ffff0] = 2^3 \cdot 3^3 \cdot 5^3 \cdot 7^3 \cdot 11^1$
$Sb(x\ v|y) = 2^a \cdot 3^b \cdot 5^c \ldots \quad \cdot p^3 \cdot j^3 \cdot k^3 \cdot l^3 \cdot m^1 \cdot q^u \cdot r^w \cdot s^y \cdot t^z \cdot \ldots$
This corresponds to the substitution of the **type 1 variable x_1** by the **type 1 sign *ffff*0**

Gödel states that the function **Sb(x v|y)** corresponds to the concept **Subst a(v|b)**, as defined in the definition of the system **P**. Note that any Gödel number may be substituted for the variable **y**, so that the function **Sb** corresponds to the substitution of a free variable by *any symbol string* of the formal system. **Subst** defines that **b** must be of the same sign as the variable **v**. Hence where Sb is used in the following functions/relations, there must be the added stipulation to that effect.

"**Sb(x v|y)** is the concept **Subst a(v|b)**"

"By **Subst a(v|b)** (where **a** stands for a formula, **v** a variable and **b** a sign of the same type as **v**) we understand the formula derived from **a**, when we replace **v** in it, wherever it is free, by **b**. Where **v** does not occur in **a** as a free variable, we must put **Subst a(v|b) = a**. Note that '**Subst**' is a sign belonging to metamathematics."

Note that, depending on what version of the translation you are using, **Sb** may be represented in this format:

$$Sb\left(x \begin{array}{c} v \\ y \end{array}\right)$$

which is the format using in Gödel's original paper.

Relations/functions 32-42: Assertions as to which numbers correspond to the axioms of the formal system

Functions 32 and 33 define some axioms of the formal system. Relations 34-42 inclusive are relations that use the previously defined relations/functions to define which Gödel numbers correspond to the axioms of the formal system.

32. **x Imp y, x Con y, x Aeq y, v Ex y**
These define the logical equivalent of '*implies*', '*and*', '*equivalence*', and '*there exists*' (see also the axioms of the system P)

33. **n Th x**
Given that **x** is a Gödel number corresponding to a formula, then **n Th x** is a function that gives the Gödel number of the formula that is the $\mathbf{n}^{\mathbf{th}}$ type-lift of the formula **x** (see Type-lift).

34. **Z–Ax(x)**
These relations assert that **x** is the Gödel number of one of the Axioms **I.1-3**. Gödel asserts that there are numbers that correspond by Gödel numbering to each of these three axioms; he does this rather than give in detail how these numbers could be defined, but clearly these numbers could be calculated from the axioms by obtaining the equivalent formulation in the symbols of the formal system, and then applying the Gödel numbering function.

> "To the axioms **I**, **1** to **3**, there correspond three determinate numbers, which we denote by $\mathbf{z_1}$, $\mathbf{z_2}$, $\mathbf{z_3}$."

35. **A_1-Ax(x)**, **A_2-Ax(x)**, **A_3-Ax(x)**, **A_4-Ax(x)**   36. **A-Ax(x)**
Relation 35, **A_1-Ax(x)** defines that **x** is a Gödel number that corresponds to an axiom defined by Axiom Schema **II.1**. Similarly for **A_2-Ax(x)**, **A_3-Ax(x)** and **A_4-Ax(x)** for axioms defined by the Axiom Schemas **II.2-4**.

Relation 36, **A-Ax(x)** defines that **x** is a Gödel number that corresponds to one of the axioms given by the Axiom Schemas **II.1-4**

37. **Q(z,y,v)**

Q(z,y,v) ≡ ~(∃n,m,w)[n ≤ l(y) ∧ m ≤ l(z) ∧ w ≤ z ∧ w = m Gl z ∧ w Geb n,y ∧ v Fr n,y]

This relation is only used in the definition of relation 38 and not elsewhere. It corresponds to the assertion:

> the symbol string **Z** does not have any variable bound at any position which is not under the influence of a quantifier on the variable **V**."

provided that **z = φ[Z]**, **y = φ[Y]**, **v = ψ[V]**.

"**z** contains no *variable bound* in **y** at a position where **v** is free."

38. **L_1-Ax(x)**

This relation asserts that **X** is an axiom given by the Axiom Schema **III.1**, where **x = φ[X]**.

39. **L_2-Ax(x)**

This relation asserts that **X** is an axiom given by the Axiom Schema **III.1**, where **x = φ[X]**.

40. **R-Ax(x)**

This relation asserts that **X** is an axiom given by the Axiom Schema **IV.1**, where **x = φ[X]**.

41. **M-Ax(x)**

This relation asserts that **x** is a number that corresponds by Gödel numbering to either the base axiom of the Axiom Schema **V.1**, or to a type-lift of the base axiom. As for relation 34, **Z–Ax(x)**, Gödel asserts that there is a number that corresponds to the base axiom of Axiom Schema **V.1**, rather than defining it in detail; this number (and the numbers for type-lifts) could be defined from the axiom by obtaining the equivalent formulation in the symbols of the formal system,and then applying the Gödel numbering function.

"To the axiom **V**, **1** there corresponds a determinate number $\mathbf{z_4}$."

42. **Ax(x)**

This relation asserts that **x** is a number that corresponds by Gödel numbering to an axiom of the formal system **P**.

Relations 43-46:
Proofs in the formal system

The relations 43-46 deal with defining the number-theoretic relations that correspond to the concepts of the rules of inference of the system, the concept of a proof-schema, and the concept of a formula being provable in the system.

43. **Fl(x y z)**
This is an assertion regarding the natural numbers **x**, **y** and **z**, and it corresponds to the assertion:
the formula **X** is derived by the rules of inference from the formulas **Y** and **Z**
where **x** = **φ[X]**, **y** = **φ[Y]**, **z** = **φ[Z]** and **X**, **Y** and **Z** are symbol strings of the formal system **P**.

"**x** is an *immediate consequence* of **y** and **z**"

44. **Bw(x)**
This relation is defined in terms of the previous relation **Fl(x y z)**; it is an assertion that

$$\mathbf{x} = 2^{\varphi[X1]} \cdot 3^{\varphi[X2]} \cdot 5^{\varphi[X3]} \cdot \ldots \; p^{\varphi[Xn]} .$$

x corresponds to a series of formulas, in that the prime factors of **x** have exponents that are the Gödel numbers of formulas. Each formula is either an axiom or a formula given by the rules of inference of th system applied to axioms or proven formulas.

NB: As for relation 22, this is *not* a direct correspondence by Gödel numbering and there is no symbol string of the formal system **P** where **x** = **φ[X]**.

"**x** is a *proof-schema* (a finite *series of formulae*, of which each is either an *axiom* or an *immediate consequence* of two previous ones)"

45. **x B y**
This is an assertion regarding the natural numbers **x** and **y** and it corresponds to the assertion that
the symbol string **Y** is a formula of the formal system **P** and there is a proof-schema for **Y** that corresponds to the number **x** by an appropriate relation.
where **y** = **φ[Y]**.
y corresponds to the exponent of the largest prime **p** in the number **x**, where:

$$\mathbf{x} = 2^{\varphi[X1]} \cdot 3^{\varphi[X2]} \cdot 5^{\varphi[X3]} \cdot \ldots \; p^{\varphi[Y]}$$

Note that, as for relation 44 above, there is no symbol string of the formal system **P** where **x** = **φ[X]**.

"**x** is a *proof* of the *formula* **y**"

46. **Bew(x)**
This is an assertion regarding the natural numbers **x** and **y** and it corresponds to the assertion:
X is a provable formula of the system **P**
i.e., there exists a proof-schema (a series of formulas) that is a proof of the formula **X**
where **x** = **φ[X]**, and there exists a number **w** such that **x** is the exponent of the largest prime **p** in the number **w**, where:

$$\mathbf{w} = 2^{\varphi[Y1]} \cdot 3^{\varphi[Y2]} \cdot 5^{\varphi[Y3]} \cdot \ldots \; p^{\varphi[X]}$$

"**x** is a *provable formula*. [**Bew(x)** is the only one of the concepts 1-46 of which it cannot be asserted that it is recursive.]"

Below is a list of names used for various relations in the text, which are mostly abbreviations of German words; translations are provided below:

A	Anzahl	= number
Aeq	Aequivalenz	= equivalence
Ax	Axiom	= axiom
B	Beweis	= proof
Bew	Beweisbar	= provable
Bw	Beweisfigur	= proof-schema
Con	Conjunktion	= conjunction
Dis	Disjunktion	= disjunction
E	Einklammern	= include in brackets
Elf	Elementarformel	= elementary formula
Ex	Existenz	= existence
Fl	unmittelbare Folge	= immediate consequence
Flg	Folgerungsmenge	= set of consequences
Form	Formel	= formula
Fr	frei	= free
FR	Reihe von Formeln	= series of formulae
Geb	gebunden	= bound
Gen	Generalisation	= generalization
Gl	Glied	= term
Imp	Implikation	= implication
l	Lange	= length
Neg	Negation	= negation
Op	Operation	= operation
Pr	Primzahl	= prime number
Prim	Primzahl	= prime number
R	Zahlenreihe	= number series
Sb	Substitution	= substitution
St	Stelle	= place
Su	Substitution	= substitution
Th	Typenerhohung	= type-lift
Typ	Typ	= type
Var	Variable	= variable
Wid	Widerspruchsfreiheit	= consistency
Z	Zahlzeichen	= number-symbol

A Step by Step Guide to Gödel's Incompleteness Proof

Gödel's Proposition V

This is a proposition about primitive recursive relations that can have any number of free variables. The formulation of the proposition as Gödel presented it is given below. For our purposes It is easier to follow the argument if we first consider the proposition as applied to a relation with only one free variable. One can then extend the argument to relations with any number of free variables. The proposition for relations with only one free variable is:

For every primitive recursive relation **R** with one free variable **x** there corresponds a number **r** and a number **u** such that for all **x,** either

(3) **R(x)** ⇒ **Bew{Sb[r u | Z(x)]}** or

(4) ~ **R(x)** ⇒ **Bew{Neg Sb[r u | Z(x)]}**

The proposition asserts that, without having to consider any 'meaning' of the formulas of the system **P**, we may assert the following:

I. for every primitive recursive relation **R(x)** there is a corresponding Gödel number **r**, and
II. if the relation **R(x)** holds for some value of **x**, then the relation Bew{Sb[r u | Z(x)]} holds, and
III. if the relation ~**R(x)** holds for some value of **x**, then the relation Bew{Neg Sb[r u | Z(x)]} holds

Unfortunately, Gödel declined to actually furnish a proof of the proposition, giving only an outline proof and claiming that it "*offers no difficulties of principle*". From the outline that Gödel sketched the arguments that support the three main assertions above are as follows:

Assertion (**I**): The argument supporting the assertion (**I**) is as follows:

A primitive recursive number-theoretic relation may be defined in terms of one or more previously defined primitive recursive number-theoretic relations or functions until the simplest such relations and functions are obtained. Such relations/functions are simple expressions that can be expressed using

a. the symbols **0**, ***f*** , ∨, ∀, ~, **(**, **)**, and/or
b. the symbols =, ≠, <, >, ≤, ≥, **ε**, +, /, ∃, **&**, ≡, ⇒, **[**, **]**, {, }, ~, and/or
c. exponentiation (raising to the power of, e.g., $\mathbf{x}^{\mathbf{y}}$), and/or
d. various symbols for variables.

The symbols of (a) are the symbols of the formal system **P.**
It can be shown that expressions using the symbols of (b) can be formulated using the basic symbols (a) of the system **P**. Similarly, exponentiation (c) can also be formulated using the basic symbols (a) of the system **P**.

The variables used can be replaced by symbols for variables in the formal system **P**. From the above, given any primitive recursive number-theoretic relation, it can be seen that it can be formulated using the symbols of that formal system **P**. This is a purely mechanical procedure, since it only requires that a set of rules for such formulation is followed, and no meaning is required to be given to the symbols.

That means that, given any primitive recursive number-theoretic relation with one free variable **R(x)**, a corresponding formula **F(x_1)** of the system **P** with one free variable can be formulated from that relation. In Gödel's terminology, this formula is a **1-place relation-sign** (also called a **class-sign**).

The Gödel numbering function **φ** can then be applied to that formula **F(x_1)**, which will give a number **r** that corresponds to the relation **R(x)** (the Gödel number of the formula, i.e., r = φ[F(x_1)]). Gödel calls this number a *1-place relation-sign* or a *class-sign* (note the distinction between the word in plain text and in italics).

Assertions (**II**) and (**III**). The argument supporting these assertions is as follows:

If the relation **R(x)** holds for some value of **x**, then, since the relation is primitive recursive, then it is possible to calculate, according to the axioms of the system **P** whether the relation holds or not, and so there will be a proof-schema in the system **P** that proves the formula that results from the substitution of the variable **x_1** in **F(x_1)** by that value of **x**.

If **F′** is the formula obtained when the free variable **u** of the formula **F** is substituted by the value **x**, i.e., F′ = Subst F(x_1/x) then Sb[r u | Z(x)] is the Gödel number of that formula **F′**, i.e., **φ[F′]**.

And since the relation **Bew** corresponds to the notion of "*Provable in the system* **P**", then Bew{Sb[r u | Z(x)]} corresponds to the fact that the formula **F′** is provable in the system **P**. Similarly Bew{Neg Sb[r u | Z(x)]} corresponds to the fact that the negation of the formula **F′** is provable in the system **P**.

So, at first sight it appears that Gödel's Proposition V is proved. But this result is a lesson in how the apparently obvious can conceal the not so obvious.

However…

The argument crucially depends on the assumption that the primitive recursive function **Z(x)** is equal/equivalent to the Gödel numbering function **φ(x)**, that is, the argument depends on the assertion that:

Z(x) = φ(x)

or, in other words:
"For every value of the domain of **x***, the value of the primitive recursive number-theoretic function* **Z(x)** *is identical to the value of the meta-language function* **φ(x)***."*

But, as noted previously for the **Z** function, on the left-hand side the variable **x** has the domain of natural numbers in any valid format of whatever number system is being used for number-theoretic relations, whereas on the right-hand side the variable **x** has the domain of symbol strings of the formal system. On the left-hand side of the equation, we can have, for example, for the value of **x**, **8** or *ffffffff*0 or **3 + 5** or **64 ÷ 2^3** or **9 - 1**, etc, whereas on the right-hand side, we can *only* have φ(*ffffffff*0).So, for example:

Z(8) = φ(*ffffffff*0), but
Z(*ffffffff*0) ≠ φ(8)

and

Z(8) = Z(3 + 5) = Z(64 ÷ 2^3) = Z(9-1) = Z(*ffffffff*0) = φ(*ffffffff*0), but
Z(*ffffffff*0) = Z(9-1) = Z(64 ÷ 2^3) = Z(3 + 5) = Z(8) ≠ φ(8).

And while the value **~*fff*0** is a perfectly valid value of the domain of **x** for the function **φ**, and gives a valid expression of the meta-language as **φ(~*fff*0)**, the same value of **x** is not a value value of the domain of **x** for the number-theoretic function **Z**, and **Z(~*fff*0)** is not a valid number-theoretic expression. i.e.:

Z(~*fff*0) ≠ φ(~*fff*0).

The only way to avoid this contradiction is to assert that the domain of the **x** on both sides be defined as being restricted to symbols strings of the form **0, *f*0, *ff*0, *ff*0, …** . But while that might superficially appear to be a fix, this ad hoc restriction of the domains should be setting off alarm bells, and warning us that we should be analyzing what is going on in a very careful manner. The entire proof is purportedly a proof that is stated in a meta-language where the statements of the meta-language are statements about a sub-language, and for this reason we need to be very careful that the distinctions between different levels of language are always maintained. But, amazingly, at this point, many logicians proceed by completely ignoring this crucial detail.

Consider the first part of Gödel's proposition, "*For every primitive recursive relation* **R**". Here **R** is a variable of the meta-language and which has the domain of number-theoretic primitive recursive relations - which means that every member of that domain is an ***object*** in the meta-language. and that means that number-theoretic primitive recursive relations are objects of the meta-language - in precisely the same way as formulas of the formal system **P** are objects in the meta-language.

And in the same way that variables of the formal system **P** are simply objects in the meta-language, the variables of primitive recursive relations are necessarily also objects of the meta-language, and they cannot be variables of the meta-language.

Otherwise we would have a nonsensical situation where variables of a sub-language would be at the same time variables of the meta-language. This is nonsensical because it is contradictory. Any variable of a sub-language can ***always*** be a member of the domain of a variable of the meta-language - since the meta-language can always refer to entities of the sub-language either in specific or in general terms. So that if a variable of the sub-language could be at the same time a variable of the meta-language, it would at the same time be both a variable of the meta-language and a specific object of the meta-language, i.e., not a variable of the meta-language - and that is a contradiction. This is covered in more detail in the paper on Gödel's proof, see http://jamesrmeyer.com/pdfs/FFGIT_Meyer.pdf.

Bearing this in mind, if we now look again at Gödel's Proposition V, we see that Gödel assumes on the one hand that **x** is a variable of the meta-language, but on the other hand, by referring to primitive recursive relations as objects of the meta-language, and by referring to variables of primitive recursive relations as objects of the meta-language, he defines **x** as a variable of a sub-language of number-theoretic relations, and so it cannot be a variable of the meta-language.

Of course, there are many cases where statements that are ostensibly similar to Gödel's Proposition are used in mathematics and no problems arise - but that is because in the vast majority of such cases the sub-language (usually number-theoretic relations) is the only matter under consideration and there is no introduction of the meta-language into the sub-language, and no confusion of meta-language and sub-language. But simply because a method does not give rise to logical anomalies in some situations, one cannot extrapolate from that to assume that the method can never give rise to logical anomalies. It is a universally accepted principle of mathematics that one can never assume the general case from a finite number of instances; one must prove the general case by other means.

And in the case of Gödel's Proposition V, the confusion of meta-language and sub-language does become of overriding importance, since the prior assumption of the equality/equivalence of the primitive recursive function **Z(x)** and the Gödel numbering function **φ(x)** is logically impossible in conjunction with Gödel's Proposition V. This is so since the Gödel numbering function **φ(x)** is a function of the meta-language, whereas the primitive recursive function **Z(x)** is necessarily a function of the sub-language of primitive recursive relations - from the statement of Proposition V it must be in a sub-language. The quantifier that is applied to primitive recursive relations by that proposition means that primitive recursive relations and the variables of primitive recursive relations are simply objects in the meta-language. This means that in the context of Proposition V, an assertion such as:

$$Z(x) = \varphi(x)$$

is logically absurd, since the **x** on the left-hand side must be an object of the meta-language and not a variable of the meta-language, whereas the **x** on the right-hand side must be a variable of the meta-language.

Why logicians studiously look the other way and turn a blind eye to this anomaly, which is patently obvious once it is pointed out, is a mystery.

When correctly analyzed, Gödel's Proposition V is seen to be a confusion of meta-language and sub-language, and that it is only by use of this confusion that the proof can appear to proceed. For this reason no further analysis of the remainder of Gödel's proof will be given here, as such further analysis is utterly pointless when it is seen that the proof contains a fundamental flaw in Proposition V. For a more detailed analysis, see the paper on Gödel's proof at see http://jamesrmeyer.com/pdfs/FFGIT_Meyer.pdf

Gödel's statement of Proposition V

The following proposition is an exact expression of a fact which can be vaguely formulated in this way: every recursive relation is definable in the system **P** (interpreted as to content), regardless of what interpretation is given to the formulae of **P**:

Proposition V
To every recursive relation $\mathbf{R(x_1 \dots x_n)}$ there corresponds an n-place *relation-sign* **r** (with the *free variables* $\mathbf{u_1}$,$\mathbf{u_2}$, …$\mathbf{u_n}$)[38] such that for every n-tuple of numbers $\mathbf{(x_1 \dots x_n)}$ the following hold:

$\mathbf{R(x_1 \dots x_n)} \Rightarrow \mathbf{Bew\{Sb[r\ (u_1 \dots u_n)\ |(Z(x_1) \dots Z(x_n)]\}}$ (3)
$\mathbf{\sim R(x_1 \dots x_n)} \Rightarrow \mathbf{Bew\{Neg\ Sb[r\ (u_1 \dots u_n)\ |(Z(x_1) \dots Z(x_n)]\}}$ (4)

We content ourselves here with indicating the proof of this proposition in outline, since it offers no difficulties of principle and is somewhat involved[39]. We prove the proposition for all relations $\mathbf{R(x_1 \dots x_n)}$ of the form: $\mathbf{x_1 = \varphi(x_2 \dots x_n)}$[40] (where $\boldsymbol{\varphi}$ is a recursive function) and apply mathematical induction on the degree of $\boldsymbol{\varphi}$. For functions of the first degree (i.e. constants and the function **x+1**) the proposition is trivial. Let $\boldsymbol{\varphi}$ then be of degree m. It derives from functions of lower degree **φ1 … φk** by the operations of substitution or recursive definition. Since, by the inductive assumption, everything is already proved for **φ1 … φk**, there exist corresponding *relation-signs* **r1 … rk** such that (3) and (4) hold. The processes of definition whereby **φ** is derived from **φ1 … φk** (substitution and recursive definition) can all be formally mapped in the system **P**. If this is done, we obtain from **r1 … rk** a new *relation-sign* **r**[41], for which we can readily prove the validity of (3) and (4) by use of the inductive assumption. A *relation-sign* **r**, assigned in this fashion to a recursive relation[42] will be called recursive.

Gödel's footnote 38: *The variables* $\mathbf{u_1 \dots u_n}$ *could be arbitrarily allotted. There is always, e.g., an* **r** *with the free variables* **17**, **19**, **23** … *etc., for which* (3) *and* (4) *hold.*
Gödel's footnote 39: *Proposition V naturally is based on the fact that for any recursive relation* ***R****, it is decidable, for every n-tuple of numbers,* ***from the axioms of the system P****, whether the relation* ***R*** *holds or not.*
Gödel's footnote 40: *From this there follows immediately its validity for every recursive relation, since any such relation is equivalent to* $\mathbf{0 = \varphi(x_1 \dots x_n)}$*, where* **φ** *is recursive.*
Gödel's footnote 41: *In the precise development of this proof,* **r** *is naturally defined, not by the roundabout route of indicating its content, but by its purely formal constitution*
Gödel's footnote 42: *Which thus, as regards content, expresses the existence of this relation.*
