GÖDEL'S INCOMPLETENESS THEOREM

Jorge del Castillo Tierz

Final degree thesis in Mathematics

University of Zaragoza

June 2016

Prologue

'Oh, you can't help that,' said the Cat:
'we're all mad here. I'm mad. You're mad.'
'How do you know I'm mad?' said Alice.
'You must be,' said the Cat, 'or you wouldn't have come here.'
Lewis Carroll, Alice's Adventures in Wonderland

From Aristotelian Logic to Logic nowadays, an incredibly huge progress has taken place. One of the first in appear and the simplest is what we call now "Propositional Logic" and we are going to review its most important aspects in the first section. Later on, during the 19th and 20th centuries, logic was rediscovered and a mathematical structure was adopted. But the astonishing approach was the one Gödel and Tarski developed: a metalogic, logic which speaks about logic.

The main result Gödel proved was that if an axiomatic system for arithmetic (Peano Arithmetic) is consistent, i.e. it does not lead to any contradictions, and its axioms are recursive, then it will be incomplete. What does it mean to be "incomplete"? To put it simply: to have at least one sentence which is true but not provable in the system or a false sentence that cannot be proven to be false. This sentence is usually called a "Gödel sentence".

This is related to the liar paradox, which consists of *j*ust one statement: "This sentence is false". By writing "not provable" instead of "false" we get what we called a Gödel sentence, this time without a contradiction. So, how can we write a sentence that states its own unprovability in the system? This is what we are going to study in this thesis.

The *d*iscovery of Gödel pointed out the limitations of axiomatic systems. A mathematical system which contains enough arithmetic will never be able to prove all true sentences without proving falsities as well.

This thesis is organised in the following way:

1. A first chapter on Propositional Logic which will serve as a review for those who already are familiar with Logic and as a brief introduction to those who are not.

2. A second chapter on Predicate Logic and first-order systems, more complex than the ones in the first chapter. Here, we are going to deal with objects, functions and relations.

3. In the third chapter, we are going to develop a first-order system in which Mathematics can be expressed, more specifically, P*e*ano Arithmetic.

4. Chapters 4 and 5 will take a vital part in the proof of the theorem since the notions of recursiveness, expressibility, representability and Gödel numbers defined then are what is going to enab*l*e us to see the problem from another point of view.

5. Finally, the last chapter is devoted to the theorem itself and a further discussion about Gödel's Second Theorem and Church's Thesis.

Resumen en español

En este trabajo de fin de grado, vamos a estudiar uno de los teoremas más importantes del campo de la Lógica, el Teorema de Incompletitud de Gödel.

En pocas palabras, lo que dice es que si un sistema matemático contiene suficiente aspectos de la Aritmética y es consistente, es decir, que no se pueda probar a la vez una cosa y su contrario, entonces, por desgracia, ese sistema siempre va a ser incompleto. Esto quiere decir que hay alguna fórmula que, a pesar de ser verdadera, no puede ser probada. No porque la demostración sea extremadamente complicada, sino porque esa demostración, simplemente, no existe.

Se ha estructurado este trabajo siguiendo un modelo constructivo, partiendo de la Lógica más básica hasta llegar al teorema. En los primeros capítulos, se han obviado las demostraciones para no hacer demasiado pesada la lectura y siempre en pos de la claridad.

La disposición del trabajo es la siguiente:

1. El primer capítulo trata sobre la Lógica Proposicional, la más básica y sencilla. En este capítulo, presentaremos a través de sus tablas de verdad los conectores lógicos más usuales, como la conjunción, la disyunción o la negación. Definiremos un sistema a partir de unos axiomas lógicos. Para terminar el capítulo, se mencionará la completitud de ese sistema formal, que nos asegurará que podemos encontrar todas las tautologías o verdades del sistema en forma de teoremas, a diferencia de lo que ocurre con otras Lógicas que veremos en capítulos posteriores.

2. Un escalón más arriba se encuentra la Logica de primer orden, o Lógica de predicados. En ella, aparecen objetos (constantes, variables,...) que podemos cuantificar, además de funciones y relaciones. Es una Lógica mucho más potente que la del capítulo anterior, ya que prácticamente, puede formalizar todas las Matemáticas. La noción de verdad aquí es un poco distinta, dado que es necesaria una interpretación. Una misma fórmula puede ser verdadera bajo una interpretación, pero falsa bajo otra. También partiremos de unos axiomas para definir un sistema formal y se verá que éste es completo, como pasaba con el sistema definido en el capítulo anterior.

3. En el tercer capítulo, ampliaremos nuestro sistema lógico a uno que formalice la Aritmética de Peano. Para ello, necesitaremos definir la igualdad y sus axiomas y también la suma y el producto. Así, construiremos un sistema $\mathscr{S}$ diseñado para este fin.

4. El capítulo 4 está dedicado a las funciones recursivas, la expresabilidad y la representabilidad. Estos tres conceptos se probará más adelante que están muy relacionados entre sí. Además, cumplirán un papel clave en el desarrollo de la demostración del Teorema de Incompletitud.

5. En el capítulo 5, veremos una de las ideas más brillantes de Gödel: expresar las fórmulas de la Lógica en forma de números. Este cambio de perspectiva será el que nos permitirá abordar

el problema más fácilmente. Relacionaremos a su vez esto con la recursividad definida en el capítulo anterior y daremos ejemplos de funciones y relaciones que quedan definidas a partir de la numeración de Gödel y que además son recursivas. Precisamente, serán algunas de estas funciones y relaciones las que utilizaremos a la hora de demostrar el Teorema de Incompletitud.

6. Por último, en el capítulo 6 nos encontraremos el Teorema de Incompletitud de Gödel. Para entonces, ya tendremos casi todas las herramientas necesarias que nos permitirán comprender la demostración. Tan solo nos hará falta demostrar el teorema del punto fijo y definir el concepto de ω-consistencia. A continuación, enunciaremos el Segundo Teorema de Gödel, que dice que la Aritmética de Peano no puede demostrar su propia consistencia. Para terminar, se hablará de la tesis de Church y lo que ella implica, que no existe ningún procedimiento mecánico para determinar si una fórmula es un teorema de la Aritmética o no.

Contents

Chapter 1

Taking the first steps: Propositional Logic

Logic is something we use on a daily basis and we do not even think about it.

- If Charles goes to the cinema, he will see the movie about the two dinosaur*s*.
- Charles is at the cinema or he is watching a romantic movie at home.
- Therefore, Char*l*es is either seeing the movie about the two dinosaurs or watching a romantic one at home.

Moreover, we can also say:

- If a borogove is crying, it is mimsy.
- A borogove is crying or it does not exist.
- Therefore, a borogove is mimsy or it *d*oes not exist.

Both share the same structure and we can apply the same reasoning without even having to know what “mimsy” or a “borogove” is. This is the magic of Logic. We deal with statements, not with their meanings, and we can infer rules of more general applicability. Regardless of their meaning, these two examples have the structure:

- If A, then B.
- A or C.
- Therefore, B or C.

Every time we see a structure like this, we can derive the conclusion, no matter what A, B or C mean. Propositional Logic studies which are the correct patterns a*n*d which are not.

We have been talking about propositions and statements. What are they? They are sentences, or situations of the world that can be true or false. Fro*m* now on, we will use 1 to represent the truth valu*e* “true” and 0 for “false”.

We can combine propositions in order to get more complex ones. For instance, ¬p stands for “not p”. It is false when p is true and true when p is false.

p	¬ p
1	0
0	1

The conjunction of two statements, p∧q, is true whenever p AND q are both true, and is false if at least one of them is false.

p	q	$p \wedge$q
1	1	1
1	0	0
0	1	0
0	0	0

The disjunction of two statements, p∨q, is true if at least one of p and q is true, and is false only if both are false.

p	q	$p \vee$q
1	1	1
1	0	1
0	1	1
0	0	0

One of the most important connectives is the conditional, p→q, which stands for “if p then q”. It is true if and only if the antecedent p is false or the consequent q is true.

p	q	$p \rightarrow$q
1	1	1
1	0	0
0	1	1
0	0	1

Finally we may denote by p↔q the statement “if p, then q AND if q, then p”, that is, “p if and only if q”.

p	q	$p \leftrightarrow$q
1	1	1
1	0	0
0	1	0
0	0	1

Definition. i) An atomic proposition is a letter which stands for an arbitrary and unspecified simple statement.

ii) A propositional formula is an expression defined recursively as:

a) Any atomic proposition is a propositional formula.

b) If $\mathscr{A}$ and $\mathscr{B}$ are two propositional formulas, then $(\neg\mathscr{A})$, $(\mathscr{A} \vee \mathscr{B})$, $(\mathscr{A} \wedge \mathscr{B})$ and $(\mathscr{A} \rightarrow \mathscr{B})$ are also propositional formulas.

An assignment is a mapping that assigns values in {0,1} to every atomic proposition in a propositional formula. As we said before, 1 stands for “true” and 0 for “false”. For $p \vee q$, an assignment could be (0,1), that is, p takes value 0 and q takes value 1. In this case, by the truth tables from before, the propositional formula would be true, since it has value 1.

Definition. A propositional formula is said to be satisfiable if there exists an assignment of truth values in which the formula takes truth value 1.

A formula is said to be a tautology if for every assignment it takes value 1.

A formula is said to be a contradiction or unsatisfiable if there is no assignment in which the formula takes value 1.

For example, p∨q is satisfiable, as we have seen. The law of excluded middle is a tautology: p∨¬p. And p∧¬p is unsatisfiable.

It is clear that a $\mathscr{A}$ is a contradiction if and only if $\neg\mathscr{A}$ is a tautology.

Let's define our logic system L by the following axioms:

Definition. The axioms of L are:
(L1) $(\mathscr{A} \to (\mathscr{B} \to \mathscr{A}))$
(L2) $((\mathscr{A} \to (\mathscr{B} \to \mathscr{C})) \to ((\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{C})))$
(L3) $(((\neg\mathscr{A}) \to (\neg\mathscr{B})) \to (\mathscr{B} \to \mathscr{A}))$
for any formulas $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{C}$.

From a set of axioms, we can prove new formulas using the inference rule: Modus Ponens. This rule allows us to derive the consequent whenever we have a conditional and its antecedent:

If $\mathscr{A} \to \mathscr{B}$ and $\mathscr{A}$, then $\mathscr{B}$

A *proof* of a propositional formula $\mathscr{A}$ in the logic system is a sequence of formulas ending in $\mathscr{A}$ such that each one of them is either an axiom or a direct derivation by Modus Ponens (MP) from two previous formulas in the proof. A theorem is a formula that is provable, i.e., the last line of a proof.

For example, let's give a proof the theorem $\mathscr{A} \to \mathscr{A}$.

1. $\mathscr{A} \to (\mathscr{A} \to \mathscr{A})$ (Axiom L1)
2. $\mathscr{A} \to ((\mathscr{A} \to \mathscr{A}) \to \mathscr{A})$ (Axiom L1)
3. $(\mathscr{A} \to ((\mathscr{A} \to \mathscr{A}) \to \mathscr{A})) \to ((\mathscr{A} \to (\mathscr{A} \to \mathscr{A})) \to (\mathscr{A} \to \mathscr{A}))$ (Axiom L2)
4. $(\mathscr{A} \to (\mathscr{A} \to \mathscr{A})) \to (\mathscr{A} \to \mathscr{A})$ (MP 2,3)
5. $\mathscr{A} \to \mathscr{A}$ (MP 1,4)

Of course, no formula that has a proof in the system can be a contradiction because the system would be inconsistent!

What about the converse? Can every tautology be proved? In other words, is the logic system complete?

In Propositional Logic, that is the case: the theorems are exactly the tautologies. Our syntactic logic system "encapsulates" the semantic notion of tautology.

Chapter 2

Objectivising Logic: Predicate Logic

Propositional Logic allows us to derive logical conclusions based only on relations among propositions. Unfortunately, it is not adequate when we want to work in a more general context.

For example, from "Every riddle has a solution" and "The prisoners and hats problem is a riddle", one would infer "The prisoners and hats problem has a solution" but Propositional Logic can't.

We are in need of quantifying objects in our language, so we introduce the symbol $\forall$, which means "for all".

For example, given a formula $\mathscr{A}(x)$ which depends on x, "$\forall x(\mathscr{A}(x))$" is the universal quantifier and means "for all x, the formula $\mathscr{A}(x)$ is true".

The symbol $\exists$ is the existential quantifier and means "there exists". For instance, "$\exists x(\mathscr{A}(x))$" is "there exists an x such that $\mathscr{A}(x)$".

For example, "every positive number has a positive square root" can be written as:

$$\forall x((x > 0) \to \exists y((y > 0) \wedge (x = y * y)))$$

Note that we could also have defined the existential quantifier by means of the universal one:

$$\exists x(\mathscr{A}(x)) \equiv \neg\forall x(\neg\mathscr{A}(x))$$

We are going to use the following alphabet of symbols for the language we will call $\mathscr{L}$:

1. x_1, x_2,... as variables.
2. a_1, a_2,... as individual constants.
3. $A_1^1, A_2^1, ..., A_1^2, A_2^2$,... as predicate letters.
4. $f_1^1, f_2^1, ..., f_1^2, f_2^2$,... as function letters.

For example, defining the predicate letters A_1^1 as "is a girl", A_1^2 as "is the father of" and the individual constants a_1 as "Emma" and a_2 as "David", we can translate "Emma is David's daughter":

$$A_1^1(a_1) \wedge A_1^2(a_2, a_1)$$

Regarding predicate and function letters, the number over them indicates their arity, that is, the number of arguments they take.

Definition. A term in our language $\mathscr{L}$ is defined recursively as follows:
i) Variables and individual constants are terms.
ii) For every function letter f_i^n and terms t_1,.., t_n, $f_i^n(t_1,...,t_n)$ is a term.

The definition of formulas in Predicate Logic (also known as *first-order logic*) is almost the same as in Propositional Logic, adding the universal quantifier.

Definition. An atomic formula in $\mathscr{L}$ is $A_j^k(t_1,...,t_k)$ where A_j^k is a predicate letter and t_1,..., t_k are terms.

Definition. A well-formed formula (wf. for short) in $\mathscr{L}$ is an expression defined recursively as:
i) Any atomic proposition is a well-formed formula.
ii) If $\mathscr{A}$ and $\mathscr{B}$ are wfs., then $(\neg\mathscr{A})$, $(\mathscr{A}\vee\mathscr{B})$, $(\mathscr{A}\wedge\mathscr{B})$, $(\mathscr{A}\to\mathscr{B})$ and $(\forall x_i(\mathscr{A}))$ are also well-formed formulas.

We say that a variable x_i is *bound* in a wf. if it occurs within the scope of a universal quan*t*ifier $\forall x_i$ in the wf., or if it is the x_i in a $\forall x_i$. x_i is said to be *free* otherwise.

For example, in $A_1^1(x_1)\to(\forall x_2(A_2^1(x_2,x_1)))$, x_1 occurs free both times it appears and x_2 is bound by the quantifier.

A term t is *free* for x_i in a wf. $\mathscr{A}$ if x_i does not occur free in $\mathscr{A}$ within the scope of a $(\forall x_j)$, where x_j is any variable occurring in t. In this case, we can substitute every free occurrence of x_i in $\mathscr{A}$ for t.

Definition. An interpretation I of the language consists of a non-empty set D_I, which is called the domain of I, some elements of D_I ($\overline{a}_1$, $\overline{a}_2$,...) (one for each individual constant), some functions on D_I ($\overline{f}_i^n$, $i>0, n>0$) (one for each function letter) and some relations on D_I ($\overline{A}_i^n$, $i>0, n>0$) (one for each predicate letter).

For example, assume we have a language with an alphabet that contains a_1, A_1^2 and f_1^2. Let D_I={0,1,2,...}, the set of natural numbers. Here, $\overline{a}_1$ is 0 (this is the interpretation o*f* the individual constant a_1). The relation = is the interpretation of A_1^2, and addition is the interpretation of f_1^2.

Now, the formula $\forall x_1(A_1^2(f_1^2(x_1,a_1),x_1)$ has the interpretation "for every natural number x, x+0=x". Obviously, the formula is true in this case.

But, what if the interpretation of f_1^2 was multiplication instead of addition? Then, the formula would be interpreted as "for every natural number x, x*0=x", which is false.

Therefore, we cannot say that a formula in a first-order language is true or false. We can only say that when we are given an interpretation of the language. As we have seen, the same formula can have different truth values for different interpretations.

As the reader may have intuitively thought, every term of $\mathscr{L}$ is related to an object in the interpretation by means of a function v from the set of terms of $\mathscr{L}$:

$$v(a_i)=\overline{a}_i,$$

$$v(f_i^n(t_1,...,t_n))=\overline{f}_i^n(v(t_1),...,v(t_n))$$

Such a function v is called a "valuation in the interpretation I".

Definition. A valuation v in I is said to satisfy a wf. $\mathscr{A}$ in the following situations:
i) v satisfies the atomic formula $A^n_j(t_1,...,t_n)$ if $\overline{A}^n_j(v(t_1),...,v(t_n))$ is true in D_I.
ii) v satisfies $\neg\mathscr{B}$ if v does not satisfy $\mathscr{B}$.
iii) v satisfies $(\mathscr{B} \to \mathscr{C})$ if v satisfies $\neg\mathscr{B}$ or v satisfies $\mathscr{C}$.
iv) v satisfies $(\forall x_i)\mathscr{B}$ if for every valuation w such that v and w have the same values on each of the variables, except possibly on x_i, w satisfies $\mathscr{B}$.

Now, we have the tools to define truth in a predicate language.

A wf. formula is said to be *true* in an interpretation if it is satisfied by every valuation in the interpretation. It is said to be *false* if there does not exist any valuation which satisfies the formula.

Moreover, a wf. is called "logically valid" if it is true under every interpretation. On the other hand, it is called a "contradiction" if it is false under every interpretation.

Notice that logically valid formulas are the analogous to tautologies in Propositional Logic.

At this point, mimicking the last chapter, we continue to define the axioms of what is going to be our formal system $K_{\mathscr{L}}$:

Definition. The axioms in our system $K_{\mathscr{L}}$ are:
(K1) $(\mathscr{A} \to (\mathscr{B} \to \mathscr{A}))$.
(K2) $((\mathscr{A} \to (\mathscr{B} \to \mathscr{C})) \to ((\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{C})))$.
(K3) $(((\neg\mathscr{A}) \to (\neg\mathscr{B})) \to (\mathscr{B} \to \mathscr{A}))$.
(K4) $((\forall x_i)\mathscr{A} \to \mathscr{A})$, if x_i does not occur free in $\mathscr{A}$.
(K5) $((\forall x_i)\mathscr{A}(x_i) \to \mathscr{A}(t))$, if $\mathscr{A}(x_i)$ is a wf. of $\mathscr{L}$ and t is a term in $\mathscr{L}$ which is free for x_i in $\mathscr{A}(x_i)$.
(K6) $(\forall x_i)(\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to (\forall x_i)\mathscr{B})$, if $\mathscr{A}$ contains no free occurrence of the variable x_i,
for any formulas $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{C}$.

As before, we will use Modus Ponens as a rule, but we will add a new one:
Generalisation: for any wf. $\mathscr{A}$ and any variable x_i, from $\mathscr{A}$, deduce $(\forall x_i)\mathscr{A}$.

The definition of a *proof* is the same: a sequence of wfs. $\mathscr{A}_1,...\mathscr{A}_n$ of $\mathscr{L}$ such that every one is either an axiom or follows from previous formulas in the proof by Modus Ponens or Generalisation. We call the last member of a proof a *theorem*.

Theorem 2.1. *Soundness Theorem*
If a wf. $\mathscr{A}$ is a theorem of $K_{\mathscr{L}}$, then it is logically valid.

Therefore, our logic system is consistent, i.e. it cannot prove both $\mathscr{A}$ and $\neg\mathscr{A}$ for a wf. $\mathscr{A}$.

Besides, the converse of Theorem 2.1 is also true in Predicate Logic:

Theorem 2.2. *Adequacy Theorem*
If a wf. $\mathscr{A}$ is logically valid, then it is a theorem of $K_{\mathscr{L}}$.

This is also known as Gödel's completeness theorem for Predicate Logic. We can perceive here the same "encapsulation" of semantics and truth by syntax and theorems as in Propositional Logic.

Chapter 3

Into deeper waters: Mathematical systems

As we argued before, the only "universal truths" in this system are the theorems. But there are more wfs. that can be true under certain interpretations. We are dealing with structures, not meanings. As mathematicians, we would like to have a logic theory that could be applied to Mathematics. We would like then to have a symbol that meant "equality", and we should introduce some axioms in order to grasp its essence.

We extend the system in the following way:

Definition. The axioms regarding equality are:
(E7) $(\forall x_1)(x_1 = x_1)$.
(E8) $(t_k = u) \to (f_i^n(t_1,...,t_k,...,t_n) = f_i^n(t_1,...,u,...,t_n))$, for any terms t_1,..., t_n, u and any function letter f_i^n.
(E9) $(t_k = u) \to (A_i^n(t_1,...,t_k,...,t_n) \to (A_i^n(t_1,...,u,...,t_n))$ for any terms t_1,..., t_n, u and any predicate symbol A_i^n.

Notice that "=" is a predicate symbol, we could also call it A_1^2, for instance, but we keep this notation because of its clarity.

Any system with axioms (K1) to (K6) and (E7) to (E9) is called a *first-order system with equality*.

For example, let's prove the commutativity property:

$$(\forall x_1)(\forall x_2)((x_1 = x_2) \to (x_2 = x_1)).$$

1. $(x_1 = x_2) \to ((x_1 = x_1) \to (x_2 = x_2))$ using (E9) for A_1^2 as the symbol "=".
2. $((x_1 = x_2) \to ((x_1 = x_1) \to (x_2 = x_1))) \to (((x_1 = x_2) \to (x_1 = x_1)) \to ((x_1 = x_2) \to (x_2 = x_1)))$ because of (K2).
3. $((x_1 = x_2) \to (x_1 = x_1)) \to ((x_1 = x_2) \to (x_2 = x_1))$ by Modus Ponens 1, 2.
4. $((x_1 = x_1) \to ((x_1 = x_2) \to (x_1 = x_1)))$ by (K1).
5. $x_1 = x_1$ by (E7).
6. $(x_1 = x_2) \to (x_1 = x_1)$ by Modus Pone*n*s 4, 5.
7. $(x_1 = x_2) \to (x_2 = x_1)$ by Modus Ponens 3, 6.
8. $(\forall x_1)(\forall x_2)((x_1 = x_2) \to (x_2 = x_1))$ using Generalisation in 7.

The transitivity of equality is also easily proved:

$$(\forall x_1)(\forall x_2)(\forall x_3)((x_1 = x_2) \to ((x_2 = x_3) \to (x_1 = x_3))).$$

Now that we have the equality symbol, can we continue and add some more elements of Mathematics? Yes, why not? Let's take 0 as the interpretation of the constant variable a_1, the successor of a number as the interpretation of f_1^1, the sum of two numbers as the interpretation of f_1^2 and the product as the interpretation of f_2^2. For the sake of clarity, we will use the common mathematical symbols: x', + and *. This system is what we are going to call $\mathscr{S}$.

Again, we should add some axioms for these new characters in the story.

Definition. $\mathscr{S}$ is the first-order system with the language $\mathscr{L}$ of Arithmetic and axioms (K1) to (K6) and:
(S1) $(\forall x_1)(\forall x_2)(\forall x_3)((x_1 = x_2) \to ((x_1 = x_3) \to (x_2 = x_3)))$.
(S2) $(\forall x_1)(\forall x_2)((x_1 = x_2) \to (x_1{}' = x_2')).$
(S3) $(\forall x_1)\neg(x_1' = 0)$.
(S4) $(\forall x_1)(\forall x_2)((x_1' = x_2') \to (x_1 = x_2)$.
(S5) $(\forall x_1)(x_1 + 0 = x_1)$.
(S6) $(\forall x_1)(\forall x_2)(x_1 + x_2' = (x_1 + x_2)')$.
(S7) $(\forall x_1)(x_1 * 0 = 0)$.
(S8) $(\forall x_1)(\forall x_2)(x_1 * x_2' = (x_1 * x_2) + x_1)$.
(S9) $\mathscr{A}(0) \to ((\forall x_1)(\mathscr{A}(x_1) \to \mathscr{A}(x_1')) \to (\forall x_1)(\mathscr{A}(x_1))$, for any wf. $\mathscr{A}$ in which x_1 occurs free.

The axioms (S1)-(S9) are called *proper axioms*.

Given a language $\mathscr{L}$, a *first-order system* or a *theory* is an extension of $K_{\mathscr{L}}$ by adding some proper axioms. In particular, $\mathscr{S}$ is a theory with proper axioms (S1)-(S9), the ones from Peano Arithmetic for the natural numbers, regarding + and *. The last one, (S9) is a version of the Principle of Mathematical Induction.

From these new axioms, one can derive (E7) to (E9), so this is a first-order system with equality. Notice that a first-order system with equality is a theory, with proper axioms only the ones regarding equality.

Let's prove, for example, $(\forall x)(\forall y)(\forall z)((x+z=y+z) \to (x=y))$. This is the so-called "cancellation law for addition". We are going to use (S9), the induction principle on z:

1. It is clear that $(x+0=y+0) \to (x=y)$ (just using (S5)).

2. Now, assume $(x+z=y+z) \to (x=y)$. Does $(x+z'=y+z') \to (x=y)$? The latter is equivalent to $((x+z)' = (y+z)') \to (x=y)$, because of (S6). Now, using (S4), $(x+z=y+z) \to (x=y)$? But this is true by the induction hypothesis.

3. By Generalisation, $(\forall x)(\forall y)(\forall z)((x+z=y+z) \to (x=y))$.

This way, we can work with natural numbers as well as with wfs. in the logic system.

It is clear that $\mathscr{S}$ is consistent, since it's just an extension of a consistent axiom system and the new axioms we added are true in the *standard model* of arithmetic (the one with domain the set of natural numbers and +, *, 0 and 1 having their ordinary meaning).

But the important question is: "Is $\mathscr{S}$ complete?" Obviously, we would like it to be. All truths would be provable and all falsities falsifiable. Such a wonderful mathematical paradise!

Unfortunately, this is not the case, as we shall see later with the proof of Gödel's Incompleteness Theorem.

Chapter 4

The path towards the theorem I: Recursiveness

The last three chapters have been a very brief introduction (or refreshment) to the main aspects of the context in which the theorem arises.

Now, we can deal with a formal system that formalises arithmetic. But we still need more too*l*s in order to understand what Gödel proved.

The strategy we are going to follow from here to the theorem is this: in this chapter, we are going to define an important class of functions and deduce several of their properties. The most important concept in this chapter is the notion of "recursiveness", which will play a determinant role in the proof of our theorem. In the next chapter, we are going to present the ingenious idea Gödel had to express logic formulas in terms of numbers. Those are commonly known as "Gödel numbers" and are also key in the proof.

Definition. A number-theoretical function (or relation) is a function (or relation) whose arguments are natural numbers. A number-theoretical function takes natural numbers as values.

For example, addition is a number-theoretical function of two arguments, and so is multiplication. "=" is a number-theoretical relation that also takes two arguments.

The terms 0, $0'$, $0''$,... are called *numerals* and are denoted by 0, $\overline{1}$, $\overline{2}$,...

Definition. a) A number-theoretical relation is expressible in a theory K if and only if there exists a wf. $\mathscr{B}(x_1,...,x_n)$ of K with free variables x_1,..., x_n such that for any natural numbers $k_1,...,k_n$:

i) If $R(k_1,...k_n)$ is true, then $\mathscr{B}(\overline{k_1},...,\overline{k_n})$ is provable in K.

ii) If $R(k_1,...k_n)$ is false, then $\neg\mathscr{B}(\overline{k_1},...,\overline{k_n})$ is provable in K.

b) A number-theoretical function is representable in a theory K if and only if there exists a wf. $\mathscr{B}(x_1,...,x_n,y)$ of K with free variables x_1,..., x_n, y such that for any natural numbers $k_1,...,k_n$, m:

i) If $f(k_1,...k_n) = m$, then $\mathscr{B}(\overline{k_1},...,\overline{k_n},\overline{m})$ is provable in K

ii) $(\exists|y)\mathscr{B}(\overline{k_1},...,\overline{k_n},y)$ is provable in K.

For exa*m*ple, the zero function, Z(x)=0, is representable in K by the wf. $(x_1 = x_1) \wedge (y = 0)$. For i), if Z($k_1$)=m, then obviously m=0 and $(\overline{k_1} = \overline{k_1}) \wedge (0 = 0)$. ii) is also easily proved.

The successor function, N(x)=x+1, is representable in K by the wf. $y = x_1'$.

The projection function, $U_j^n(x_1,...,x_n) = x_j$, is representable in K by the wf. $(x_1 = x_1) \wedge (x_2 = x_2) \wedge \cdots \wedge (x_n = x_n) \wedge (y = x_j)$.

Let R be a relation that takes n arguments. The *characteristic function of R* is defined by:

$$C_R(x_1,...,x_n) = \begin{cases} 0 & \text{if} \quad R(x_1,...,x_n) \quad \text{is true} \\ 1 & \text{if} \quad R(x_1,...,x_n) \quad \text{is false} \end{cases}$$

Now, we see a relation between expressibility and representability.

Proposition 4.1. *For any theory K with equality, if it is provable in the system that* $0 \neq \overline{1}$*, then a number-theoretic relation R is expressible in K if and only if the function* C_R *is representable in K.*

Proof. =>) R is expressible in K by a wf. $\mathscr{A}(x_1,...,x_n)$. It is easy to prove that C_R is representable by $(\mathscr{A}(x_1,...,x_n) \wedge (y=0)) \vee (\neg\mathscr{A}(x_1,...,x_n) \wedge (y=1))$.

<=) C_R is representable in K by $\mathscr{B}(x_1,...,x_n)$. Using the fact that "$0 \neq \overline{1}$" is a theorem of the system, then R is expressible in K by $\mathscr{B}(x_1,...,x_n,0)$.

Definition. A function f is recursive if one of the following is true:

a) It is the zero function, Z(x)=0 for all x, the successor function, N(x)=x+1 for all x, or a projection function, $U_i^n(x_1,...,x_n) = x_i$ for all x_1,..., x_n.

b) It can be obtained from the functions in a) by a finite number of steps using the following rules:

i) (Substitution) $g(h_1(x_1,...,x_n)...,h_m(x_1,...,x_n))$ is the result of substituting the functions $h_1(x_1,...,x_n)$, ..., $h_m(x_1,...,x_n)$ in the function $g(y_1,...,y_m)$.

ii) (Recursion) Given $g(x_1,...,x_n)$ and $h(x_1,...,x_{n+2})$, the function f such that $f(x_1,...,x_n,0) = g(x_1,...,x_n)$ and $f(x_1,...,x_n,y+1) = h(x_1,...,x_n,y,f(x_1,...,x_n,y))$ is said to be obtained by recursion. In the case n=0, we have that $f(0) = k$ for k a fixed natural number and $f(y+1) = h(y,f(y))$.

iii) (Restricted μ-Operator) Given a function $g(x_1,...,x_n,y)$ such that, for every x_1,..., x_n there exists a y such that $g(x_1,...,x_n,y) = 0$, define the function $\mu g(x_1,...,x_n) = min\{y|g(x_1,...,x_n,y) = 0\}$ where μ is called μ-operator.

Examples

1. If $f(x_1,...x_k)$ is a recursive function, then $g(x_1,...,x_k,x_{k+1},...,x_n) = f(x_1,...x_k)$ is also recursive.

2. Another trivial example is that if $f(x_1,...x_n)$ is a recursive function, then we can permute variables and $f(x_3,x_1,x_2,...,x_n)$ is also recursive.

3. If $f(x_1,...,x_n)$ is a recursive function and $g(x_1,...,x_{n-1}) = f(x_1,...,x_{n-1},x_1)$, that is, we identify the last variable in f with the first one, then g is also recursive.

   The following functions are recursive:

4. x+y. Let $f(x,y) = x+y$ and $g(x) = x$. g is recursive since it is the identity, a projection function. Now, $f(x,0) = x+0 = x = g(x)$. And $f(x,y') = x+y' = (x+y)' = h(x,y,x+y) = h(x,y,f(x,y))$, where h is the composition of a projection and the successor functions, recursive by the substitution rule.

5. x*y. The proof is very similar to the one in 4.

6. The predecessor function

$$\delta(x) = \begin{cases} x-1 & \text{if} \quad x>0 \\ 0 & \text{if} \quad x=0 \end{cases}$$

   is recursive. Here the proof is also based on the recursion rule.

7. The function

$$x \dot{-} y = \begin{cases} x-y & \text{if} \quad x \geq y \\ 0 & \text{if} \quad x < y \end{cases}.$$

8.

$$|x-y| = \begin{cases} x-y & \text{if} \quad x \geq y \\ y-x & \text{if} \quad x < y \end{cases}.$$

Observe that $|x-y| = (x \dot{-} y) + (y \dot{-} x)$.

9.

$$sg(x) = \begin{cases} 0 & \text{if} \quad x = 0 \\ 1 & \text{if} \quad x \neq 0 \end{cases}.$$

Here, $sg(x) = x \dot{-} \delta(x)$, therefore it is clear that it is recursive.

10.

$$\overline{sg}(x) = \begin{cases} 0 & \text{if} \quad x = 0 \\ 1 & \text{if} \quad x \neq 0 \end{cases}.$$

11. $min(x_1, ..., x_n)$.

12. $max(x_1, ..., x_n)$.

13. rm(x,y)= remainder upon division of y by x.

14. qt(x,y)= quotient upon division of y by x.

15. If g_1,..,g_k and R_1,...,R_k are recursive functions and re*l*ations, respectively, and for any x_1,...x_n, one and only one of the relations $R_1(x_1, ..., x_n)$,..., $R_k(x_1, ..., x_n)$ is true, then the function:

$$f(x_1, ...x_n) = \begin{cases} g_1(x_1, ..., x_n) & \text{if } R_1(x_1, ..., x_n) \text{ is true} \\ g_2(x_1, ..., x_n) & \text{if } R_2(x_1, ..., x_n) \text{ is true} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ g_k(x_1, ..., x_n) & \text{if } R_k(x_1, ..., x_n) \text{ is true} \end{cases}.$$

is also recursive. The reason is that $f(x_1, ..., x_n) = g_1(x_1, ..., x_n) * \overline{sg}(C_{R_1}(x_1, ..., x_n)) + \cdots + g_k(x_1, ..., x_n) * \overline{sg}(C_{R_k}(x_1, ..., x_n))$.

16. If $f(x_1, ..., x_n, y)$ is recursive, then

$$\prod_{y<z} f(x_1, ..., x_n, y) = \begin{cases} 1 & \text{if} \quad z = 0 \\ f(x_1, ..., x_n, 0) * \cdots * f(x_1, ..., x_n, z-1) & \text{if} \quad z > 0 \end{cases}$$

is recursive.

17. The function p(x) (or, as we shall denote it in the future, p_x) given by p(x)= x^{th} prime number, is recursive. For example, p_0=2, p_1=3, p_2=5,...

18. Given a number x and its factorisation into prime powers $x = p_0^{a_0} p_1^{a_1} \cdots p_k^{a_k}$, define the function $(x)_j = a_j$. It is also recursive.

We would like sometimes to define functions by a recursion in which $f(x_1,...,x_n,y+1)$ does not depend only on $f(x_1,...,x_n,y)$, but on several (possibly all) values of $f(x_1,...,x_n,u)$ for $u \leq y$. This type of recursion is called *course-of-values recursion*. Let $f\#(x_1,...,x_n,y) = \prod_{u<y} p_u^{f(x_1,...,x_n,u)}$. That is, $f\#$ "stores" all the previous values of f. Note that f can be obtained from $f\#$: $f(x_1,...x_n,y) = (f\#(x_1,...,x_n,y+1))_y$.

Proposition 4.2. *If $f(x_1,...,x_n,y,z)$ is recursive, then $h(x_1,...,x_n,y) = f(x_1,...,x_n,y,f\#(x_1,...,x_n,y))$ is recursive.*

Proof.

$$h\#(x_1,...,x_n,0) = 1$$

$$h\#(x_1,...,x_n,y+1) = h\#(x_1,...,x_n,y) * p_y^{h(x_1,...,x_n,y)}$$

$$= h\#(x_1,...,x_n,y) * p_y^{f(x_1,...,x_n,y,h\#(x_1,...,x_n,y))}$$

Therefore, by the recursion rule, $h\#$ is recursive and

$$h(x_1,...,x_n,y) = (f\#(x_1,...,x_n,y+1))_y.$$

Corollary 4.1. *Let $H(x_1,...,x_n,y,z)$ be a recursive relation.*
If $R(x_1,...,x_n,y)$ holds if and only if $H(x_1,...,x_n,y,(C_R)\#(x_1,...,x_n,y))$, then R is recursive.

Proof. We can express $C_R(x_1,...,x_n,,y) = C_H(x_1,...,x_n,y,C_R(x_1,...,x_n,y))$. C_H is recursive, so C_R is recursive too by the previous proposition, and, therefore, so is R.

Now, we are arriving at one of the important results in this chapter: that the notion of recursiveness implies representability or expressibility, depending on whether we are talking about functions or relations. But, first, we need two lemmas.

Lemma 4.1. *Gödel's β-function*
Define $\beta(x_1,x_2,x_3) = rm(1+(x_3+1)*x_2,x_1)$ (remember that rm is just the remainder of the division).
We know that β is recursive. Furthermore, it is representable as well. And a wf. that represents β is

$$B(x_1,x_2,x_3,y) : (\exists W)((x_1 = (1+(x_3+1)*x_2)*w+y) \wedge (y < 1+(x_3+1)*x_2)).$$

Lemma 4.2. *Let k_0, k_1,..., k_n be a sequence of natural numbers. Then there exist natural numbers b and c such that $\beta(b,c,i) = k_i$ for $0 \leq i \leq n$.*

Proposition 4.3. *Every recursive function is representable in $\mathscr{S}$.*

Proof. For the proof, see proposition 3.24 of [Mendelson].

Corollary 4.2. *Every recursive relation is expressible in $\mathscr{S}$.*

Proof. Let $R(x_1,...,x_n)$ be a recursive relation. We know that C_R is also recursive. By the last proposition, C_R is representable in $\mathscr{S}$. Now, using Proposition 4.1, we conclude that R is expressible in $\mathscr{S}$.

Chapter 5

The path towards the theorem II: Gödel numbers

Definition. The Gödel number of a symbol u in a first-order theory K is an odd positive integer g(u) defined as follows:

i) g(()=3, g())=5, g(,)=7, g($\neg$)=9, g($\rightarrow$)=11, g($\forall$)=13.
ii) g(x_k)= 13+8k for $k \geq 1$.
iii) g(a_k)= 7+8k for $k \geq 1$.
iv) g(f_k^n)= 1+8($2^n 3^k$) for $k, n \geq 1$.
v) g(A_k^n)= 3+8($2^n 3^k$) for $k, n \geq 1$.

Indeed, every Gödel number is an odd number. Moreover, we can "recover" the symbols from their numbers in this way: if the number is 3, 5, 7, 9, 11 or 13, then it is clear which symbol it comes from (the ones in i)); otherwise, divide by 8 and if we can express it with a remainder of 5, then the origi*n*al symbol is a variable as in ii); if the remainder is 7, it is an individual constant as in iii); if it is 1, it is a function letter as in iv); and if it is 3, then the symbol is a predicate letter as in v).

Examples

1. The Gödel number of x_3 is g(x_3)=13+8*3=37.
2. The Gödel number of a_2 is g(a_2)=7+8*2=23.
3. The Gödel number of f_2^3 is g(f_2^3)=1+8($2^3 3^2$)=1+576=577.
4. The Gödel number of A_1^3 is g(A_1^3)=3+8($2^3 3^1$)=3+192=195.
5. Which is the symbol whose Gödel number is 45? Dividing by 8, we get 45=5*8+5. But 5 is not one of our remainders. No, but... 45=4*8+13. And therefore, the symbol is x_4.
6. Which is the symbol whose Gödel number is 31? Again, we divide by 8 and 31=3*8+7. Therefore, the original symbol is a_3.
7. What if the number is 145? Proceeding in the same manner, 145=8*18+1. The symbol must be a function letter and, since $18 = 2 * 3^2$, it must be f_2^1.
8. If the Gödel number is 51=8*6+3, then it must be the p*r*edicate letter A_1^1 because 6=2*3.

Now, how can we translate into numbers in a similar way longer expressions, not only single symbols? For instance, for $(\forall x_1)$, should we write 313215, or rather 3+13+21+5? It does not seems a good way to do so, because we wouldn't be able to recover anything. For example, in the first case, 313215 could also be $a_3(x_1)$ and, in the second case, 3+13+21+5=42=3+39, that is, $(a_4$, or even $a_4($ or $)(x_1\forall$. So what should we do?

Definition. Let $u_0u_1...u_r$ be an expression where each u_i is a symbol of the first-order theory K. Its Gödel number is given by:

$$g(u_0u_1...u_r) = 2^{g(u_0)}3^{g(u_1)}...p_r^{g(u_r)}$$

where p_r means the j^{th} prime number, starting with $p_0 = 2$.

<u>**Examples**</u>

1. The Gödel number of the expression $(\forall x_1)$ considered above is thus $2^3 3^{13} 5^{21} 7^5$.
2. The Gödel number of $f_1^2(x_1,x_2)$ is $2^{97}3^3 5^{21} 7^7 11^{29} 13^5$ (do not forget the comma!). But in a different order, $f_1^2(x_2,x_1)$ gives $2^{97}3^3 5^{29} 7^7 11^{21} 13^5$, which is a different number.
3. The number $2^9 3^{51} 5^3 7^{15} 11^5$ comes from $\neg A_1^1(a_1)$.
4. The symbol x_2 has Gödel number 29, but the expression consisting only of x_2 has Göde*l* number 2^{29}.

Given a positive number, there exists only one expression (in case such an expression exists) whose Gödel number is that number, because of the unique factorization of integers into primes. Since every expression has at least one symbol, the number 2 will be a factor of every Gödel number of an expression, i.e. Gödel numbers are even. Moreover, since the Gödel number of a symbol is always an odd number, these are not only even, but the exponent of 2 in their factorization is odd. This is important, as we shall see in a moment.

Definition. Let $e_0, e_1, ..., e_r$ be a finite sequence of expressions of the first-order theory K. Its Gödel number is given by:

$$g(e_0, e_1, ..., e_r) = 2^{g(e_0)}3^{g(e_1)} \cdots p_r^{g(e_r)}.$$

How could we tell whether a given Gödel number comes from a sequence of expressions or from a single expression? They have almost the same definition! There is no need to worry, the answer is truly simple. As we state*d* before, the Gödel number of an expression is an even number such that the exponent of 2 in its factorization is odd. Well, since it is even, the Gödel number of a sequence of expressions will have an even power of 2 (and therefore, it will also be even). So, in conclusion: even Gödel number with an odd power of 2 => expression. Even Gödel number with an even power of 2 => sequence of expressions.

Notice that not every positive integer is the Gödel number of something. For example, 14 or 20 are not Gödel numbers.

Since a proof in K is a certain kind of finite sequence of expressions, every proof has a Gödel number.

Definition. A theory K has a recursive vocabulary if the following relations are recursive:
i) IC(x): x is the Gödel number of an individual constant of K,
ii) FL(x): x is the Gödel number of a function letter of K, and
iii) PL(x): x is the Gödel number of a predicate letter of K.

Definition. A theory K has a recursive axiom set if:
PrAx(x): x is the Gödel number of a proper axiom of K (i.e. it is one of (S1)-(S9))
is recursive.

Proposition 5.1. *Let K be a theory having a recursive vocabulary and a recursive axiom set, and whose language contains the individual constant 0 and the successor function. Then the following functions and relations are recursive:*

1. *EVbl(x): x is the Gödel number of an expression consisting of a variable.*
2. *EIC(x): x is the Gödel number of an expression consisting of an individual constant.*
3. *EFL(x): x is the Gödel number of an expression consisting of an function letter.*
4. *EPL(x): x is the Gödel number of an expression consisting of an predicate letter.*
5. *Wf(x): x is the Gödel number of a wf. of K.*
6. *MP(x,y,z): z is the Gödel number of the expression that is a direct consequence of the expressions with Gödel numbers x and y by Modus Ponens.*
7. *Gen(x,y): y is the Gödel number of the expression that comes from the expression with Gödel number x by the Generalisation rule.*
8. *Fr(y,v): y is the Gödel number of a wf. or term of K that contains free occurrences of the variable with Gödel number v.*
9. *Neg(x): the Gödel number of the negation of the wf. whose Gödel number is x.*
10. *LAx(x): x is the Gödel number of a logical axiom of K.*
11. *Prf(x): x is the Gödel number of a proof in K.*
12. *Pf(x,y): x is the Gödel number of a proof in K of the wf. with Gödel number y.*
13. *Sub(y,u,v): the Gödel number of the result of substituting the term with Gödel number u for all free occurrences in the expression with Gödel number y of the variable with Gödel number v.*
14. *D(u): the Gödel number of $\mathscr{B}(\overline{u})$, if u is the Gödel number of a wf. $\mathscr{B}(x_1)$.*

We now have the means to prove the converses of proposition 4.3 and corollary 4.2.

Proposition 5.2. *Let $\mathscr{S}$ be the theory as in the previous chapter. Let $f(x_1,...,x_n)$ be a representable function in $\mathscr{S}$. Then, f is recursive.*

Proof. For the proof, see propositio*n* 3.29 of [Mendelson].

Therefore, the class of recursive functions is identical to the class of representable functions in $\mathscr{S}$.

Corollary 5.1. *In the same situation, every number-theoretic relation that is expressible in $\mathscr{S}$ is recursive.*

Proof. Let R be an expressible relation. We *k*now that C_R is representable in $\mathscr{S}$ if and only if R is expressible in $\mathscr{S}$. Therefore, C_R is representable. By the proposition, C_R is recursive. But, by the definition, this means that R is recursive.

In conclusion, a number-theoretic relation $R(x_1,...,x_n)$ is recursive if and only if it is expressible in $\mathscr{S}$.

Chapter 6

Facing the reality: The Incompleteness Theorem

Remember we have defined a function D (which we will call diagonal function) such that D(u) is the Gödel number of $\mathscr{B}(\overline{u})$, if u is the Gödel number of a wf. $\mathscr{B}(x_1)$. We are now going to use this function to prove the following proposition.

Proposition 6.1. *Fixed-point theorem*

Let $\mathscr{A}(x_1)$ be a wf. of the theory $\mathscr{S}$ in which x_1 is the only free variable.

Then, there exists a closed wf. $\mathscr{B}$ such that it is provable in the theory that

$$\mathscr{B} \leftrightarrow \mathscr{A}(\overline{q}),$$

where q is the Gödel number of $\mathscr{B}$.

Proof. D is recursive. Therefore, it is representable by a wf. $D(x_1,x_2)$ in $\mathscr{S}$. Let m be the Gödel number of the wf. $(\forall x_2)(D(x_1,x_2) \rightarrow \mathscr{A}(x_2))$. Substituting $\overline{m}$ for x_1 in this formula, we get $(\forall x_2)(D(\overline{m},x_2) \rightarrow \mathscr{A}(x_2))$. Call this formula $\mathscr{B}$ and let q be its Gödel number.

By the definition of the diagonal function, D(m)=q. Since $D(x_1,x_2)$ represents D in $\mathscr{S}$, it is provable that $D(\overline{m},\overline{q})$.

We now have to prove that this $\mathscr{B}$ is the formula we are looking for. First, we prove that $\mathscr{B} \rightarrow \mathscr{A}(\overline{q})$ is provable.

1. $\mathscr{B}$ — Hypothesis
2. $(\forall x_2)(D(\overline{m},x_2) \rightarrow \mathscr{A}(x_2))$ — Same as 1
3. $D(\overline{m},\overline{q}) \rightarrow \mathscr{A}(\overline{q})$ — 2, K5
4. $D(\overline{m},\overline{q})$ — Proven above
5. $\mathscr{A}(\overline{q})$ — MP 3-4

And, therefore, $\mathscr{B} \rightarrow \mathscr{A}(\overline{q})$. Now, let's prove the converse.

1. $\mathscr{A}(\overline{q})$ — Hypothesis
2. $D(\overline{m},x_2)$ — Hypothesis
3. $(\exists|x_2)(D(\overline{m},x_2))$ — Representability of D
4. $D(\overline{m},\overline{q})$ — Proved above
5. $x_2 = \overline{q}$ — 2-4, properties of =

6. $\mathscr{A}(x_2)$ Properties of =

7. $D(\overline{m},x_2) \to \mathscr{A}(x_2)$

8. $(\forall x_2)(D(\overline{m},x_2) \to \mathscr{A}(x_2))$ Generalization

Therefore, $\mathscr{A}(\overline{q}) \to (\forall x_2)(D(\overline{m},x_2) \to \mathscr{A}(x_2))$, that is, $\mathscr{A}(\overline{q}) \to \mathscr{B}$.
By biconditional introduction, $\mathscr{B} \leftrightarrow \mathscr{A}(\overline{q})$.

We just need two more ingredients to add: the notions of ω-consistency and undecidability, and everything will be ready for the Theorem.

Definition. A theory K whose language contains the individual constant 0 and the successor function is said to be ω-consistent if for every wf. $\mathscr{A}$(x) of K containing x as its only free variable, if $\neg\mathscr{A}(\overline{n})$ is provable in K for every natural number n, then $(\exists x)(\mathscr{A}(x))$ is not provable in K.

It is easy to prove that ω-consistency implies consistency: let $\mathscr{B}$(x) be a wf. with x its only free variable. Consider the formula $\mathscr{B}(x) \wedge \neg\mathscr{B}(x)$, let's call it $\mathscr{A}(x)$. Of course, the negation of $\mathscr{A}(\overline{n})$ is an instance of a tautology. Therefore, $\neg\mathscr{A}(\overline{n})$ is provable in K for every natural number n. Since, K is ω-consistent, $(\exists x)(\mathscr{A}(x))$ is not provable in K. If K were not consistent, it would be possible to prove everything in K. But we found something that is not provable, so K is consistent.

Definition. An undecidable sentence of a theory K is a closed wf. $\mathscr{C}$ of K such that neither $\mathscr{C}$ nor $\neg\mathscr{C}$ is a theorem of K.

Recall that Pf(x,y) means that x is the Gödel number of a proof in K of the wf. with Gödel number y. If we stick to the theory $\mathscr{S}$, that is, Peano Arithmetic, we know Pf is recursive and, therefore, expressible in $\mathscr{S}$ by a wf. $\mathscr{P}f(x_1,x_2)$.

Applying the fixed-point theorem to the wf. $(\forall x_1)(\neg\mathscr{P}f(x_1,x_2))$, there exists a closed wf. $\mathscr{G}$ such that

$$\mathscr{G} \leftrightarrow (\forall x_1)(\neg\mathscr{P}f(x_1,\overline{q}))$$

is provable in $\mathscr{S}$, where q is the Gödel number of $\mathscr{G}$.

As we stated in the introduction of this thesis, we were looking for a sentence that states its own unprovability. But this is precisely what $\mathscr{G}$ does! $(\forall x_1)(\neg\mathscr{P}f(x_1,\overline{q}))$ says that there is no natural number that is the Gödel number of a proof in $\mathscr{S}$ of the wf. $\mathscr{G}$, that is, that there is no proof in $\mathscr{S}$ of $\mathscr{G}$. And, since $\mathscr{G}$ is equivalent to $(\forall x_1)(\neg\mathscr{P}f(x_1,\overline{q}))$, it is clear that it does indeed assert its own unprovability.

This wf. $\mathscr{G}$ is called a Gödel sentence and we can now prove that $\mathscr{G}$ is undecidable.

Theorem 6.1. *Gödel's Incompleteness Theorem*
Let $\mathscr{S}$ be Peano Arithmetic.
i) If $\mathscr{S}$ is consistent, then $\mathscr{G}$ is not a theorem of $\mathscr{S}$.
ii) If $\mathscr{S}$ is ω-consistent, then $\mathscr{G}$ is not a theorem of $\mathscr{S}$.
Hence, if $\mathscr{S}$ is ω-consistent, $\mathscr{G}$ is an undecidable sentence of $\mathscr{S}$.

Proof. i) By Reductio ad Absurdum, assume $\mathscr{G}$ is provable in $\mathscr{S}$. Let r be the Gödel number of a proof, so Pf(r,q). Hence, $\mathscr{P}f(\overline{r},\overline{q})$ is provable in $\mathscr{S}$. But, on the other hand, $\mathscr{G} \leftrightarrow (\forall x_1)(\neg\mathscr{P}f(x_1,\overline{q}))$ is provable in $\mathscr{S}$, so $(\forall x_1)(\neg\mathscr{P}f(x_1,\overline{q}))$ is also provable in $\mathscr{S}$. In particular, $\neg\mathscr{P}f(\overline{r},\overline{q})$ is provable by Modus Ponens. But this leads to a contradiction, since $\mathscr{S}$ is consistent and $\mathscr{P}f(\overline{r},\overline{q})$ is provable in $\mathscr{S}$.

ii) Again, by Reductio ad Absurdum, assume that $\neg\mathscr{G}$ is provable in $\mathscr{S}$. From the same property as before, $\neg(\forall x_1)(\neg\mathscr{P}f(x_1,\overline{q}))$ is provable. This formula is equivalent to $(\exists x_1)(\mathscr{P}f(x_1,\overline{q}))$. On the other hand, $\mathscr{S}$ is consistent, since it it ω-consistent. Therefore, since $\neg\mathscr{G}$ is provable, it is not the case that $\mathscr{G}$ is provable. In other words, there is no proof in $\mathscr{S}$ of $\mathscr{G}$. This means that Pf(n,q) is false for every natural number n. Therefore, $\neg\mathscr{P}f(\overline{n},\overline{q})$ is provable in $\mathscr{S}$ for every n. Finally, by ω-consistency, it is not the case that $(\exists x_1)(\mathscr{P}f(x_1,\overline{q}))$ is provable in $\mathscr{S}$, which is a contradiction.

In consequence, $\mathscr{G}$ is a sentence that states its own unprovability in $\mathscr{S}$ and it is not provable in $\mathscr{S}$. Therefore $\mathscr{G}$ is true in the standard model.

Another important consequence of what we have derived is Gödel's Second Theorem. We know that there are some undecidable sentences. If we could express somehow the notion of consistency, would it be a theorem of arithmetic? In other words, would the theory be "conscious" of its own consistency? This is exactly what the Second theorem answers.

First, how can we express consistency by means of a wf.? A theory is consistent if there is no proof of a wf. and its negation. The relation Pf and the function Neg are recursive. Hence, Pf is expressible in $\mathscr{S}$ by a wf. $\mathscr{P}f(x_1,x_2)$ and Neg is representable in $\mathscr{S}$ by a wf. $\mathscr{N}eg(x_1,x_2)$.

Let $\mathscr{C}on$ be the following wf.:

$$(\forall x_1)(\forall x_2)(\forall x_3)(\forall x_4)\neg(\mathscr{P}f(x_1,x_3)\wedge\mathscr{P}f(x_2,x_4)\wedge\mathscr{N}eg(x_3,x_4)).$$

In the standard interpretation, this is the same as that there are no proofs in $\mathscr{S}$ of a wf. and its negation.

Theorem 6.2. *Gödel's Second Theorem*
Let $\mathscr{S}$ be the theory we have been working with (Peano Arithmetic).
If $\mathscr{S}$ is consistent, then $\mathscr{C}on$ is not a theorem of $\mathscr{S}$.

Proof. For the proof, see proposition 3.42 of [Mendelson].

This means that a proof of consistency must use ideas and methods that are not available in $\mathscr{S}$. In fact, there are consistency proofs but it is not possible to formalise them in $\mathscr{S}$.

Finally, we would like to conclude this dissertation by talking about Church's thesis and what it implies.

Church's thesis: A number-theoretic function is effectively computable if and only if it is recursive.

What does it mean to be "effectively computable"? That there is an algorithm that correctly calculates the function.

Definition. Let K be a theory. K is recursively decidable if $Th = \{n \in \mathbb{N}|$ n is the Gödel number of a theorem of K$\}$ is a recursive set (that is, "$x \in Th$ is recursive"). Otherwise, K is recursively undecidable.

Assuming Church's thesis, the notion of recursive decidability is equivalent to the fact that there exists an algorithm that decides whether a formula is a theorem of K or not.

Let $K_{\mathscr{L}}$ be a first-order system on the language $\mathscr{L}$ of Arithmetic. Now, extend it by adding ALL formulas of Arithmetic which are true in the standard model as proper axioms, obtaining a system that is clearly complete. $Tr = \{n \in \mathbb{N}|$ n is the Gödel number of a true wf. of K in the standard model$\}$ is not recursive: if it was, we would have a recursive axiom set and, hence, the theory would be incomplete by Gödel's Theorem, in contradiction with the fact that it is complete. In other words, accepting Church's thesis, there is no algorithm or decision procedure for determining whether a wf. of Arithmetic is true in the standard model or not. This is a completely different situation from the one we had, for example, in the first chapter. Back then, we had a decision method: constructing the truth table of the formula and checking whether the last column has only 1's. Besides, Th, the set of Gödel numbers of theorems of $\mathscr{S}$, is not recursive either (for the proof, see section 7.4 of [Hamilton]). Hence, by Church's thesis, there is no algorithm capable of deciding whether a given formu*l*a is a theorem of Arithmetic or not.

Bibliography

[Berger] U. BERGER *Logic for Computer Science. Lecture notes*, University of Swansea, 2015.

[Genesereth] M. GENESERETH *Introduction to Logic: Second Edition (Synthesis Lectures on Computer Science)*, disponible en `https://d396qusza40orc.cloudfront.net/intrologic/notes/notes.html`.

[Hamilton] A. G. HAMILTON *Logic for Mathematicians. Revised edition*, Cambridge University Press, 1978.

[Mendelson] E. MENDELSON *Introduction to mathematical Logic. Fourth edition*, Chapman & Hall, 1997.