El Teorema de Incompletitud de Gödel

Víctor López Martínez

Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Director del trabajo: Carlos Gómez Ambrosi
Junio de 2016

Summary

In this dissertation, we aim to give a somewhat complete proof of Gödel's Incompleteness Theorem, one of the central Theorems in mathematical logic. The basic idea behind this Theorem[1] is that, if we attempt to formalize number theory in computable way, there will be number-theoretic truths that can't be proven mechanically, and so any such attempt will be essentially incomplete. We will spend the first chapter defining formal systems and key concepts regarding them, and giving some results on formal systems that will prove useful for our study, while in the second chapter we will focus on how we can apply these results to study number theory from a formal point of view.

We start by defining the idea of a formal system, the mathematical object on which the rest of our study will be based, and we explain some of the key concepts related to them, such as axioms, theorems, proofs and deductions. As an example, we define one of the most basic formal systems, *L*, which formalizes propositional calculus and involves concepts such as truth values, tautologies... We then proceed to give two of the most important properties of this system. These properties are consistency, which assures that we don't encounter any intuitively impossible situation, such as a sentence being true and false at the same time; and completeness, which refers to the system's ability to accurately capture our idea of truth in its theorems. Gödel's Incompleteness Theorem proves precisely that a certain kind of system does *not* have this property of completeness.

In the next section we introduce a particular kind of formal system (first-order systems) which will be of special interest to our study, defining their language first and their behaviour afterwards. These systems widen our study of propositional calculus through *L* to predicate calculus, in a way "splitting" sentences into their subject and predicate in a formal way. We also introduce the idea of interpretation and truth, which correspond to the concept of truth value that we saw in *L*. Then, we define all the concepts we need to prove the properties of completeness and consistency for a particular first-order system (the predicate calculus), such as extensions and models, and give some important results regarding models such as the Skolem-Löwenheim Theorem, which allows us to reduce any model of a first-order system to a denumerable model.

We close the chapter by introducing first-order systems with equality (that is, systems where we have a predicate, or property, that *can* be interpreted as equality) and normal models (in which said predicate *is* indeed seen as equality). This will be our first step towards formalizing number theory, which we will complete during the second chapter. We then extend the Skolem-Löwenheim Theorem to deal with normal models as well.

The second chapter is focused on the study of formal systems applied to arithmetic. First, we give the formal system S, based on the Peano postulates, which aims to capture arithmetic in a formal system, and then give an explanation about its properties on which we will expand later. *S* is indeed a first-order system with equality, so we can apply the theory we've already seen in the first chapter. We also briefly discuss one of the most unintuitive points about the system *S*; that is, the fact that its consistency is not something that can be proven like we did with *L*, but has to be taken as an unproven truth.

We then move to the study of functions and relations in number theory, which we name arithmetic functions and relations. After the basic definitions and the idea of characteristic function, we introduce the concept of recursivity and primitive recursivity. The concepts of representability and expressibility

[1] As we will explain in the first chapter, we refer to important results on formal systems as Theorems (capital T), to avoid confusion with the concept of theorem of a formal system.

allow us to link the formal systems we've been studying so far with arithmetic functions and relations. We then state that recursive functions and relations are, respectively, representable and expressible in S, highlighting the importance of recursivity.

Next, we define Gödel numbers, which allow us, in a certain way, to express the properties of a formal system inside the formal system itself, and as such are the fundamental tool for the proof of Gödel's Incompleteness Theorem. Through Gödel numbers we can define arithmetic functions and relations which translate properties of our system to number theory, and by means of recursivity and under certain conditions we find that we can represent and express these properties in S, looping back around into formal systems. Although we could do extensive study of the arithmetic functions and relations that can or cannot be represented or expressed in S, we consider here only a selected few that will be needed for the proof of Gödel's Incompleteness Theorem.

The last section is devoted to defining the last few concepts we will need for the proof, such as ω-consistency, and to finally use all the tools we've introduced to give a proof of Gödel's Incompleteness Theorem. We do this by creating a sentence $\mathscr{G}$ that asserts its own unprovability, via the technique we mentioned before of moving our properties back and forth between formal systems and number theory, by means of recursivity and Gödel numbers. We close with a brief explanation of the consequences of this Theorem for mathematics and computability; essentially, all our attempts at using formal systems for number theory will ultimately fail to capture every possible number-theoretic truth, and any system that so did could not be described in a recursive (i.e. computable) way and as such it would be inaccessible to us or any machine we could build with that purpose.

Índice general

Capítulo 1

Sistemas formales

1.1. El sistema *L*

Comenzaremos dando una definición precisa de lo que será el objeto de nuestro estudio, los sistemas formales.

Definición. Un *sistema formal* o *teoría* viene dado por los siguientes elementos:

- Un *alfabeto*, es decir, un conjunto numerable de símbolos. Una cadena finita de símbolos se dirá *expresión*.
- Un subconjunto de las posibles expresiones. Estas se llamarán *fórmulas bien formadas*, o *fórmulas* para abreviar. Si hemos definido un alfabeto y sus fórmulas bien formadas, diremos que tenemos un *lenguaje*.
- Un subconjunto de las fórmulas bien formadas, a las que llamaremos *axiomas*.
- Una cantidad finita de *reglas de deducción*, que nos permiten obtener una nueva fórmula a partir de una o más fórmulas dadas. Diremos que la nueva fórmula se deduce de las anteriores mediante dicha regla.

Dado un conjunto de fórmulas Γ, una *prueba desde* Γ (o simplemente *prueba* si Γ es vacío) es una sucesión finita de fórmulas $\mathscr{A}_1, ..., \mathscr{A}_n$ donde cada $\mathscr{A}_i$ es un axioma, un elemento de Γ o una fórmula que se deduce de las anteriores de la sucesión mediante una de las reglas de deducción. Diremos que $\mathscr{A}$ es *consecuencia* de Γ si es el último elemento de una prueba desde Γ (escrito $\Gamma \vdash_F \mathscr{A}$), y diremos que es un *teorema* si es el último elemento de una prueba (escrito $\vdash_F \mathscr{A}$), donde F es el sistema formal en cuestión. Omitiremos el subíndice $_F$ cuando no haya ambigüedad posible.

A modo de ejemplo, introduciremos el sistema formal *L*, que formaliza la lógica proposicional.

Definición. El sistema *L* para el cálculo proposicional viene dado por lo siguiente:

- Sus símbolos son las *letras de sentencia* $A_1, A_2, A_3, ...$, los *conectores* $\rightarrow$ y $\neg$, y los símbolos "(", "," y ")".[1]
- Toda letra de sentencia es una fórmula bien formada. Asimismo, si $\mathscr{A}$ y $\mathscr{B}$ son fórmulas, entonces $(\mathscr{A} \rightarrow \mathscr{B})$ y $(\neg \mathscr{A})$ también lo son.
- Cualquier fórmula de una de las siguientes formas es un axioma, donde $\mathscr{A}$, $\mathscr{B}$ y $\mathscr{C}$ son fórmulas cualesquiera:

  (L1) $\mathscr{A} \rightarrow (\mathscr{B} \rightarrow \mathscr{A})$

[1] Notación: Los paréntesis se omitirán cuando no haya ambigüedad posible. Del mismo modo, las comas no son técnicamente necesarias, pero se usarán por claridad de lectura. Seguiremos la convención establecida en [1, pág. 17–18].

(L2) $(\mathscr{A} \to (\mathscr{B} \to \mathscr{C})) \to ((\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to \mathscr{C}))$

(L3) $(\neg\mathscr{B} \to \neg\mathscr{A}) \to ((\neg\mathscr{B} \to \mathscr{A}) \to \mathscr{B})$[2]

- La única regla de deducción es Modus Ponens, abreviado MP: si $\mathscr{A}$ y $\mathscr{B}$ son fórmulas cualesquiera, entonces $\mathscr{B}$ se deduce de $\mathscr{A}$ y $(\mathscr{A} \to \mathscr{B})$ mediante Modus Ponens.

Además de los conectores dados en el alfabeto, usaremos también $(\mathscr{A} \wedge \mathscr{B})$, $(\mathscr{A} \vee \mathscr{B})$ y $(\mathscr{A} \leftrightarrow \mathscr{B})$, como abreviaturas de $\neg(\mathscr{A} \to \neg\mathscr{B})$, $((\neg\mathscr{A}) \to \mathscr{B})$ y $((\mathscr{A} \to \mathscr{B}) \wedge (\mathscr{B} \to \mathscr{A}))$, respectivamente.

Este sistema nos permite estudiar desde un punto de vista puramente formal la lógica proposicional. Las fórmulas de L se ven, intuitivamente, como proposiciones que pueden tomar el valor de verdad "verdadero" o "falso", y el valor de verdad de una fórmula con conectores depende de los valores de verdad de las fórmulas más pequeñas que contiene. Para un estudio más riguroso de la lógica proposicional, ver [1, pág. 10–25]. Por ahora, sólo nos interesa trasladar el concepto de *tautología* al sistema formal L, entendiendo por tautología una fórmula que siempre toma el valor de verdad "verdadero".

Como ejemplo de los conceptos introducidos, presentamos una prueba sencilla en L:

(1)	$A_1 \to (A_2 \to A_1)$	(L1)
(2)	$(A_1 \to (A_2 \to A_1)) \to ((A_1 \to A_2) \to (A_1 \to A_1))$	(L2)
(3)	$(A_1 \to A_2) \to (A_1 \to A_1)$	(1), (2), MP

Por tanto, $\vdash (A_1 \to A_2) \to (A_1 \to A_1)$. Notar que esta fórmula es también una tautología en el sentido de la lógica proposicional.

El siguiente Teorema[3] dice que este es precisamente el caso para todo teorema de L.

Teorema 1.1 (Completitud de L)**.** *$\vdash_L \mathscr{A}$ si y sólo si $\mathscr{A}$ es una tautología.*

Para una demostración completa, ver [1, pág. 34].

Es fácil probar que todo teorema es una tautología; basta probar que todo axioma lo es, y que la regla Modus Ponens preserva las tautologías. Este resultado, sin embargo, es más fuerte: nos dice que, si una fórmula se corresponde con nuestra noción de verdad (es una tautología), entonces es demostrable en el sistema L. Está claro que los sistemas formales con esta propiedad serán de especial interés.

Por otra parte, este Teorema recoge otra de las propiedades importantes de L:

Teorema 1.2 (Consistencia de L)**.** *No existe ninguna fórmula $\mathscr{A}$ tal que $\vdash_L \mathscr{A}$ y $\vdash_L (\neg\mathscr{A})$. (En general, un sistema con esta propiedad se dirá* consistente*.)*

Demostración. Si $\vdash_L \mathscr{A}$, por 1.1, $\mathscr{A}$ es una tautología, y por tanto $\neg\mathscr{A}$ es una contradicción, así que no puede ser un teorema. □

Si queremos que nuestros sistemas formales reflejen adecuadamente nuestra idea de verdad matemática, es importante que no se dé el caso de que una fórmula y su negación sean a la vez teoremas, para evitar contradicciones. Por tanto, será importante comprobar que el sistema formal con el que estemos trabajando sea consistente.

El sistema L tiene las dos propiedades importantes que hemos mencionado (consistencia y completitud); sin embargo, podemos encontrarnos con sistemas que no tengan dichas propiedades. El Teorema central de este trabajo versará sobre completitud y consistencia de los sistemas de primer orden, que pasamos a estudiar.

[2] Existen otras formas de definir los axiomas del sistema L; en [2, pág. 28], se da una definición distinta de L3.

[3] Es importante distinguir entre teoremas (t minúscula), que son fórmulas que tienen demostración en un sistema formal, y Teoremas (T mayúscula), que son resultados sobre los sistemas formales.

1.2. Sistemas formales de primer orden

En el sistema L usamos las proposiciones como base para construir frases más complejas y estudiar las relaciones lógicas entre ellas. Sin embargo, en algunas ocasiones esto no es suficiente y debemos descomponer las proposiciones en sujetos y predicados. Para representar este nuevo nivel de complejidad necesitaremos una nueva clase de lenguajes.

Estos lenguajes incluirán constantes individuales $a_1, a_2, ...$ que harán las veces de objetos matemáticos fijos, como por ejemplo el número 0; incluirán también letras de función f_i^n que deben entenderse como si actuaran sobre n objetos matemáticos para dar otro objeto; y por último, las letras de predicado A_i^n representarán propiedades que n objetos pueden (o no) tener. En ambos casos, el índice i es puramente para distinguir letras de función o de predicado entre sí.

La mayor diferencia con el lenguaje que hemos usado hasta ahora será el cuantificador universal $\forall$. Este símbolo debe interpretarse de forma que actúe sobre las variables $x_1, x_2...$ Por ejemplo, leeremos "$(\forall x_1)...$" como "Para todo x_1, ...".

Al igual que en el lenguaje de L, usaremos los símbolos $\vee$, $\wedge$ y $\leftrightarrow$ para acortar las fórmulas; del mismo modo, usaremos $(\exists x_i)\mathscr{A}$ como abreviatura de $\neg(\forall x_i)\neg\mathscr{A}$.

Definición. Un *lenguaje de primer orden* $\mathscr{L}$ tiene en su alfabeto:

- Variables $x_1, x_2, x_3, ...$ y algunas de las constantes individuales $a_1, a_2, a_3, ...$
- Algunas de las letras de predicado A_i^n y algunas de las letras de función f_i^n.
- Los símbolos "(", "," y ")", los conectores $\rightarrow$ y $\neg$, y el cuantificador universal $\forall$.

Las fórmulas bien formadas se definen de forma inductiva:

- Toda variable o constante individual es un *término*. Si $t_1, ..., t_n$ son términos, entonces $f_i^n(t_1, ..., t_n)$ también es un término, para cualquier i.
- Si $t_1, ..., t_n$ son términos, entonces $A_i^n(t_1, ..., t_n)$ se llama *fórmula atómica*, y es una fórmula bien formada, para cualquier i. Si $\mathscr{A}$ y $\mathscr{B}$ son fórmulas, entonces $(\mathscr{A} \rightarrow \mathscr{B})$, $(\neg\mathscr{A})$ y $(\forall x_i)\mathscr{A}$ también son fórmulas, para cualquier i.

Los términos se deben interpretar como sujetos sobre los que actúan los predicados A_i^n. Así, "$(\forall x_1)A_3^1(f_2^1(x_1))$" se lee como "Para todo x, el resultado de aplicar f_2^1 a x tiene la propiedad A_3^1". Más rigurosamente:

Definición. Una *interpretación* M de un lenguaje de primer orden $\mathscr{L}$ consiste en un conjunto no vacío D y una serie de asignaciones:

- A cada constante a_i, M le asigna un elemento de D $(a_i)^M$.
- A cada f_i^n, M le asigna una función $(f_i^n)^M : D^n \longrightarrow D$.
- A cada A_i^n, M le asigna un subconjunto $(A_i^n)^M \subseteq D^n$. (Este subconjunto se debe entender como una relación: verdadero si los n argumentos del predicado están en el subconjunto $(A_i^n)^M$, falso si no.)

En una fórmula de tipo $(\forall x_i)\mathscr{A}$, se dice que $\mathscr{A}$ es el ámbito del cuantificador $(\forall x_i)$. Se dice que una variable x_i aparece *ligada* en una fórmula si está en el ámbito de un cuantificador universal $(\forall x_i)$ con el mismo índice i; se dice *libre* si no lo está. A modo de ejemplo, en la fórmula $(\forall x_3)A_2^2(x_5, x_3)$, la variable x_3 aparece ligada y la variable x_5, libre. Una fórmula se dice *cerrada* (o *sentencia*) si no contiene variables libres.

Por otra parte, dado un término t y una fórmula $\mathscr{A}$, se dice que t es *libre para* x_i *en* $\mathscr{A}$ si al sustituir todas las x_i libres de $\mathscr{A}$ por t, ninguna de las variables de t pasaría a estar ligada; en otras palabras, si podemos sustituir x_i por t sin que cambie la estructura de $\mathscr{A}$. En el ejemplo anterior, el término $f_1^2(x_2, x_3)$

no es libre para x_5, pero el término $f_3^1(x_4)$ sí lo es. En muchas ocasiones, como en el axioma (L4) (ver más adelante), necesitaremos que un término t sea libre para una variable x_i para poder sustituirlo.

Bajo una interpretación M, las variables recorren los elementos de D. Así, una fórmula cerrada $\mathscr{A}$ se convierte en una afirmación sobre D que puede ser cierta o falsa, mientras que una fórmula no cerrada $\mathscr{B}(x_i, x_j, ...)$ contiene variables libres $x_i, x_j, ...$, y por tanto no tiene por qué tener un valor de verdad definido (pues $x_i, x_j, ...$ pueden tomar un valor en D que haga cierta a $\mathscr{B}$ y otro que no). La idea de que una fórmula cerrada sea cierta o falsa bajo una interpretación M es, por tanto, intuitiva; para una definición precisa, ver [1, pág. 47–48].

Una fórmula $\mathscr{A}$ se dice *lógicamente válida* si es cierta para toda interpretación posible, y *contradictoria* si es falsa para toda interpretación posible. La idea de verdad en una interpretación extiende a la idea de asignación de los valores de verdad en la lógica proposicional; las fórmulas lógicamente válidas son análogas, por tanto, a las tautologías de L.

De hecho, si tenemos un lenguaje de primer orden $\mathscr{L}$ y una fórmula bien formada $\mathscr{A}$ del sistema L, podemos sustituir cada letra de sentencia por una fórmula bien formada de $\mathscr{L}$ de forma consistente (esto es, si dos letras de sentencia A_i, A_j de $\mathscr{A}$ cumplen $i = j$, entonces se sustituyen por la misma fórmula). El resultado de esta sustitución es una fórmula bien formada $\mathscr{B}$ de $\mathscr{L}$. En ese caso, diremos que $\mathscr{B}$ es una instancia de $\mathscr{A}$; es claro, entonces, que toda instancia de una tautología es lógicamente válida.

Estamos listos para definir los sistemas formales con los que trabajaremos de aquí en adelante:

Definición. Un *sistema formal de primer orden K* viene dado por lo siguiente:

- Un lenguaje de primer orden, con su alfabeto y sus fórmulas bien formadas.
- Cualquier fórmula de una de las siguientes formas es un axioma de K, donde $\mathscr{A}$, $\mathscr{B}$ y $\mathscr{C}$ son fórmulas cualesquiera:

  (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) $(\neg\mathscr{B} \to \neg\mathscr{A}) \to ((\neg\mathscr{B} \to \mathscr{A}) \to \mathscr{B})$

  (L4) $(\forall x_i)\mathscr{A}(x_i) \to \mathscr{A}(t)$, donde t es un término libre para x_i en $\mathscr{A}$.

  (L5) $(\forall x_i)(\mathscr{A} \to \mathscr{B}) \to (\mathscr{A} \to (\forall x_i)\mathscr{B})$, donde $\mathscr{A}$ no contiene a x_i libre.

  Además, un sistema de primer orden puede tener otros axiomas cualesquiera, llamados *axiomas propios*. Los axiomas (L1)-(L5) reciben el nombre de *axiomas lógicos*. Si K no tiene axiomas propios, entonces K se llama un *cálculo de predicados*.

- K tiene dos reglas de inferencia:

  Modus Ponens (Abreviada MP): De $\mathscr{A} \to \mathscr{B}$ y $\mathscr{A}$ se deduce $\mathscr{B}$.

  Generalización (Abreviada Gen): De $\mathscr{A}$ se deduce $(\forall x_i)\mathscr{A}$.

No todos los sistemas de primer orden K serán consistentes; esto dependerá de los axiomas propios de K. Sin embargo, si K no tiene axiomas propios, podemos demostrar su consistencia:

Teorema 1.3 (Consistencia del cálculo de predicados)**.** *Si K no tiene axiomas propios, no existe ninguna fórmula* $\mathscr{A}$ *tal que* $\vdash_K \mathscr{A}$ *y* $\vdash_K (\neg\mathscr{A})$.

Demostración. Dada una fórmula $\mathscr{A}$, denotemos por $h[\mathscr{A}]$ a la fórmula que resulta de eliminar los cuantificadores de $\mathscr{A}$ y sustituir toda fórmula atómica de $\mathscr{A}$ por una letra de sentencia de L, sustituyendo siempre la misma letra de predicado por la misma letra de sentencia (por ejemplo, A_i^n por $A_{2^i 3^n}$, aunque hay otras formas). Entonces $h[\mathscr{A}]$ es una fórmula de L. Además, se demuestra fácilmente que, si $\mathscr{A}$ es un axioma lógico de K, entonces $\vdash_L h[\mathscr{A}]$; por tanto, si $\vdash_K \mathscr{A}$, entonces $\vdash_L h[\mathscr{A}]$, pues se puede demostrar a partir de los axiomas de L.

Se deduce que, si $\vdash_K \mathscr{B}$ y $\vdash_K (\neg\mathscr{B})$, entonces $\vdash_L h[\mathscr{B}]$ y $\vdash_L h[\neg\mathscr{B}]$, que es $\neg h[\mathscr{B}]$, lo cual no es posible por 1.2. □

Podemos demostrar un resultado análogo a 1.1; sin embargo, para ello necesitamos resultados algo más complejos, que obtendremos en el estudio de otros sistemas formales de primer orden en la siguiente sección.

1.3. Modelos y extensiones. Completitud

Si Γ es un conjunto de fórmulas de un sistema formal, un *modelo M de* Γ es una interpretación M bajo la cual todas las fórmulas de Γ son ciertas. Por otra parte, un modelo M de un sistema de primer orden K es un modelo del conjunto de sus axiomas; esto es, una interpretación bajo la cual todos los axiomas (y por extensión, los teoremas) de K son ciertos.

Se dice que un sistema K' es una *extensión* de K si todo teorema de K es un teorema de K'. En particular, dado un sistema K cuyos axiomas son Γ y una fórmula $\mathscr{A}$, el sistema con el mismo lenguaje y reglas de deducción que K y cuyos axiomas son $\Gamma \cup \{\mathscr{A}\}$ es una extensión de K, a la que llamaremos $K \cup \mathscr{A}$ (no se trata de una auténtica unión de conjuntos, es pura notación).

Diremos que un sistema K es *completo*[4] si, dada cualquier fórmula cerrada $\mathscr{A}$ de K, o se cumple que $\vdash_K \mathscr{A}$ o se cumple que $\vdash_K (\neg\mathscr{A})$.

Antes de demostrar resultados sobre completitud, veamos una herramienta que nos será necesaria para algunos resultados:

Teorema 1.4 (de Deducción en sistemas de primer orden)**.** *Si tenemos una prueba de* $\Gamma, \mathscr{A} \vdash \mathscr{B}$ *en la que no se usa la regla Gen con una variable libre de* $\mathscr{A}$ *(en particular, si* $\mathscr{A}$ *es cerrada), entonces* $\Gamma \vdash (\mathscr{A} \to \mathscr{B})$.

No daremos una demostración porque sólo nos interesa su aplicación; una prueba completa está en [1, pág. 59].

Teorema 1.5 (Lema de Lindenbaum)**.** *Si K es un sistema formal consistente, entonces existe* K' *extensión completa de K.*

Demostración. Probamos primero el siguiente hecho: si una fórmula cerrada $\mathscr{A}$ no es un teorema de K, entonces $K \cup (\neg\mathscr{A})$ es consistente. Para ello, supongamos que no lo es; entonces existe una fórmula $\mathscr{B}$ tal que $\vdash_{K\cup(\neg\mathscr{A})} \mathscr{B}$ y $\vdash_{K\cup(\neg\mathscr{A})} (\neg\mathscr{B})$. Ahora, tenemos que $\vdash_{K\cup(\neg\mathscr{A})} (\neg\mathscr{B}) \to (\mathscr{B} \to \mathscr{A})$, por la siguiente prueba y dos aplicaciones de 1.4, ya que no usamos Gen:

(1)	$\mathscr{B}$	Hipótesis
(2)	$\neg\mathscr{B}$	Hipótesis
(3)	$\mathscr{B} \to (\neg\mathscr{A} \to \mathscr{B})$	(L1)
(4)	$\neg\mathscr{A} \to \mathscr{B}$	(1), (3), MP
(5)	$\neg\mathscr{B} \to (\neg\mathscr{A} \to \neg\mathscr{B})$	(L1)
(6)	$\neg\mathscr{A} \to \neg\mathscr{B}$	(2), (4), MP
(7)	$(\neg\mathscr{A} \to \neg\mathscr{B}) \to ((\neg\mathscr{A} \to \mathscr{B}) \to \mathscr{A})$	(L3)
(8)	$(\neg\mathscr{A} \to \mathscr{B}) \to \mathscr{A}$	(6), (7), MP
(9)	$\mathscr{A}$	(4), (8), MP

Por tanto, $\mathscr{B}, \neg\mathscr{B} \vdash_{K\cup(\neg\mathscr{A})} \mathscr{A}$; pero, como $\mathscr{B}$ y $\neg\mathscr{B}$ son teoremas de $K \cup (\neg\mathscr{A})$, entonces tenemos que $\vdash_{K\cup(\neg\mathscr{A})} \mathscr{A}$. Esto nos muestra el hecho crucial de que, en un sistema inconsistente como suponemos que es $K \cup (\neg\mathscr{A})$, cualquier fórmula es un teorema.

[4] No se debe confundir la idea de sistema formal completo con los Teoremas de Completitud como 1.1, que hacen referencia al hecho de que toda fórmula que nosotros entendemos como cierta sea un teorema, esto es, demostrable.

Ahora, como $\vdash_{K\cup(\neg\mathscr{A})} \mathscr{A}$, entonces $\neg\mathscr{A} \vdash_K \mathscr{A}$ y como $\neg\mathscr{A}$ es cerrada, $\vdash_K \neg\mathscr{A} \to \mathscr{A}$ por 1.4. Como $\vdash_K (\neg\mathscr{A} \to \mathscr{A}) \to \mathscr{A}$ (es una instancia de tautología; la prueba es similar a la que se muestra arriba), por MP, $\vdash_K \mathscr{A}$, lo que contradice la hipótesis. Notar también que esto es equivalente a decir que, si $\neg\mathscr{A}$ no es un teorema, entonces $K\cup\mathscr{A}$ es consistente, pues $\mathscr{A} \leftrightarrow (\neg\neg\mathscr{A})$ es también una instancia de tautología.

Una vez que sabemos que $K\cup\mathscr{A}$ es consistente cuando $\neg\mathscr{A}$ no es un teorema de K, basta enumerar todas las fórmulas cerradas de K en una lista $\mathscr{B}_1, \mathscr{B}_2, \mathscr{B}_3, ...$ Esto es posible porque existe una aplicación inyectiva del conjunto de todas las fórmulas a los números naturales, como veremos en 2.3. Ahora, definiremos una sucesión $J_0, J_1, J_2, ...$ de sistemas formales: J_0 es K, y, para cada n, si $\vdash_{J_n} \mathscr{B}_{n+1}$, entonces J_{n+1} es J_n; si no, J_{n+1} es $J_n \cup \neg\mathscr{B}_{n+1}$. Ahora, J_0 es consistente, y en cada paso preservamos la consistencia por el resultado anterior. Si llamamos J al sistema que tiene como axiomas todos los axiomas de los J_i, entonces J es una extensión de K consistente, y J es claramente completo, pues para cada fórmula cerrada $\mathscr{A}$ tenemos que o $\mathscr{A}$ o $\neg\mathscr{A}$ es un axioma de algún J_i (pues la lista anterior contiene a todas las fórmulas cerradas), así que es un axioma de J. □

Teorema 1.6. *Todo sistema de primer orden consistente K tiene un modelo numerable (esto es, un modelo en el que el dominio D es un conjunto numerable).*

La demostración de este Teorema es algo más compleja; se basa en expandir el lenguaje de K con una cantidad numerable de constantes individuales nuevas $b_0, b_1, ...$, enumerar todas las fórmulas que sólo tienen una variable libre $\mathscr{F}_0(x_{i_0}), \mathscr{F}_1(x_{i_1}), ...$, y construir una sucesión de sistemas formales $K\cup\mathscr{G}_0\cup\mathscr{G}_1\cup...$, donde los $\mathscr{G}_i$ se construyen a partir de los $\mathscr{F}_i$ y los b_i. Después, por 1.5, existe una extensión completa T que, por construcción, tiene una cantidad numerable de términos cerrados (términos sin variables, es decir, que sólo contienen letras de función y constantes); finalmente se construye una interpretación en la que el dominio es el conjunto de términos cerrados de T. Para una demostración completa, ver [2, pág. 91–96].

Por fin podemos probar la propiedad de los cálculos de predicados que buscábamos:

Teorema 1.7 (de Completitud de Gödel)**.** *Para un cálculo de predicados K, $\vdash \mathscr{A}$ si y sólo si $\mathscr{A}$ es lógicamente válida.*

Demostración. Notar primero que $\mathscr{A}$ es lógicamente válida si y sólo si lo es su *clausura universal* $\mathscr{A}'$, esto es, la fórmula cerrada $(\forall x_{i_1})...(\forall x_{i_n})\mathscr{A}$, donde $x_{i_1}, ..., x_{i_n}$ son las variables libres de $\mathscr{A}$. Del mismo modo, $\vdash \mathscr{A}$ si y sólo si $\vdash \mathscr{A}'$.[5]

Supongamos, primero, que $\mathscr{A}$ es lógicamente válida; entonces lo es $\mathscr{A}'$. Supongamos también que $\mathscr{A}$ no es un teorema y, por tanto, tampoco $\mathscr{A}'$. Entonces sabemos que $K\cup\neg\mathscr{A}'$ es consistente, así que, por 1.6, tiene un modelo M. Por tanto, $\neg\mathscr{A}'$ es cierta en M (por ser modelo de $K\cup\neg\mathscr{A}'$), así que $\mathscr{A}'$ es falsa en M, pues es cerrada. Pero esto contradice que $\mathscr{A}'$ sea lógicamente válida; se deduce que $\vdash \mathscr{A}'$ y, por tanto, $\vdash \mathscr{A}$.

Por otra parte, al igual que en L, es sencillo comprobar que los axiomas lógicos (L1)-(L5) son lógicamente válidos, y las reglas MP y Gen conservan la validez lógica, así que todo teorema es lógicamente válido. □

Este Teorema, junto con 1.1, ilustra la utilidad de los sistemas formales. Nuestro objetivo al definir estos sistemas es tomar una idea semántica, como puede ser la veracidad o falsedad de una frase, y reducirla a un concepto puramente sintáctico como el de teorema, que depende únicamente del lenguaje y los axiomas y reglas de deducción de un sistema formal. Los Teoremas de Completitud como 1.1 y 1.7 nos muestran que en nuestro sistema formal la idea de teorema es capaz de abarcar todas las frases verdaderas que podemos expresar en nuestro lenguaje.

Antes de pasar a estudiar los sistemas con igualdad, mostramos un último resultado sobre modelos:

[5] Esto se demuestra por inducción en n, el número de variables libres de $\mathscr{A}$; una implicación usa la regla Gen n veces, y la otra usa el axioma (L4) n veces.

Teorema 1.8 (de Skolem-Löwenheim)**.** *Todo sistema de primer orden K que tenga un modelo tiene un modelo numerable.*

Demostración. Notar primero que, si K tiene un modelo M, entonces K debe ser consistente. En efecto, si no lo fuera, existiría una fórmula $\mathscr{B}$ tal que $\vdash \mathscr{B}$ y $\vdash \neg\mathscr{B}$; y, por ser M modelo, tanto $\mathscr{B}$ como $\neg\mathscr{B}$ serían ciertas bajo M, lo cual no es posible.

Una vez visto esto, 1.6 nos asegura que K tiene un modelo numerable. □

1.4. Sistemas de primer orden con igualdad

Nos interesa estudiar sistemas en los que uno de los predicados de dos argumentos tiene propiedades similares a la relación de igualdad, y podemos interpretarlo como la relación "es igual a..."; podemos suponer que este predicado es A_1^2 sin pérdida de generalidad. De ahora en adelante, abreviaremos $A_1^2(t,s)$ como $t = s$, y $\neg A_1^2(t,s)$ como $t \neq s$, para términos t, s cualesquiera.

Definición. Un sistema de primer orden K se dice sistema *con igualdad* si las fórmulas (I1), (I2) están entre sus teoremas:

(I1) $(\forall x_1)x_1 = x_1$

(I2) $(x_i = x_j) \to (\mathscr{A}(x_i) \to \mathscr{A}(x_j))$, donde x_j es libre para x_i en $\mathscr{A}$, y $\mathscr{A}(x_j)$ es el resultado de sustituir algunas (no necesariamente todas) de las x_i que ocurren libres en $\mathscr{A}$ por x_j.

Notar que, si K es un sistema con igualdad, entonces $\vdash t = t$ para cualquier término t, por el axioma (L4). De forma similar se prueban la simetría y transitividad de la igualdad: respectivamente, $\vdash (x_i = x_j \to x_j = x_i)$, y $\vdash (x_i = x_j \to (x_j = x_k \to x_i = x_k))$.

Dado que nos interesa poder interpretar el predicado A_1^2 como la igualdad, debemos particularizar las interpretaciones del lenguaje de K que tengan esta propiedad. Así, diremos que un modelo M de K es un *modelo normal* si $(A_1^2)^M$ es precisamente la relación de igualdad en el dominio D. Cabe destacar que, aunque M no sea normal, si llamamos $\sim$ a $(A_1^2)^M$, entonces $\sim$ es una relación de equivalencia, por las propiedades de simetría y transitividad vistas arriba; por tanto, podemos construir un nuevo modelo M' cuyo dominio sea $D/\sim$. Entonces M' es claramente normal, y se llamará *contracción* de M.

Los dos resultados importantes sobre modelos vistos en la anterior sección se pueden extender a sistemas con igualdad:

Teorema 1.9 (Extensión de 1.6)**.** *Todo sistema de primer orden K con igualdad que sea consistente tiene un modelo normal finito o numerable.*

Demostración. Por 1.6, K tiene un modelo numerable M. Su contracción M' es un modelo normal, y está claro que es finito o numerable pues $|D/\sim| \leq |D|$. □

Teorema 1.10 (Extensión de 1.8)**.** *Todo sistema de primer orden K con igualdad que tenga un modelo normal infinito M tiene un modelo normal numerable.*

Demostración. Añadimos al lenguaje de K constantes individuales nuevas $b_1, b_2, \ldots$ Sean $\mathscr{B}_{i,j}$ las fórmulas $(b_i \neq b_j)$, y llamemos K' a la extensión $K \cup \{\mathscr{B}_{i,j}\}$, para todo $i \neq j$. Podemos extender M a un modelo de K', puesto que M es infinito y podemos escoger los $(b_i)^M$ distintos entre sí. Puesto que K' tiene un modelo, debe ser consistente, como hemos visto en 1.8.

Ahora, como K' es consistente, por 1.9, tiene un modelo finito o numerable N. Entonces los $\mathscr{B}_{i,j}$ deben ser ciertos en N, de lo cual se deduce que los $(b_i)^N$ son distintos entre sí. Como hay una cantidad numerable de ellos, el dominio de N no puede ser finito, así que debe ser numerable. □

Capítulo 2

Teoría de números

2.1. El sistema *S*

En este capítulo nos centraremos en cuestiones formales de la teoría de números, estudiándola desde el punto de vista de los sistemas formales y utilizando los resultados que hemos visto en el capítulo 1.

Puesto que una gran parte de las matemáticas tiene su base en la aritmética de los números naturales, es lógico que esta sea nuestro punto de partida. El sistema formal que estudiaremos a lo largo de este capítulo se basará en los conocidos como "postulados de Peano". No nos será posible representar fielmente estos postulados en nuestra teoría, pues uno de ellos trata con conceptos del tipo "para cualquier propiedad P de un número...", conceptos que escapan a nuestra lógica de primer orden.[1]

Sin embargo, podemos construir un sistema de primer orden basado en los postulados de Peano, como sigue:

Definición. El *sistema formal S* es un sistema de primer orden que viene dado por lo siguiente:

- Su lenguaje $\mathscr{L}$ consiste en la letra de predicado A_1^2, las letras de función f_1^1, f_1^2 y f_2^2 (que representarán, respectivamente, las funciones sucesor, suma y producto) y la constante individual a_1. Abreviaremos $A_1^2(t,s)$ como $(t=s)$; $f_1^1(t)$, $f_1^2(t,s)$ y $f_2^2(t,s)$ como (t'), $(t+s)$ y $(t\cdot s)$ respectivamente; y a_1 como 0, omitiendo paréntesis cuando no haya lugar a confusión.

- Sus axiomas propios son las fórmulas (I1) e (I2) de los sistemas con igualdad, y además:

  (S1) $0 \neq x_1'$

  (S2) $x_1' = x_2' \to x_1 = x_2$

  (S3) $x_1 + 0 = x_1$

  (S4) $x_1 + x_2' = (x_1 + x_2)'$

  (S5) $x_1 \cdot 0 = 0$

  (S6) $x_1 \cdot x_2' = (x_1 \cdot x_2) + x_1$

  (S7) $\mathscr{A}(0) \to ((\forall x_i)(\mathscr{A}(x_i) \to \mathscr{A}(x_i')) \to (\forall x_i)\mathscr{A}(x_i))$, para cualquier fórmula $\mathscr{A}(x_i)$ de S en la que x_i aparezca libre.

Los axiomas de la forma (S7) están basados en el *axioma de inducción* de los postulados de Peano; sin embargo, no se corresponden exactamente dado que este último trata con una cantidad no numerable de propiedades, y en nuestro sistema sólo tenemos una cantidad numerable de fórmulas, como veremos en 2.3.

Por otra parte, es importante notar que cualquier fórmula que resulte de sustituir las variables por términos cualesquiera en uno de los axiomas (S1)-(S6) es un teorema. Esto se comprueba fácilmente usando Gen y el axioma (L4).

[1] Es posible estudiar ideas similares a esta mediante sistemas de *segundo orden*; en estos sistemas, los cuantificadores como $\forall$ no sólo actúan sobre variables, sino también sobre predicados.

S es claramente un sistema de primer orden con igualdad. Inmediatamente nos preguntamos si S es consistente, para lo cual ya hemos visto que basta con encontrar un modelo. La opción más evidente es la interpretación con dominio $\mathbb{N}$ en la que la constante 0 representa el 0 de los números naturales; las letras de función abreviadas con $'$, $+$ y $\cdot$ representan el sucesor, la suma y el producto; y la letra de predicado abreviada con $=$ representa la relación de igualdad. Tal interpretación recibirá el nombre de *modelo estándar*, y cualquier interpretación distinta se llamará *modelo no estándar*.

Podríamos, por tanto, razonar que dado que S tiene el modelo estándar, es claramente consistente. Sin embargo, dado que los números naturales se definen en un principio a través de unos axiomas, precisamente como los postulados de Peano, no está del todo claro que este argumento sobre la consistencia de S no sea un razonamiento circular en el que afirmamos que S sirve como su propio modelo.[2] Como consecuencia, el consenso es aceptar la consistencia de S como una afirmación cierta sin demostrar.

Por último, al igual que hemos hecho antes con el sistema L y el cálculo de predicados (ver 1.1 y 1.7), nos preguntamos: ¿es S un sistema completo, en el sentido de que las fórmulas que se corresponden a nuestra idea de certeza son demostrables? El Teorema central de este trabajo, el Teorema de Incompletitud de Gödel, prueba que S no es un sistema completo. Para poder demostrarlo necesitaremos desarrollar algo más la teoría de números formal a través de S.

Para ello, antes de pasar a la siguiente sección, introducimos algunas abreviaturas que simplificarán enormemente la notación:

- $(\exists_1 x_i)\mathscr{A}(x_i)$ significa $(\exists x_i)\mathscr{A}(x_i) \wedge (\forall x_i)(\forall x_j)(\mathscr{A}(x_i) \wedge \mathscr{A}(x_j) \rightarrow x_i = x_j)$. Esto se puede utilizar en cualquier sistema con igualdad, y se puede comprobar que representa la idea intuitiva de que existe un único objeto que cumple la propiedad que representa $\mathscr{A}$.

- $t < s$ significa $(\exists x_i)(x_i \neq 0 \wedge x_i + t = s)$. Del mismo modo, $t \leq s$ es $(t < s \vee t = s)$, etc.

- Si n es un número natural, $\overline{n}$ es el término $0'''^{\cdots\ (n\ veces)}$, esto es, la letra de función abreviada por $'$ aplicada n veces a 0. Los términos $\overline{0}$ (que es 0), $\overline{1}$, $\overline{2}$, ... se llamarán *numerales*. Es importante recordar que un numeral es un término de un sistema formal, a diferencia de un número natural, y por tanto para una fórmula $\mathscr{A}(x_i)$ tiene sentido escribir $\mathscr{A}(\overline{n})$ pero no $\mathscr{A}(n)$.

2.2. Funciones y relaciones recursivas

Una *función aritmética* de n argumentos es una aplicación cuyos argumentos y valor son números naturales, esto es, una aplicación de $\mathbb{N}^n$ en $\mathbb{N}$; el sucesor y la suma son ejemplos de funciones aritméticas. Asimismo, una *relación aritmética* de n argumentos es una relación en los números naturales, esto es, un subconjunto de $\mathbb{N}^n$.

Una relación aritmética de n argumentos R se dirá *expresable* en S si existe una fórmula $\mathscr{A}(x_1,...,x_n)$ con n variables libres tal que, para números naturales $k_1,...,k_n$ cualesquiera:

- Si $R(k_1,...,k_n)$ es cierta, entonces $\vdash_S \mathscr{A}(\overline{k}_1,...,\overline{k}_n)$.

- Si $R(k_1,...,k_n)$ es falsa, entonces $\vdash_S \neg\mathscr{A}(\overline{k}_1,...,\overline{k}_n)$.

Del mismo modo, una función aritmética de n argumentos f se dirá *representable* en S si existe una fórmula $\mathscr{A}(x_1,...,x_{n+1})$ con $n+1$ variables libres tal que, para números naturales $k_1,...,k_n,m$ cualesquiera:

- Si $f(k_1,...,k_n) = m$, entonces $\vdash_S \mathscr{A}(\overline{k}_1,...,\overline{k}_n,\overline{m})$.

- $\vdash_S (\exists_1 x_{n+1})\mathscr{A}(\overline{k}_1,...,\overline{k}_n,x_{n+1})$.

[2] Quizá un ejemplo más claro es el del sistema de primer orden ZF, o de Zermelo-Fraenkel, que definen la teoría de conjuntos. Es evidente que no podemos dar definiciones como "Un lenguaje consta de un *conjunto* de símbolos..." y apoyar la teoría de conjuntos sobre ellas.

Por ejemplo, la relación aritmética de igualdad es expresable en S por la fórmula $x_1 = x_2$, y la función sucesor es representable en S por la fórmula $x_2 = x_1'$. Observamos que podemos establecer una conexión entre relaciones y funciones aritméticas, del siguiente modo: si R es una relación aritmética de n argumentos, su *función característica* C_R es una función aritmética definida por:

$$C_R(x_1,...,x_n) = \begin{cases} 0 \text{ si } R(x_1,...,x_n) \text{ es falsa.} \\ 1 \text{ si } R(x_1,...,x_n) \text{ es cierta.} \end{cases}$$

Es inmediato ver que una relación aritmética R es expresable en S si y sólo si su función característica C_R es representable en S.

Para el estudio de las funciones aritméticas representables en S, definimos una nueva clase de funciones mediante los siguientes pasos:

1. Las siguientes funciones aritméticas se llamarán *funciones iniciales*:
   a) La función cero: $Z(x) = 0$ para todo x.
   b) La función sucesor: $N(x) = n+1$ para todo x.
   c) Las funciones proyección: $U_i^n(x_1,...,x_n) = x_i$, para todos $x_1,...,x_n$.
2. Las siguientes reglas se usan para obtener funciones nuevas a partir de otras dadas:
   Sustitución: Si $f(x_1,...,x_n) = g(h_1(x_1,...,x_n),...,h_m(x_1,...,x_n))$ para todos $x_1,...,x_n$, se dice que f se obtiene por sustitución de g y las $h_1,...,h_m$.
   Recursión: Si $f(x_1,...,x_n,0) = g(x_1,...,x_n)$ y $f(x_1,...,x_n,y+1) = h(x_1,...,x_n,y,f(x_1,...,x_n,y))$, se dice que f se obtiene por recursión de g y h, o sólo de h cuando $n = 0$. Es importante ver que f está bien definida, pues el valor para $y = 0$ lo da la función g y para $y > 0$ lo obtenemos de h y el valor anterior de f.
   Operador μ: Sea R una relación aritmética de $n+1$ argumentos. Denotamos por $\mu^y R(x_1,...,x_n,y)$ el menor número natural y tal que $R(x_1,...,x_n,y)$ es cierta, si existe. Notar que si g es una función aritmética de $n+1$ argumentos, $g(x_1,...,x_n,y) = 0$ define una relación aritmética. Si $f(x_1,...,x_n) = \mu^y(g(x_1,...,x_n,y) = 0)$, decimos que f se obtiene por el operador μ de g.
3. Una función aritmética f se dice *recursiva* si se obtiene de las funciones iniciales mediante un número finito de aplicaciones de las reglas anteriores. f se dice *recursiva primitiva* si se obtiene de las funciones iniciales utilizando sólo las reglas de sustitución y recursión. Del mismo modo, una relación aritmética es recursiva (primitiva) si su función característica es recursiva (primitiva).

A modo de ejemplo, las siguientes funciones son recursivas (de hecho, recursivas primitivas):

1. $x+y$
2. $x \cdot y$
3. $\delta(x)$ (La función inversa al sucesor si $x > 0$, 0 si no)
4. $x \ominus y$ (La función $x-y$ si $x > y$, 0 si no)
5. $|x-y|$
6. $sg(x)$ (Vale 0 si $x = 0$, 1 si no)
7. $\overline{sg}(x)$ (Vale 1 si $x = 0$, 0 si no)
8. $qt(x,y)$ (El cociente de la división de x por y)

9. $exp(x, j)$ (El exponente del j-ésimo primo en la descomposición en factores primos de x)

Por otra parte, es importante notar que, si R_1 y R_2 son relaciones recursivas (primitivas), entonces $(R_1 \to R_2)$ y $(\neg R_1)$ son recursivas (primitivas). Esto no es cierto para los cuantificadores $\forall$ y $\exists$, pero podemos introducir un nuevo símbolo: entenderemos por $(\forall y)_{y<z}R(x_1,...,x_n,y)$ la relación "para todo y, si $y < z$, entonces $R(x_1,...,x_n,y)$ es cierta". Del mismo modo definimos los símbolos $(\forall y)_{y\leq z}$, $(\exists y)_{y<z}$, etc. Las relaciones obtenidas a partir de relaciones recursivas (primitivas) a través de estos símbolos, llamados *cuantificadores acotados*, sí son recursivas (primitivas).

El siguiente Teorema muestra la importancia de las funciones recursivas en nuestro estudio:

Teorema 2.1. *Toda función recursiva es representable en S. Asimismo, toda relación recursiva es expresable en S.*

No daremos una demostración completa de este Teorema, ya que es muy extensa; se puede encontrar una en [1, pág. 143-145]. La prueba pasa por demostrar que las funciones iniciales son representables, y que las reglas de sustitución, recursión y operador μ dan funciones representables a partir de funciones representables. La segunda parte se sigue inmediatamente como corolario.

Como última nota, aunque hemos dado la definición de funciones representables y relaciones expresables para S, sólo es necesario que un sistema contenga los símbolos del lenguaje de S (posiblemente más) para poder dar dichas definiciones. Tal sistema K se dirá *sistema aritmético*, y podemos decir, por ejemplo, que una función f es representable en K, ya que un sistema aritmético contiene todos los numerales.

2.3. Numeración de Gödel

La idea central del Teorema de Gödel pasa por encontrar una forma de que las fórmulas de un sistema formal puedan, de algún modo, representar propiedades del propio sistema. Para sistemas aritméticos (en particular S), dado que las fórmulas se pueden ver como afirmaciones sobre los números naturales, este método será la numeración de Gödel, que consiste en asignar un número natural a cada fórmula mediante una aplicación g.

Esto se puede hacer de maneras diversas, pero aquí usaremos la numeración dada en [1]. La aplicación g asigna números naturales a los símbolos de cualquier lenguaje de primer orden del siguiente modo:

- $g(() = 3; g()) = 5; g(,) = 7; g(\neg) = 9; g(\to) = 11; g(\forall) = 13$
- $g(x_k) = 13 + 8k$ para $k = 1, 2, ...$
- $g(a_k) = 7 + 8k$ para $k = 1, 2, ...$ En particular, en S, $g(0) = 15$.
- $g(f_k^n) = 1 + 8(2^k 3^n)$ para $k, n \geq 1$
- $g(A_k^n) = 3 + 8(2^k 3^n)$ para $k, n \geq 1$

Ahora, si $u_0 u_1 ... u_r$ es una expresión formada por símbolos u_k, se define $g(u_0 u_1 ... u_r)$ como $p_0^{g(u_0)} p_1^{g(u_1)} ... p_r^{g(u_r)}$, donde p_j es el j-ésimo primo, empezando por $p_0 = 2$. Del mismo modo, si $e_0, ..., e_r$ son expresiones, entonces definimos $g(e_0, ..., e_r) = p_0^{g(e_0)} ... p_r^{g(e_r)}$. Diremos que el resultado de la aplicación g es el *número de Gödel* del símbolo, expresión o sucesión de expresiones correspondiente.

Como ya hemos mencionado, hay distintas formas de asignar números de Gödel, pero es importante poder invertir el proceso y decidir si un número natural dado es el número de Gödel de algún símbolo, expresión o sucesión. Claramente, el Teorema fundamental de la aritmética nos asegura que nuestro sistema de numeración de Gödel cumple con este requisito; notar que el número de Gödel de un símbolo es impar, el de una expresión es divisible por 2 una cantidad impar de veces y el de una sucesión, una cantidad par de veces.

Ahora, sea K un sistema aritmético con un lenguaje $\mathscr{L}$. Diremos que $\mathscr{L}$ es un *lenguaje recursivo* (o recursivo primitivo) si las relaciones $IC(x)$, $FL(x)$ y $PL(x)$, ciertas si x es el número de Gödel de una constante individual, una letra de función o una letra de predicado respectivamente, son recursivas (o recursivas primitivas).

Si K es un sistema con lenguaje recursivo (primitivo), entonces las siguientes funciones y relaciones son recursivas (primitivas):

Relaciones: $Exp(x)$, $Trm(x)$, $Fml(x)$, $LAx(x)$: x es el número de Gödel de una expresión en $\mathscr{L}$, un término en $\mathscr{L}$, una fórmula bien formada de $\mathscr{L}$ o un axioma lógico de K, respectivamente.

Funciones: $Num(u)$: El número de Gödel de la expresión $\overline{u}$ (esto es, $g(\overline{u})$), así como $Sub(x,u,v)$: El número de Gödel de la expresión resultante de tomar la expresión $g^{-1}(x)$, tomar todas las veces que la variable $g^{-1}(u)$ ocurre libre en x, y sustituirlas por el término $g^{-1}(v)$.[3]

Además, si la relación $PrAx(x)$ (esto es, x es el número de Gödel de un axioma propio de K) es recursiva (primitiva), entonces la relación $Pf(x,s)$ (esto es, s es el número de Gödel de una sucesión que es una prueba de la fórmula $g^{-1}(x)$ en K) es recursiva (primitiva).

El uso de la numeración de Gödel nos permite probar el siguiente resultado sobre recursividad:

Teorema 2.2. *Sea K un sistema aritmético con igualdad y con lenguaje recursivo que cumple que $PrAx(x)$ es recursiva, y además cumple que, si $\vdash \overline{m} = \overline{n}$, entonces $m = n$. En ese caso, toda función aritmética representable en K es recursiva.*

Demostración. Sea f una función aritmética de n argumentos representable en K: entonces existe una fórmula de K de $n+1$ argumentos que la representa, $\mathscr{A}(x_1,...,x_n,x_{n+1})$. Sea m el número de Gödel de $\mathscr{A}$. Construyamos la relación aritmética de $n+2$ argumentos $B(u_1,...,u_n,u_{n+1},y)$ que sea cierta si y es el número de Gödel de una prueba de $\mathscr{A}(\overline{u}_1,...,\overline{u}_{n+1})$:

$$B(u_1,...,u_{n+1},y) = Pf(Sub(...Sub(m,g(x_1),Num(u_1))...,g(x_{n+1}),Num(u_{n+1})),y)$$

Puesto que hemos construido B a partir de funciones y relaciones recursivas, es claro que B es recursiva. Ahora, si llamamos $r = f(u_1,...,u_n)$ entonces $\vdash \mathscr{A}(\overline{u}_1,...,\overline{u}_n,\overline{r})$ y por tanto existe una prueba en K de dicha fórmula. Digamos que esta prueba tiene número de Gödel j; entonces $B(u_1,...,u_n,r,j)$ por construcción. Llamemos $y = 2^r 3^j$.

Notar ahora que, si $B(u_1,...,u_n,u_{n+1},y)$, entonces $r = u_{n+1}$, porque $\mathscr{A}$ representa a f y esto implica que $\vdash (\exists_1 x)\mathscr{A}(\overline{u}_1,...,\overline{u}_n,x)$ y como $\vdash \mathscr{A}(\overline{u}_1,...,\overline{u}_n,\overline{u}_{n+1})$ tenemos $\vdash \overline{r} = \overline{u}_{n+1}$, y por hipótesis esto implica que $r = u_{n+1}$. Se deduce que:

$$r = f(u_1,...,u_n) = exp(\mu^y(B(u_1,...,u_n,exp(y,0),exp(y,1))),0)$$

Por lo que f es recursiva, ya que sabemos que B lo es. □

Como consecuencia inmediata, toda relación aritmética expresable en K es recursiva.

2.4. Incompletitud

Contamos, por fin, con todas las herramientas necesarias para probar el hecho que afirmamos en 2.1: que el sistema S no es completo. Esto requiere probar la existencia de una *sentencia indecidible*, esto es, una fórmula cerrada $\mathscr{A}$ tal que ni $\vdash \mathscr{A}$ ni $\vdash \neg\mathscr{A}$.

Con este propósito, definimos una función aritmética D, llamada *función diagonal*:

$$D(u) = Sub(u,g(x_1),Num(u))$$

[3] Aunque no es estrictamente correcto, usaremos "la expresión/sucesión/etc. $g^{-1}(x)$" para referirnos a "la expresión/sucesión/etc. que tiene como número de Gödel a x".

Si u es el número de Gödel de una fórmula $\mathscr{A}(x_1)$, la función diagonal $D(u)$ nos da el número de Gödel de la fórmula cerrada $\mathscr{A}(\overline{u})$; en cierto modo, si $\mathscr{A}$ afirma una propiedad de x_1, entonces la función diagonal nos da una fórmula que afirma que su propio número de Gödel tiene esa propiedad.

Teorema 2.3 (Lema de diagonalización)**.** *Sea K un sistema aritmético con igualdad en el cual D es representable. Entonces, para cualquier fórmula con una sola variable libre $\mathscr{B}(x_1)$, existe una fórmula cerrada $\mathscr{C}$ tal que $\vdash \mathscr{C} \leftrightarrow \mathscr{B}(\overline{g(\mathscr{C})})$. Notar que $\overline{g(\mathscr{C})}$ es el numeral correspondiente al número de Gödel de $\mathscr{C}$.*

Demostración. Puesto que D es representable en K, digamos que está representada por una fórmula $\mathscr{D}(x_1,x_2)$. Construyamos la fórmula $\mathscr{A}(x_1)$ que será $(\forall x_2)(\mathscr{D}(x_1,x_2) \leftrightarrow \mathscr{B}(x_2))$. Sea $m = g(\mathscr{A}(x_1))$, y sea $\mathscr{C}$ la fórmula cerrada $\mathscr{A}(\overline{m})$. Llamamos $q = g(\mathscr{C})$; está claro, por construcción, que $D(m) = q$, y por tanto tenemos (*):

$$(*) \qquad \vdash \mathscr{D}(\overline{m},\overline{q})$$

Además, como $\mathscr{D}$ representa a D, tenemos (**):

$$(**) \qquad \vdash (\exists_1 x_2)(\mathscr{D}(\overline{m},x_2))$$

Ahora, probamos que $\vdash \mathscr{C} \to \mathscr{B}(\overline{q})$:

(1)	$(\forall x_2)(\mathscr{D}(\overline{m},x_2) \to \mathscr{B}(x_2))$	Hipótesis
(2)	$((\forall x_2)(\mathscr{D}(\overline{m},x_2) \to \mathscr{B}(x_2)) \to (\mathscr{D}(\overline{m},\overline{q}) \to \mathscr{B}(\overline{q}))$	(L4)
(3)	$\mathscr{D}(\overline{m},\overline{q}) \to \mathscr{B}(\overline{q})$	(1), (2), MP
(4)	$\mathscr{D}(\overline{m},\overline{q})$	(*)
(5)	$\mathscr{B}(\overline{q})$	(3), (4), MP

Se deduce que $\mathscr{C} \vdash \mathscr{B}(\overline{q})$. Por el Teorema de Deducción 1.4, $\vdash \mathscr{C} \to \mathscr{B}(\overline{q})$. Por otra parte, vemos que $\vdash \mathscr{B}(\overline{q}) \to \mathscr{C}$:

(1)	$\mathscr{B}(\overline{q})$	Hipótesis
(2)	$\mathscr{D}(\overline{m},x_2)$	Hipótesis
(3)	$(\exists_1 x_2)(\mathscr{D}(\overline{m},x_2))$	(**)
(4)	$\mathscr{D}(\overline{m},\overline{q})$	(*)
(5)	$\overline{q} = x_2$	(3), (4), definición de $\exists_1$
(6)	$(\overline{q} = x_2) \to (\mathscr{B}(\overline{q}) \to \mathscr{B}(x_2))$	(I2)
(7)	$\mathscr{B}(\overline{q}) \to \mathscr{B}(x_2)$	(5), (6), MP
(8)	$\mathscr{B}(x_2)$	(1), (7), MP

Por tanto, $\mathscr{B}(\overline{q}), \mathscr{D}(\overline{m},x_2) \vdash \mathscr{B}(x_2)$. De nuevo, por 1.4, $\mathscr{B}(\overline{q}) \vdash \mathscr{D}(\overline{m},x_2) \to \mathscr{B}(x_2)$. Aplicando una vez Gen y luego 1.4 otra vez, tenemos $\vdash \mathscr{B}(\overline{q}) \to \mathscr{C}$.

Por la definición de $\leftrightarrow$, tenemos que $\vdash \mathscr{C} \leftrightarrow \mathscr{B}(\overline{q})$, como queríamos demostrar.

□

Las condiciones siguientes serán necesarias como hipótesis para el Teorema de Gödel:

1. La relación $PrAx(x)$ es recursiva.
2. $\vdash_K 0 \neq \overline{1}$.
3. Toda función aritmética recursiva es representable en K.

En particular, S y cualquier extensión de S que satisfaga 1[4] cumple las propiedades 1 a 3.

En un sistema que cumpla estas propiedades, en particular, la relación $Pf(x,s)$ (que hemos visto que es recursiva por la propiedad 1) es representable por una fórmula, digamos $\mathscr{P}(x_1,x_2)$. Sea $\mathscr{B}(x_1)$ la fórmula $(\forall x_2)\neg\mathscr{P}(x_1,x_2)$; podemos ver $\mathscr{B}$ como la afirmación de que la fórmula con número de Gödel correspondiente al numeral x_1 no tiene demostración. Ahora, por el Lema de Diagonalización 2.3, existe una sentencia $\mathscr{G}$ tal que:

$$(\diamond) \qquad \vdash_K \mathscr{G} \leftrightarrow \mathscr{B}(\overline{g(\mathscr{G})})$$

Bajo el modelo estándar, esta sentencia se interpreta como: "La fórmula $\mathscr{G}$ es cierta si y sólo si la fórmula con número de Gödel $g(\mathscr{G})$ (es decir, $\mathscr{G}$) no tiene demostración"; esto es, $\mathscr{G}$ es cierta si y sólo si no es demostrable. $\mathscr{G}$ se dice *sentencia de Gödel*, y es nuestro mejor candidato a fórmula indecidible. El Teorema de Gödel probará que efectivamente lo es; pero antes tenemos que introducir un último concepto.

Un sistema aritmético se dice *ω-consistente* si, para cada fórmula $\mathscr{A}(x)$ que contenga a x como su única variable libre tal que $\vdash \neg\mathscr{A}(\overline{n})$ para todo número natural n, no es cierto que $\vdash (\exists x)\mathscr{A}(x)$. En particular:

Teorema 2.4. *Todo sistema aritmético K que sea ω-consistente es consistente.*

Demostración. Basta recordar que, en un sistema no consistente, toda fórmula es un teorema, así que si encontramos una fórmula que no es un teorema el sistema es consistente. Ahora, si llamamos $\mathscr{A}(x)$ a $(\mathscr{B}(x)\wedge\neg\mathscr{B}(x))$ para cualquier fórmula $\mathscr{B}(x)$ que contenga a x como su única variable libre, entonces es claro que, para cada n, $\vdash \neg\mathscr{A}(\overline{n})$, por ser instancia de una tautología. Ahora, si K es ω-consistente, entonces $(\exists x)\mathscr{A}(x)$ no es un teorema, así que es consistente. □

Notar que cualquier sistema que admita como modelo al modelo estándar (en particular, S) es ω-consistente.

Teorema 2.5 (de Incompletitud de Gödel)**.** *Sea K un sistema aritmético que cumple las propiedades 1 a 3. Entonces:*

- *Si K es consistente, $\mathscr{G}$ no es un teorema de K.*
- *Si K es ω-consistente, $\neg\mathscr{G}$ no es un teorema de K.*

Por tanto, si K es ω-consistente, entonces contiene una sentencia indecidible, a saber, $\mathscr{G}$.

Demostración. Sea $q = g(\mathscr{G})$.

- Supongamos que $\vdash_K \mathscr{G}$. Sea r el número de Gödel de una prueba de $\mathscr{G}$ en K. Entonces $Pf(q,r)$ es cierto, es decir, $\vdash_K \mathscr{P}(\overline{q},\overline{r})$. Por otra parte, como $\vdash_K \mathscr{G}$, por $(\diamond)$ y MP, $\vdash \mathscr{B}(\overline{g(\mathscr{G})})$, esto es, $\vdash (\forall x_2)\neg\mathscr{P}(\overline{g(\mathscr{G})},x_2)$. Usando el axioma (L4) y MP, pues $\overline{r}$ es cerrado y por tanto libre para x_2, llegamos a $\vdash \neg\mathscr{P}(\overline{q},\overline{r})$. K es, por tanto, inconsistente.

[4] Los sistemas que satisfacen la propiedad 1 se dicen *recursivamente axiomatizables*.

- Supongamos que K es ω-consistente y que $\vdash_K \neg\mathscr{G}$. Por $(\diamond)$ llegamos a $\vdash \neg(\forall x_2)\neg\mathscr{P}(\overline{q},x_2)$ o, lo que es lo mismo, $\vdash (\exists x_2)\mathscr{P}(\overline{q},x_2)$. Por otra parte, como K es ω-consistente, es consistente y, como $\vdash_K \neg\mathscr{G}$, entonces no se da que $\vdash_K \mathscr{G}$; es decir, $Pf(q,n)$ es falso para cualquier n, pues $\mathscr{G}$ no tiene prueba. Se deduce que $\vdash \neg\mathscr{P}(\overline{q},\overline{n})$ para todo n natural. Pero, como $\vdash (\exists x_2)\mathscr{P}(\overline{q},x_2)$, K no puede ser ω-consistente, por definición.

□

Notar que, si el sistema K admite el modelo estándar, entonces la sentencia $\mathscr{G}$ es *verdadera* para dicha interpretación (pues $\mathscr{G}$ afirma que $\mathscr{G}$ no es demostrable, y en efecto no lo es por el Teorema de Gödel). Si añadimos algunas condiciones adicionales a K, podemos construir una sentencia algo más compleja $\mathscr{R}$ (llamada *sentencia de Rosser*) que nos permite cambiar la condición de ω-consistencia por la de consistencia. Este resultado se conoce como Teorema de Gödel-Rosser.

Por último, recordamos que el objetivo de los sistemas formales es reducir una idea semántica como es la veracidad de una frase al concepto puramente sintáctico de teorema. El Teorema de Incompletitud de Gödel nos muestra que cualquier sistema aritmético en el que se tengan las propiedades 1 a 3 será incapaz de atrapar por completo esta idea de veracidad. Notar que la propiedad 2 es simplemente de no-trivialidad, pues un sistema que no la cumpla sólo tiene un numeral. Por otra parte, si tratamos de añadir a nuestro sistema las fórmulas verdaderas y que no es capaz de demostrar, el sistema obtenido es completo; comprobamos que, por el Teorema de Gödel, la propiedad 1 no se puede cumplir, pues la propiedad 3 no se ve afectada por extensiones. Esto muestra el hecho crucial de que la propiedad de ser una fórmula verdadera ($Tr(x)$: x es el número de Gödel de una fórmula verdadera en el modelo estándar) no es recursiva.

A grandes rasgos, la idea de recursividad se corresponde con la de computabilidad[5]; esencialmente, que la veracidad de las fórmulas aritméticas sea computable quiere decir que es teóricamente posible diseñar un algoritmo capaz de decidir si una fórmula cualquiera dada es verdadera o no en un número finito de pasos. Una de las consecuencias más importantes del Teorema de Gödel es, por tanto, que es matemáticamente imposible construir un ordenador capaz de, dada una sentencia cualquiera de la aritmética, decidir si es verdadera o no. De hecho, ni siquiera es posible que sea capaz de decidir si una fórmula cualquiera dada es un teorema o no de un sistema aritmético que cumpla las condiciones 1 a 3, pues la propiedad de ser un teorema de tal sistema ($Th(x)$: x es el número de Gödel de un teorema) tampoco es recursiva (ver [1, pág. 169]).

[5]Esta idea recibe el nombre de Tesis de Church: ver [1, pág. 168]

Bibliografía

[1] E. MENDELSON, *Introduction to Mathematical Logic*, 3.ª ed., Wadsworth & Brooks/Cole, Belmont (California), 1987.

[2] A. G. HAMILTON, *Logic for Mathematicians*, Cambridge University Press, Cambridge, 1978.

Índice alfabético