

Capítulo 3

Teoría de la Computabilidad

Comentario 112 *La materia condensada en estas ocho secciones constituye un curso tranquilo de segundo ciclo. Los alumnos deben saber ya lógica de primer orden, haber demostrado los teoremas de completud y corrección del cálculo y es mejor que posean nociones muy elementales de Teoría de Modelos. Si esto no fuera así, podría empezarse por ello, introduciendo los conceptos de **teoría**, **teoría de un sistema** o **de una clase de sistemas**, y su simétrico, el de **modelo de un conjunto de sentencias**. También introduciremos las propiedades de **completud de teorías** y la de **axiomatizabilidad**. Se usará también el concepto de **relación definible** en un sistema.*

El objetivo fundamental no es tanto el de introducir el concepto básico de *recursividad*, sino el de trazar la interconexión profunda que con la lógica, en el sentido restrictivo de *Ciencia de la Consecuencia*, existe. Este curso yo lo titularía *Recursión para la Metalógica*. Se demostrará el teorema más famoso de nuestra disciplina, el de *incompletud de Gödel*, por lo que conviene situarse históricamente. Es de sumo interés que se destaque el vínculo con la *Teoría de Modelos*.

3.1. Concepto intuitivo: algoritmo

¿Cuándo decimos que una función es recursiva, qué significa ser recursiva?

Hay varias definiciones precisas, equivalentes entre sí, de este concepto. La noción intuitiva correspondiente a ser recursiva es la de ser efectivamente computable. Fijémonos en las funciones parciales n -arias sobre los naturales.

¿Cuándo decimos que una función parcial n -aria f sobre los naturales es efectivamente computable?

Sencillamente, cuando hay un procedimiento efectivo —esto es, un algoritmo— que la computa. Un procedimiento efectivo debe cumplir los siguientes requisi-

tos:

1. Instrucciones precisas, de longitud finita. Estas instrucciones no deben requerir astucia ninguna por parte de la persona o máquina que las siga y no deben incluir el azar —lanzar una moneda al aire y actuar según el resultado, por ejemplo—
2. Si se le da un argumento \bar{x} en el dominio de la función, debe producir $f(\bar{x})$ en un número finito de pasos de cálculo
3. Si se le da un \bar{x} que no esté en el dominio de la función, debe seguir indefinidamente o parar, pero en ningún caso producir un valor $f(\bar{x})$

Como ejemplos de funciones efectivamente computables podemos citar la suma y la multiplicación. Los procedimientos efectivos de cálculo son los que aprendimos en el *cole*. Por otra parte, toda función con dominio finito es efectivamente computable: las instrucciones pueden ser, sencillamente, la lista de los valores. Hay, por supuesto, funciones efectivamente computables más interesantes que las de estos ejemplos.

Cuando tenemos un algoritmo que determina si se cumple o no una propiedad, el conjunto o relación definido por ella se llama *decidible*. Así pues, el concepto intuitivo de computabilidad efectiva para funciones se generaliza para conjuntos. Hay algunos que son decidibles sólo a medias y los llamamos *efectivamente enumerables*: sus elementos pueden ser listados.

Nosotros no imponemos restricciones de naturaleza práctica a los procedimientos efectivos. Cuando se trata de funciones sobre los naturales, los argumentos de las funciones han de ser números naturales, pero no se restringe su tamaño. Por otra parte, aunque el número de pasos ha de ser finito, tampoco ponemos un tope. Además, no se prefija la cantidad de papel —o espacio de memoria— que haya de precisarse para realizar el cálculo. Estas consideraciones son importantes cuando comparamos la computabilidad efectiva con la práctica: la clase de las funciones efectivamente computables la obtenemos en una situación ideal en donde no importa el tiempo ni la memoria requerida.

Aunque la descripción intuitiva del concepto de función efectivamente computable que hemos dado no puede considerarse una definición, cuando se trate de mostrar que una función es efectivamente computable será suficiente con esta presentación informal. Otra cosa muy distinta es cuando se quiera demostrar que no lo es¹.

Las funciones recursivas son las efectivamente computables, pero reservamos el vocablo "*recursiva*" para el concepto matemático; es decir, el definido con precisión.

Comentario 113 *Nosotros definiremos en un principio el concepto de recursividad como ser representable en una teoría consistente y finitamente axiomatizada.*

¹Veremos en la página 311 del capítulo 11, que fue justamente el deseo de resolver problemas de indecidibilidad lo que movió a un significativo grupo de lógicos a definir la computabilidad efectiva, a introducir matemáticamente el concepto de recursividad.

tizable. En esto seguiremos fielmente el libro de Enderton [9] ya que se adapta bien al nivel de nuestra licenciatura.

Lógica y recursión

¿Por qué es la teoría de las funciones recursivas una parte de la lógica?

Claramente, si no se hubiera inventado la teoría de las funciones recursivas en lógica, se habría hecho, más tarde, en informática. Sin embargo, no fue una casualidad histórica: hay aspectos fundamentales de la lógica que requieren el concepto de procedimiento efectivo. De hecho, donde por vez primera se define con precisión fue en el artículo de Gödel de la incompletud de la lógica superior.

Pero incluso a un nivel mucho más básico la lógica emplea algoritmos; nosotros hemos establecido con claridad requisitos de *efectividad* del lenguaje y del cálculo lógico:

1. Al especificar los signos primitivos —dado un signo hay que poder determinar si es o no un signo primitivo—
2. Al definir fórmulas —dada una sucesión de signos hay que poder determinar si es o no una fórmula—
3. Al especificar los axiomas —dada una fórmula hay que poder determinar si es o no un axioma—
4. Al definir las reglas de inferencia —cuando se propone una fórmula como conclusión de una inferencia inmediata se debe poder determinar si es correcta, conforme a las reglas—

Como hemos visto, en la lógica se construyen cálculos para probar teoremas. Los cálculos han de ser tales que cualquiera pueda comprobar la corrección de la prueba mediante un procedimiento efectivo. El conjunto de las pruebas ha de ser *decidible* y el de los teoremas del cálculo *efectivamente enumerable*. Por consiguiente, si ya sabemos que un conjunto de fórmulas no es recursivamente enumerable —y para ello tendremos que haber definido matemáticamente este concepto—, también conocemos que no hay cálculo capaz de generarlas y no tendremos que hacer esfuerzos inútiles para encontrarlo.

¿Qué se entiende por conjunto efectivamente enumerable?

No hay que confundir *enumerable* con *numerable*. Un conjunto es numerable cuando se puede poner en correspondencia con el conjunto de los números naturales; cuando hay una función inyectiva del conjunto en el de los naturales, distinguiéndose entre finito-numerable e infinito-numerable. La enumerabilidad supone un procedimiento efectivo que liste en un cierto orden los elementos del conjunto. La idea es que, al menos idealmente, se pueda programar un computador que anote exactamente los elementos del conjunto en una lista. No importa que la máquina tenga que dar un número infinito de pasos, lo realmente importante es que el procedimiento pueda ser seguido mecánicamente y sin que se cuele el azar; debe ser lo que denominamos un *procedimiento efectivo* o *algoritmo*.

El concepto de enumerabilidad es más débil que el de decidibilidad. Un conjunto Δ es decidible cuando existe un procedimiento efectivo que, dada una expresión nos diga si pertenece o no al conjunto. Aquí resulta pertinente el ejemplo de la lógica proposicional y el de la de primer orden. En la primera, que es decidible, hay procedimientos para determinar si una fórmula cualquiera es o no es una tautología; las tablas de verdad, por ejemplo. En la segunda, si una deducción no sale, no sabemos si es falta de pericia o imposibilidad real. Es verdad que siempre puedo hacer una prueba de independencia semántica, pero las pruebas de independencia no son efectivas.

¿Cuál es la diferencia entre enumerable y decidible?

No todos los conjuntos enumerables son decidibles porque siguiendo el procedimiento efectivo del listado de Δ podemos encontrar las fórmulas C tales que $C \in \Delta$; pero, ¿qué pasa si $C \notin \Delta$? Por regla general, no hay respuesta, ya que el que la fórmula no haya aparecido aún en el listado no nos garantiza que no vaya a aparecer, nada nos asegura que no esté. Es por esto por lo que para que Δ sea decidible hace falta no solamente que Δ sea enumerable, sino que también su complementario lo sea. Más tarde veremos que el conjunto de las fórmulas lógicamente válidas de primer orden no es decidible.

Por otra parte, una teoría axiomatizable —mediante un conjunto decidible de axiomas, claro— tiene un conjunto efectivamente enumerable de teoremas. Si además de ser axiomatizable es completa, entonces sus teoremas formarán un conjunto decidible.

Estos son sólo unos cuantos ejemplos para subrayar que la teoría de la recursión es de la lógica una parte fundamental.

3.2. Concepto matemático: Tesis de Church

La Teoría de la recursión clásica abarca el estudio de las funciones definidas sobre los naturales, pero la teoría de la recursión actual, al ir desarrollando su potencial propio y sus métodos específicos, ha alcanzado un notable desarrollo abstracto y aplicaciones insospechadas. Los orígenes de la teoría clásica pueden hallarse en Dedekind, cuando en 1888 introduce el estudio de las funciones definibles sobre el conjunto de los números naturales usando ecuaciones y, recurrentemente, la inducción sobre los naturales que él había formulado y precisado. De ahí viene, justamente, el que se adoptara el nombre de Teoría de la recursión.

Por lo que respecta a su estadio presente, cuyo radio de acción cubre la totalidad de las funciones efectivamente computables, los orígenes hay que buscarlos en el grupo de Princeton; empezó con Church, pero si hay que atribuirle un padre, éste es Kleene. El fue quien la impulsó, definió y acotó. Suyos son los teoremas de la forma normal y el de la recursión.

Hay varias maneras equivalentes de matematizar el concepto intuitivo de algoritmo, algunas aparecieron en los años 30 para captar nociones que en un principio parecían diferentes: la primera era la caracterización de Gödel de las funciones definidas mediante recursión, la segunda era la noción de función definible con el operador λ , que Church y Kleene introdujeron, y la tercera,

era la de función computable mediante una máquina abstracta, las máquinas de Turing². Pronto se demostró que las tres nociones definían las mismas funciones.

Si bien estaba claro que todas las funciones definidas mediante cualquiera de los procesos anteriores era efectivamente computable, este resultado seguía sin servir para demostrar que un conjunto es indecidible; necesitamos la inversa. Por su propia naturaleza esta afirmación no puede ser demostrada, y si no se considera adecuado que sea una definición, hay que proporcionarle otro estatus.

Tesis de Church

Por esas mismas fechas Church se preguntó si el concepto de recursividad de Turing se correspondía con la noción intuitiva de computabilidad efectiva. Su respuesta, conocida como *tesis de Church*, es que sí. Church y Kleene definieron también el concepto de *λ -definibilidad*, que se corresponde —según demostró Turing— con el de recursividad³.

Shoenfield [20] dice que la tesis de Church surgió de manera casual, yo diría algo muy parecido, que surgió de forma natural⁴.

Aunque por tratarse de una tesis no se puede demostrar matemáticamente, hay razones poderosas para aceptarla:

1. La primera es una razón basada en la experiencia acumulada: todos los algoritmos conocidos son recursivos.
2. La segunda está basada en la equivalencia entre los tres conceptos introducidos para establecer la noción de algoritmo: resulta tranquilizador que desde planteamientos tan distintos se lleguen a definir exactamente las mismas funciones.
3. La tercera, que fue la que sin duda convenció a Gödel, está relacionada con el análisis que Turing hizo del concepto de computabilidad; no se interesa tanto por la naturaleza de las funciones computables como por el propio proceso de computación. Las funciones se construyen a partir de funciones elementalísimas de las que no cabe duda de su carácter algorítmico mediante procesos que tampoco plantean dudas al respecto.
4. De hecho, el argumento anterior es muy similar al dado por Church al que nos referiremos en 11.6.
5. Finalmente, la razón que convenció a Kleene: la del fracaso del argumento diagonal⁵.

²La versión mediante máquinas imaginarias fue concebida por Turing en 1936. Su trabajo es anterior a las computadoras e influyó poderosamente en la creación de éstas.

³La sección 11.6 está enteramente dedicada a dilucidar cómo actuó el cálculo lambda en la destilación de estos conceptos. Yo confío en su alquimia, llegando incluso a pensar que el cálculo lambda fue la piedra filosofal de Church.

⁴Lo veremos con detalle en la página 312, cuando hayamos introducido el cálculo lambda.

⁵La extrañeza ante semejante aserto, nos movió a Enrique Alonso y a mí a escribir el artículo *Diagonalization and Church's Thesis: Kleene's Homework*.

Ahora se podía demostrar que un conjunto no era decidible pues bastaba con mostrar que no era recursiva la función que cifraba la pertenencia a él. Esto lo hizo Church en 1936 para un problema en el cálculo lambda. En el mismo año, apoyándose en los resultados de Gödel, Church demostró que si fuera decidible la lógica de primer orden, también lo sería el del problema anterior, que había sido resuelto negativamente.

3.3. Enumerabilidad

Comentario 114 *El objetivo fundamental de esta sección sería probar el teorema de enumerabilidad para la lógica de primer orden; es decir, demostrar que el conjunto de las fórmulas lógicamente válidas de esta lógica es efectivamente enumerable. Además de eso se vería la relación existente entre enumerabilidad, axiomatizabilidad y decidibilidad en el caso de teorías de primer orden.*

El teorema de enumerabilidad es una consecuencia del de completud para la lógica de primer orden. Dicho teorema afirma que el conjunto de las fórmulas lógicamente válidas de primer orden es efectivamente enumerable.

La demostración del teorema de enumerabilidad es fácil. Dado que tenemos un cálculo deductivo que nos permite obtener como teoremas todas las fórmulas válidas y sólo ellas —teoremas de corrección y completud del cálculo—, sólo hay que hacer un programa que produzca sistemáticamente todas las deducciones.

Aquí pueden demostrarse las implicaciones precisas para completar el esquema del capítulo precedente, el de la relación entre distintas propiedades de una teoría (ver figura: 2.10) :

Nociones fundamentales

Una vez introducidos los conceptos de *procedimiento efectivo*, *conjunto efectivamente enumerable* y *conjunto decidible*, se verá:

Proposición 115 *Existe un procedimiento efectivo que, dada una fila de signos, nos dice si es una fórmula o no.*

Teorema 116 *Un conjunto de expresiones es decidible si y sólo si tanto él como su complementario —respecto del conjunto de todas las expresiones— son efectivamente enumerables.*

Se introducirá también el concepto de *lenguaje razonable*; es decir, cuando sus constantes forman un conjunto efectivamente enumerable y los relatores y funtores n-arios constituyen conjuntos decidibles.

Teorema 117 *(Enumerabilidad) VAL es efectivamente enumerable*

Lo esencial en la demostración del teorema de enumerabilidad es que tanto los axiomas del cálculo —en su defecto, reglas sin premisas— como el de las deducciones en él son decidibles. Basándonos en este hecho se prueba que los

teoremas del cálculo forman un conjunto efectivamente enumerable. Utilizando completud y corrección se prueba que también las fórmulas válidas forman un conjunto efectivamente enumerable. Como corolario de este teorema demostramos que siempre que tengamos un conjunto decidable de fórmulas Γ el conjunto de sus teoremas

$$\{C \mid \Gamma \vdash C\}$$

o, de forma equivalente, el de sus consecuencias

$$\mathbf{CON}(\Gamma) = \{C \mid \Gamma \vDash C\}$$

es efectivamente enumerable.

Otro corolario del teorema de enumerabilidad es que si Γ es decidable, el lenguaje es razonable y para cada sentencia C o bien $\Gamma \vDash C$ o bien $\Gamma \vDash \neg C$, entonces el conjunto de las sentencias que se siguen de Γ es decidable.

Teorías

Utilizamos los conceptos de teoría, teoría de un sistema, teoría de una clase de sistemas y modelo de un conjunto de sentencias definidos en el capítulo precedente junto a los de axiomatizabilidad y axiomatizabilidad finita, y también el de completud de teorías. Se demuestra lo siguiente:

Teorema 118 *Si $\mathbf{CON}(\Sigma)$ es finitamente satisfacible, entonces hay un subconjunto finito suyo $\Sigma_0 \subseteq \Sigma$ tal que $\mathbf{CON}(\Sigma_0) = \mathbf{CON}(\Sigma)$. —Haciendo uso del teorema de compacidad—*

Hecho eso, los corolarios mencionados anteriormente pueden reescribirse:

Corolario 119 *Una teoría axiomatizable en un lenguaje razonable es efectivamente enumerable.*

Corolario 120 *Una teoría completa y axiomatizable en un lenguaje razonable es decidable.*

3.4. El sistema \mathcal{N}_s

En esta sección estudiamos el sistema

$$\mathcal{N}_s = \langle \mathbb{N}, 0, s \rangle$$

formado por los números naturales, el cero y la función del siguiente. La teoría de este sistema $Th(\mathcal{N}_s)$ resulta ser axiomatizable —los axiomas los recogemos en Nat_s —, siendo dicha axiomatización completa y decidable, pero no categórica.

Recordaremos no sólo el concepto de completud de teorías, sino también las definiciones equivalentes de dicho concepto que vamos a manejar y algunos de los procedimientos comunes para demostrarlo; distinguiremos completud de

categoricidad e introduciremos el concepto de λ -categoricidad⁶. En especial, en este tema para demostrar que $\mathbf{CON}(Nat_s)$ es completa vamos a utilizar el denominado test de Vaught⁷.

La teoría del sistema \mathcal{N}_s —es decir, las sentencias verdaderas en \mathcal{N}_s — es axiomatizable y la axiomatización es completa. Los axiomas son las sentencias del conjunto infinito Nat_s formado por:

1. El cero no es el siguiente de ningún número

$$C_1 := \forall x \sigma x \neq c$$

2. La función del siguiente es inyectiva

$$C_2 := \forall xy (\sigma x = \sigma y \rightarrow x = y)$$

3. Todo número distinto del cero es el siguiente de otro

$$C_3 := \forall y (y \neq c \rightarrow \exists x y = \sigma x)$$

4. No hay ciclos de longitud 1

$$C_{4,1} := \forall x \sigma x \neq x$$

5. No hay ciclos de longitud 2

$$C_{4,2} := \forall x \sigma \sigma x \neq x$$

en general, para cada número natural n ,

6. no hay ciclos de longitud n

$$C_{4,n} := \forall x \sigma^{(n)} \dots \sigma x \neq x$$

Evidentemente, \mathcal{N}_s es modelo de Nat_s pues en \mathcal{N}_s el cero no es el siguiente de ningún número, la función del siguiente es inyectiva, todo número distinto de cero es el siguiente de otro y no hay ciclos en \mathcal{N}_s . Por consiguiente

$$\mathbf{CON}(Nat_s) \subseteq Th(\mathcal{N}_s)$$

Para demostrar que Nat_s axiomatiza $Th(\mathcal{N}_s)$ lo que hacemos es ver que

$$Th(\mathcal{N}_s) = \mathbf{CON}(Nat_s)$$

⁶Es decir, daremos un breve repaso a los primeros apartados del capítulo 7 de mi libro de *Teoría de Modelos*.

⁷Haría falta recordar la prueba de dicho teorema, que está basada en el de Löwenheim-Skolem

Para demostrar que vale la igualdad lo que hacemos es ver que $\mathbf{CON}(\mathcal{N}_S)$ es una teoría completa. La razón es que una vez demostrado que $\mathbf{CON}(\mathcal{N}_S)$ es completa, es inmediato ver que

$$Th(\mathcal{N}_S) \subseteq \mathbf{CON}(\mathcal{N}_S)$$

pues si hubiera una sentencia C de $Th(\mathcal{N}_S)$ que no estuviera en $\mathbf{CON}(\mathcal{N}_S)$, por ser ésta completa $\neg C \in \mathbf{CON}(\mathcal{N}_S)$. Pero, puesto que

$$\mathbf{CON}(\mathcal{N}_S) \subseteq T(\mathcal{N}_S)$$

también $\neg C \in Th(\mathcal{N}_S)$. Esto último es imposible, un mismo sistema no puede ser simultáneamente modelo de una sentencia y de su negación.

¿Cómo demostrar que $\mathbf{CON}(\mathcal{N}_S)$ es completa?

De hecho, hay dos formas sencillas de hacerlo: Una es aplicando el test de Vaught —es decir, demostrando que la teoría no tiene modelos finitos y que los de la misma cardinalidad son isomorfos—; la otra es demostrar que admite eliminación de cuantificadores —es decir, que dada una fórmula C cualquiera, es posible encontrar una fórmula D sin cuantificadores tal que

$$\mathcal{N}_S \vdash C \leftrightarrow D$$

Para poder aplicar el test de Vaught, necesitaremos demostrar que $\mathbf{CON}(\mathcal{N}_S)$ no tiene modelos finitos y que los de la misma cardinalidad son isomorfos. Esto no es difícil de probar utilizando la noción de \mathbb{Z} -cadena. Veamos qué son las denominadas \mathbb{Z} -cadenas.

¿Cómo ha de ser un sistema $\mathcal{A} = \langle \mathbf{A}, c^{\mathcal{A}}, \sigma^{\mathcal{A}} \rangle$ para ser modelo de \mathcal{N}_S ?

1. En primer lugar, $\sigma^{\mathcal{A}}$ tiene que ser una función biyectiva de \mathbf{A} en $\mathbf{A} - \{c^{\mathcal{A}}\}$. Esto se sigue de los tres primeros axiomas de \mathcal{N}_S .
2. Por el axioma $C_{4,n}$ sabemos que no hay ciclos de longitud n . En \mathbf{A} tiene que haber una parte estándar,

$$c^{\mathcal{A}} \rightarrow \sigma^{\mathcal{A}}(c^{\mathcal{A}}) \rightarrow \sigma^{\mathcal{A}}(\sigma^{\mathcal{A}}(c^{\mathcal{A}})) \rightarrow \dots$$

de puntos distintos. En \mathbf{A} podría haber también otros puntos —por ejemplo \mathbf{a} —; caso de haberlos, tienen que tener sus siguientes. Pero, puesto que por C_3 todo número que no sea el cero tiene un predecesor, y por C_2 sabemos que es único, \mathbf{a} estará situado en

$$\dots * \rightarrow * \rightarrow \mathbf{a} \rightarrow \sigma^{\mathcal{A}}(\mathbf{a}) \rightarrow \sigma^{\mathcal{A}}(\sigma^{\mathcal{A}}(\mathbf{a})) \rightarrow \dots$$

a las que llamamos \mathbb{Z} -cadenas por su semejanza con los enteros.

En un modelo de \mathcal{N}_S puede haber cualquier número de \mathbb{Z} -cadenas, siendo todas disjuntas y también disjuntas de la parte estándar.

Por otra parte, todo sistema formado de una parte estándar y de cualquier número de \mathbb{Z} -cadenas es un modelo de Nat_s .

La noción de \mathbb{Z} -cadena nos permite demostrar con facilidad que dos modelos cualesquiera de Nat_s con el mismo número de \mathbb{Z} -cadenas son isomorfos; a partir de ahí demostramos que dos modelos supernumerables de Nat_s de la misma cardinalidad son isomorfos. Ahora podemos aplicar el test de Vaught y concluir que $\mathbf{CON}(Nat_s)$ es una teoría completa.

Sabiendo que $Th(\mathcal{N}_s)$ es axiomatizable y su axiomatización es completa, aplicando los resultados del tema anterior, es fácil ver que es decidible. Esta prueba de la decidibilidad no es constructiva y habrá casos en los que estemos interesados no sólo en saber que es decidible, sino en conocer el método para poder aplicarlo en casos concretos.

Eliminación de cuantificadores

Hemos dicho que $Th(\mathcal{N}_s)$ es decidible.

¿Cuál es el procedimiento de decisión?

Para esto es para lo que es interesante demostrar que la teoría admite eliminación de cuantificadores⁸; es decir, que toda fórmula de su lenguaje es equivalente en el marco de la teoría a una sin cuantificadores. El procedimiento de eliminación de cuantificadores es en sí mismo un procedimiento de decisión.

También se demostrará que, como consecuencia de admitir eliminación de cuantificadores, las relaciones definibles en \mathcal{N}_s lo son mediante fórmulas sin ellos y por lo tanto es sencillo comprobar que los únicos subconjuntos de \mathbb{N} definibles en \mathcal{N}_s son los finitos y los cofinitos.

3.5. Relaciones y funciones representables

La Subteoría $\mathbf{CON}(Nat_E)$

En este tema introduzco las nociones de relación y función representable en una teoría y pruebo algunos resultados para el caso particular de la teoría $\mathbf{CON}(Nat_E)$, una subteoría finitamente axiomatizable de $Th(\mathcal{N})$. Aunque, de hecho, las nociones las introduzco para demostrar en la sección 3.7 los teoremas de incompletud de Gödel y de indecidibilidad de Church, creo que son lo suficientemente interesantes como para merecer un tratamiento detallado.

Considero el sistema estándar de los naturales con las operaciones de suma, producto, exponenciación, así como el cero, la función del siguiente y la relación de orden. Sabemos que $Th(\mathcal{N})$ no es ni decidible ni axiomatizable, pero hay una subteoría suya $\mathbf{CON}(Nat_E)$ que siendo finitamente axiomatizable, permite representar, en un sentido que precisaré, hechos sobre conjuntos decidibles.

Las nociones de representabilidad y definibilidad describen ambas relaciones sobre los naturales mediante fórmulas; la diferencia es que en la representabilidad lo que interesa es que la fórmula sea deducible a partir de los axiomas de

⁸La demostración detallada está en mi libro de *Teoría de Modelos*, en la página 264 de la edición española.

Nat_E y en la definibilidad que sea verdadera en \mathcal{N} . Veremos que representabilidad en $\mathbf{CON}(Nat_E)$ implica definibilidad en \mathcal{N} . El interés que tiene la noción de representabilidad es que también implica la de decidibilidad. Concretamente, se demuestra que toda relación representable en una teoría consistente y axiomatizable es decidible. Como caso particular, cuando la teoría sea finitamente axiomatizable, la relación representable en ella seguirá siendo decidible.

¿Vale el recíproco?

Es decir, *¿es equivalente el ser decidible al ser representable en una teoría consistente y finitamente axiomatizable?*

Se trata de una versión de la tesis de Church, que aceptamos. Dicha tesis no es una sentencia matemática, ni tan siquiera metamatemática que pueda ser demostrada, es más bien el criterio de que la noción intuitiva de decidibilidad puede formularse correctamente si la identificamos con la de representabilidad en una teoría consistente y finitamente axiomatizable.

Pues bien, si a todo esto lo llamamos ser recursiva —es decir, identificamos ser recursiva con ser representable en teoría consistente y finitamente axiomatizable— podemos expresar la tesis de Church así: una relación es decidible si y sólo si es recursiva. Ahora la recursividad se convierte en la precisión de la noción intuitiva de decidibilidad.

Hay que insistir aquí en la diferencia entre la computabilidad teórica y la práctica. Y señalar que para que el algoritmo se pueda implementar en un ordenador real, con una memoria limitada, que produzca resultados en un tiempo razonable, no basta con saber que es decidible, sino también cómo crece la complejidad del procedimiento con la complejidad de la fórmula.

Presentación de Nat_E

El sistema en estudio es $\mathcal{N} = \langle \mathbb{N}, 0, s, +, \cdot, E, < \rangle$. En realidad bastaría con tener suma y producto⁹ pues el resto de las operaciones pueden definirse en función de ellas; no obstante, se simplifican las pruebas teniéndolos.

Los axiomas de Nat_E son los siguientes:

1. C_1 y C_2 de Nat_s
2. $B_1 := \forall xy(x < sy \leftrightarrow x \leq y)$
3. $B_2 := \forall x \neg x < c$
4. $B_3 := \forall xy(x < y \vee x = y \vee y < x)$

junto con las definiciones habituales de suma, producto y exponenciación.

Puesto que \mathcal{N} es modelo de Nat_E tenemos que $\mathbf{CON}(Nat_E) \subseteq Th(\mathcal{N})$. Demostraremos algunos teoremas de esta teoría y veremos que todas las sentencias sin cuantificadores que son verdaderas en \mathcal{N} son deducibles en Nat_E . Por supuesto, no se puede demostrar que admita eliminación de cuantificadores.

⁹De hecho, en el apéndice 3.10 demuestro el teorema de indecidibilidad de Church apoyándome en la propiedad homónima de la llamada Aritmética de Robinson, que no contiene exponenciación ni orden.

Relaciones representables

Recordaremos el concepto de relación definible en un sistema del capítulo anterior e introduciremos el de relación representable en una teoría y el de relación numeralmente determinada por Nat_E .

Definición 121 Sea \mathbf{R} una relación n -aria sobre \mathbb{N} —esto es, $\mathbf{R} \subseteq \mathbb{N}^n$ —. Decimos que \mathbf{R} es **representable en** Nat_E si hay una fórmula C del lenguaje de la aritmética con a lo sumo x_1, \dots, x_n libres, tal que:

$$\begin{aligned} \text{Si } \langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle \in \mathbf{R} \text{ entonces } C\left(\frac{\sigma^{\mathbf{x}_1}}{x_1}, \dots, \frac{\sigma^{\mathbf{x}_n}}{x_n}\right) \in \mathbf{CON}(Nat_E) \\ \text{Si } \langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle \notin \mathbf{R} \text{ entonces } \neg C\left(\frac{\sigma^{\mathbf{x}_1}}{x_1}, \dots, \frac{\sigma^{\mathbf{x}_n}}{x_n}\right) \in \mathbf{CON}(Nat_E) \end{aligned}$$

Definición 122 Una fórmula C del lenguaje de la aritmética con a lo sumo x_1, \dots, x_n libres, está **numeralmente determinada** por Nat_E si para cada n -tupla $\langle \mathbf{x}_1, \dots, \mathbf{x}_n \rangle \in \mathbb{N}$ se cumple una de estas dos cosas:

$$\begin{aligned} Nat_E \vdash C(\sigma^{\mathbf{x}_1}c, \dots, \sigma^{\mathbf{x}_n}c) \\ Nat_E \vdash \neg C(\sigma^{\mathbf{x}_1}c, \dots, \sigma^{\mathbf{x}_n}c) \end{aligned}$$

Fácilmente se demuestra que una fórmula C representa una relación \mathbf{R} en $\mathbf{CON}(Nat_E)$ si y sólo si:

1. C esta numeralmente determinada por Nat_E .
2. C define a \mathbf{R} en \mathcal{N} .

Así mismo se demuestra el teorema principal, que establece que las relaciones representables en una teoría consistente y axiomatizables son decidibles. De este teorema se sigue lo siguiente

Corolario 123 Toda relación representable en una teoría consistente y finitamente axiomatizable es decidible.

3.6. Aritmetización de la sintaxis

El objetivo de esta sección es introducir el instrumental necesario para demostrar en la próxima el teorema de incompletud de Gödel; es decir, introduciremos la denominada gödelización del lenguaje.

En especial, veremos:

Primero : Ciertas afirmaciones sobre fórmulas pueden convertirse en afirmaciones sobre números.

Segundo : Algunas de estas afirmaciones sobre números (hechas en meta-lenguaje) pueden retrotraducirse al lenguaje formal. Además, la teoría $\mathbf{CON}(Nat_E)$ es suficientemente potente como para demostrar en ella las traducciones obtenidas.

Esto nos permitirá construir fórmulas que aunque aparentemente expresan proposiciones sobre números, indirectamente hablan de fórmulas, incluso de sí mismas. En la sección siguiente explotaremos esta capacidad autorreflexiva al demostrar resultados fundamentales sobre indefinibilidad e indecidibilidad.

Para dar cuenta del apartado **Primero** lo que haremos es asignar números a los signos del lenguaje y, sobre ellos, números de Gödel a cada expresión, cada conjunto de expresiones e incluso a cada sucesión de expresiones —como por ejemplo, una deducción—. Lo más interesante de esta asignación de números es que es biunívoca; no sólo sabemos asignar un número de Gödel a cada expresión, sino también, dado un número de Gödel, sabemos exactamente a qué fórmula corresponde. Esta última propiedad se debe a que los números de Gödel son productos de potencias de primos y la descomposición de un número en factores primos es única.

Para desarrollar el apartado **Segundo** lo que hacemos es demostrar que algunas relaciones y funciones definidas sobre números naturales, que tienen que ver con números de Gödel, son representables en $\mathbf{CON}(Nat_E)$. En especial, veremos que el conjunto de los números de Gödel de las deducciones a partir de un conjunto finito de axiomas es representable en $\mathbf{CON}(Nat_E)$.

Funciones representable: Catálogo

Utilizando el concepto de función representable en $\mathbf{CON}(Nat_E)$, estableceremos los resultados siguientes:

1. La función del siguiente es representable
2. Toda función constante es representable
3. La función proyección es representable
4. Adición, multiplicación y exponenciación son representables

Además de esto demostramos que la clase de las funciones representables está cerrada bajo composición y que es posible encontrar una fórmula que represente al denominado *operador mínimo-cero*.

Números de Gödel

En primer lugar, se asignan números distintos a todos los signos del lenguaje aritmético; es decir, a los signos lógicos, a las variables y a los funtores y relatores específicos que representan al: cero, siguiente, suma, producto, exponenciación, menor que. Sea h dicha asignación de números a los signos del lenguaje.

Además de eso, dada una expresión cualquiera

$$\varepsilon := \varrho_0 \varrho_1 \dots \varrho_n$$

definimos su correspondiente número de Gödel $\#\varepsilon$ así:

$$\#\varepsilon = \langle h(\varrho_0) \dots h(\varrho_n) \rangle = p_0^{h(\varrho_0)+1} \dots p_n^{h(\varrho_n)+1}$$

Por otra parte, a un conjunto de expresiones le asignamos el conjunto de sus números de Gödel y a una sucesión de expresiones le asignamos también un número de Gödel; a saber,

$$G(\langle \alpha_1 \dots \alpha_n \rangle) = \langle \#\alpha_1 \dots \#\alpha_n \rangle$$

Hecho esto, se debe probar que son representables en $\mathbf{CON}(Nat_E)$ una gran cantidad de funciones, relacionadas con números de Gödel, que necesitaremos en la demostración del teorema de incompletud. Entre otras, las funciones codificadoras y decodificadoras, los conjuntos de los números de los axiomas y de las deducciones, etc.

Teoremas

Utilizando la maquinaria de la aritmetización se puede demostrar que toda relación recursiva es representable en $\mathbf{CON}(Nat_E)$. Como se recordará, nosotros habíamos definido el ser recursiva diciendo que tenía que ser representable en alguna teoría consistente y finitamente axiomatizable. Lo que aquí se demuestra es que nos podemos limitar siempre a $\mathbf{CON}(Nat_E)$, pues del hecho de que lo sea en cualquier teoría de las características antes dichas, se sigue que lo es en la prepotente teoría $\mathbf{CON}(Nat_E)$.

Teorema 124 *Una relación es recursiva si y sólo si es representable en $\mathbf{CON}(Nat_E)$*

Corolario 125 *Toda relación recursiva es definible en \mathcal{N}*

Del teorema anterior se sigue fácilmente la equivalencia entre recursividad —ser representable en una teoría consistente y finitamente axiomatizable— y ser representable en $\mathbf{CON}(Nat_E)$, ya que $\mathbf{CON}(Nat_E)$ es consistente y finitamente axiomatizable.

Además, usando resultados anteriores, se verá que toda relación recursiva es definible en \mathcal{N} .

Para finalizar demostraremos que si el conjunto de los números de Gödel de un conjunto de axiomas es recursivo, y sus consecuencias semánticas constituyen una teoría completa, entonces el conjunto de sus números de Gödel es también recursivo.

Teorema 126 *Si $\#\Gamma$ es recursivo y $\mathbf{CON}(\Gamma)$ es completo, entonces $\#\mathbf{CON}(\Gamma)$ es recursivo.*

Teorema de Tarski, Gödel y Church

Mostraremos en primer lugar el teorema del punto fijo. A partir de él es fácil demostrar el teorema de indefinibilidad de Tarski.

Teorema 128 (*indefinibilidad de Tarski*) *El conjunto $\#Th(\mathcal{N})$ no es definible en \mathcal{N} .*

Demostración. *Para demostrar este teorema supondremos que el conjunto $\#Th(\mathcal{N})$ fuera definible mediante una fórmula B . Aplicando el teorema del punto fijo a $\neg B$ encontramos una fórmula D equivalente a $\neg B(\sigma^{\#D}c)$. En particular*

$$\mathcal{N} \Vdash D \leftrightarrow \neg B(\sigma^{\#D}c) \quad (3.2)$$

¿Qué dice esta fórmula?

Si B define $\#Th(\mathcal{N})$, entonces D dice indirectamente “yo soy falsa”. Entonces, D es verdadera en \mathcal{N} si y sólo si $B(\sigma^{\#D}c)$ es falsa en \mathcal{N}

$$\mathcal{N} \Vdash D \text{ sys } \mathcal{N} \not\Vdash B(\sigma^{\#D}c)$$

De donde fácilmente se obtiene una contradicción tanto cuando D sea verdadera en \mathcal{N} como en el caso en que sea falsa. Por consiguiente, B no puede definir a $\#Th(\mathcal{N})$. ■

Corolario 129 *$\#Th(\mathcal{N})$ no es recursivo.*

Demostración. *Porque los conjuntos recursivos son definibles en \mathcal{N} ■*

Aquí se podrá utilizar el teorema de indefinibilidad de Tarski para demostrar el teorema de incompletud de Gödel, que expresaremos así:

Teorema 130 (*Incompletud de Gödel*) *Si $\Gamma \subseteq Th(\mathcal{N})$ y $\#\Gamma$ es recursivo, entonces $\mathbf{CON}(\Gamma)$ no es una teoría completa. —Esto es, no hay una axiomatización completa de $Th(\mathcal{N})$*

Demostración. *Si $\Gamma \subseteq Th(\mathcal{N})$ también $\mathbf{CON}(\Gamma) \subseteq Th(\mathcal{N})$. Si fuera una teoría completa, valdría la igualdad. Pero, si fuera completa, $\#\mathbf{CON}(\Gamma)$ sería recursivo y sabemos que $\#Th(\mathcal{N})$ no es recursivo. ■*

De hecho, el enunciado original del Teorema de Gödel no es éste y en su demostración no se utiliza el teorema de Tarski. Él en vez de definir a las funciones recursivas en la forma en que lo hacemos nosotros, las definió directamente, como aquellas que se obtienen por composición y recursión a partir de unas muy simples, las denominadas recursivas primitivas.

Esta demostración tiene la ventaja de ser más simple y, desde el punto de vista docente, se puede abordar directamente después de un curso de lógica de primer orden.

Para finalizar este apartado se demuestra el teorema de la indecidibilidad de $\mathbf{CON}(Nat_E)$ y el teorema de Church (1936) que afirma que el conjunto de las sentencias válidas en el lenguaje de \mathcal{N} no es recursivo, puesto que el de sus números de Gödel no lo es. La idea es probar:

Lema 131 *Si*

$$\# \text{CON}(\text{Nat}_E)$$

es recursivo, entonces

$$\# \text{CON}(\text{Nat}_E \cup \{D\})$$

también lo es

Teorema 132 (*indecidibilidad fuerte de $\# \text{CON}(\text{Nat}_E)$*). *Sea Γ una teoría —esto es, cerrada bajo consecuencia—.*

Si $\Gamma \cup \# \text{CON}(\text{Nat}_E)$ es consistente, entonces $\# \Gamma$ no es recursivo.

Corolario 133 *Si $\# \Gamma$ es recursivo y $\Gamma \cup \text{Nat}_E$ es consistente, entonces $\text{CON}(\Gamma)$ no es una teoría completa.*

Este corolario es una versión del teorema de Gödel en donde las sentencias en vez de ser verdaderas en \mathcal{N} son consistentes con Nat_E .

Teorema 134 (*indecidibilidad de Church¹⁰*) *El conjunto de los números de Gödel de las fórmulas válidas (en el lenguaje de la aritmética) no es recursivo.*

Demostración. *En el teorema de la indecidibilidad fuerte de $\# \text{CON}(\text{Nat}_E)$ tomamos como conjunto Γ al de las fórmulas válidas. ■*

En cualquier caso, haremos que se perciba el clima emocional en el que el teorema de Gödel vino a caer¹¹. Desde la publicación de los trabajos de Frege, y sobre todo de Russell y Whitehead, los lógicos pensaban que toda demostración intuitivamente correcta podía reconstruirse en el cálculo lógico. También se pensaba que añadiendo axiomas adecuados al cálculo podían construirse teorías —en el sentido lógico— en las que toda sentencia matemática intuitivamente correcta podría deducirse. Como ya comenté en la sección 1.1.2, la idea de Hilbert era explotar al máximo la naturaleza finita de las pruebas para proporcionar una fundamentación de la matemática. El proceso consistía en axiomatizarla y demostrar que la teoría es: *Consistente*, *Completa* y *Decidible*.

No es de extrañar que en este clima el teorema de Gödel fuera considerado como la prueba del fracaso del programa de Hilbert.

3.8. Máquinas de Turing

Hay varios caminos distintos para llegar a la noción de recursividad. Una versión mediante máquinas computadoras imaginarias fue concebido por Turing en 1936.

Intuitivamente una máquina de Turing consiste en una máquina abstracta (una caja negra), una cabeza lectora, un mecanismo de escritura y una cinta infinita en ambas direcciones. La máquina puede estar en cualquiera de los

¹⁰Una prueba más detallada de este teorema aparece en el apéndice 3.10.

¹¹A este respecto el artículo de Gandy [12] es fantástico.

$v_j = 0$	escribir 0 en el recuadro
$v_j = 1$	escribir 1 en el recuadro
$v_j = 2$	mover la cinta un cuadro hacia la derecha
$v_j = 3$	mover la cinta un cuadro hacia la izquierda
$v_j = 4$	parar

Es fácil *construir* algunas máquinas sencillas, tales como la máquina de ir hacia la izquierda —se va un cuadro hacia la izquierda y se para—, la de ir hacia la derecha, la de sumar, la de restar, etc.

Cuando se introduce el concepto de función recursiva primitiva se demuestra que todas ellas son computables mediante máquinas de Turing; es decir:

Primero : Todas las funciones iniciales son Turing-computables

Segundo : Los procedimientos de recursión primitiva y composición aplicados sobre funciones Turing-computables producen funciones Turing-computables.

3.9. La Jerarquía Aritmética

En matemáticas es bastante común la introducción de mediciones. Así, en teoría de conjuntos vemos que entre los conjuntos infinitos los hay que son “*más infinitos que otros*”. Para precisar la, por así llamar, medida de infinitud, se define la noción de cardinalidad. Normalmente el concepto métrico permite establecer un orden en el ámbito de estudio, tal vez sólo un orden parcial.

Ahora nos ocuparemos de los números naturales. Este caso no es tan particular y restringido como pudiera parecer, pues vimos en temas anteriores que gracias a la aritmetización podemos hablar de números y conjuntos de números refiriéndonos a fórmulas y conjuntos de fórmulas. En principio estamos en disposición de extrapolar los resultados a cualquier entorno que pueda ser codificado mediante números. En especial, los resultados que obtengamos se aplicarán no sólo a conjuntos de números, sino también a conjuntos de fórmulas.

En el ámbito de los números naturales hemos establecido ya una partición: un conjunto puede ser recursivo o no serlo.

Ejemplos de conjuntos recursivos tenemos muchos: el conjunto de los números pares, el conjunto de los números de Gödel de los términos, de las fórmulas, etc.

Ejemplos de conjuntos no recursivos tenemos el de la teoría de los naturales $\#Th(\mathcal{N})$ —es decir, el de los números de Gödel de las sentencias verdaderas en \mathcal{N} —, el de las fórmulas válidas en el lenguaje de los naturales —es decir, el conjunto de los números de Gödel de las sentencias lógicamente válidas—, etc. Puesto que el conjunto de los conjuntos recursivos es numerable, en $\wp(\mathbb{N})$ que es supernumerable, casi todos los conjuntos no son recursivos.

Otra medición que se puede establecer en el ámbito de los conjuntos de naturales es la de los grados de irresolubilidad, que indican lo lejos que está un conjunto de ser recursivo.

También se puede usar como escala de medida la definibilidad, y aplicarla a conjuntos que sean definibles en $\mathcal{N} = \langle \mathbb{N}, 0, s, +, \cdot \rangle$ mediante fórmulas de primer orden —no nos ocuparemos de los conjuntos definibles mediante fórmulas de segundo orden—. Recordemos que un conjunto $\mathbf{R} \subseteq \mathbb{N}$ es definible en \mathcal{N} si y sólo si hay una fórmula C , con a lo sumo una variable libre tal que:

$$\mathbf{R} = \{\mathbf{n} \in \mathbb{N} \mid \mathcal{N} [\mathbf{n}/x] \Vdash C\}$$

Esto es,

$$\mathbf{R} = \{\mathbf{n} \in \mathbb{N} \mid \mathcal{N} \Vdash C(\sigma^{\mathbf{n}}c)\}$$

Por consiguiente, \mathbf{R} contiene justamente a los elementos de \mathbb{N} que hacen verdadera a C en \mathcal{N} . Para relaciones n -arias usaremos fórmulas con a lo sumo n variables libres. Naturalmente, puesto que la clase de las fórmulas del lenguaje de \mathcal{N} es numerable, el conjunto de los subconjuntos de \mathcal{N} que son definibles es sólo numerables. No obstante y pese a que, como ocurría con la recursión, la mayoría de los elementos de $\wp(\mathbb{N})$ no son definibles, los más interesantes suelen ser definibles. La razón es que si estamos interesados en una cierta relación \mathbf{R} es porque sabemos qué es \mathbf{R} y este conocimiento normalmente nos permite definir a \mathbf{R} en \mathcal{N} .

Como ejemplos de relaciones definibles en \mathcal{N} podemos citar: la relación de orden, el conjunto de los números primos, el de los números pares, la relación de divisibilidad y la de exponenciación. En efecto, el orden se define mediante:

$$C := (x = y \vee \exists z \ x + \sigma z = y)$$

mientras que los números primos se definen mediante:

$$D := (x \neq \sigma c \wedge \forall yz \ (x = y \cdot z \rightarrow y = \sigma c \vee z = \sigma c))$$

Como ejemplos de relación no definible¹² podemos citar el de los números de Gödel de las sentencias verdaderas en \mathcal{N} .

Si un conjunto es definible en \mathcal{N} , entonces nosotros queremos saber en qué grado o medida lo es.

¿Cómo se mide esto?

Para empezar, llamamos *aritméticas* a las relaciones definibles en \mathcal{N} mediante fórmulas de primer orden y *analíticas* a las definibles con fórmulas de segundo orden. En la clase de los conjuntos aritméticos podemos intentar establecer una medición —no binaria— de la definibilidad. Puesto que la definibilidad se establece mediante fórmulas, es natural fijarse en ellas para establecer dicha medición. Puede pensarse que una relación definible mediante una fórmula cortita o sin cuantificadores es más definible que otra que precise una fórmula complicadísima. La idea va un poco por ahí, pero es más sutil.

Por supuesto, las relaciones o funciones destacadas en el sistema son las más definibles, ¡tienen incluso nombre propio!; por ejemplo, la suma y el producto. Nosotros hemos escogido un sistema \mathcal{N} en el que la exponenciación no era una

¹²En el capítulo 2 aparece un ejemplo muy sencillo (el 78) de conjunto no definible.

operación destacada, pero esta elección fue bastante arbitraria; podríamos haber escogido un sistema en el que la exponenciación fuera también una operación destacada y no hubiera cambiado casi nada. Sin embargo, la fórmula que define a la exponenciación en un sistema que no la tenga como función destacada es bastante larga y por lo tanto tendríamos a la suma y al producto en distinto lugar en la ordenación por definibilidad que a la exponenciación. Como digo, esto no parece satisfactorio y nos gustaría que la suma, el producto y la exponenciación estuvieran en el mismo paquete; queremos que la medida de la definibilidad esté exenta de estas arbitrariedades. Es por ello por lo que mediremos la definibilidad módulo recursividad.

El siguiente teorema, demostrado por Gödel, puede ser aplicado.

Teorema 136 *Toda relación recursiva sobre \mathbb{N} es aritmética.*

Las relaciones aritméticas son las definibles en un sistema con universo \mathbb{N} y que tuviera destacada a todas las relaciones recursivas; esta es la forma natural de medir definibilidad módulo recursividad. Toda relación definible en este sistema lo es, en particular, mediante una fórmula en forma normal prenexa. Lo que se hace es utilizar la alternancia de cuantificadores en dicha fórmula prenexa como criterio métrico.

La jerarquía aritmética que se construye sobre esta base es así:

Δ_1 : relaciones recursivas

Σ_1 : recursivamente enumerables

Π_1 : relaciones cuyo complementario está en Σ_1

En general, por recursión se define,

Las relaciones de Σ_n son dominios de relaciones de Π_{n-1} y las relaciones de Π_n son relaciones cuyo complemento está en Σ_n .

$$\Delta_n = \Sigma_n \cap \Pi_n$$

$$\Sigma_1 = \{ \vec{a} \mid \exists b \langle \vec{a}, b \rangle \in \mathbf{R} \} \quad \mathbf{R} \text{ es recursiva}$$

$$\Pi_1 = \{ \vec{a} \mid \forall b \langle \vec{a}, b \rangle \in \mathbf{R} \} \quad \mathbf{R} \text{ es recursiva}$$

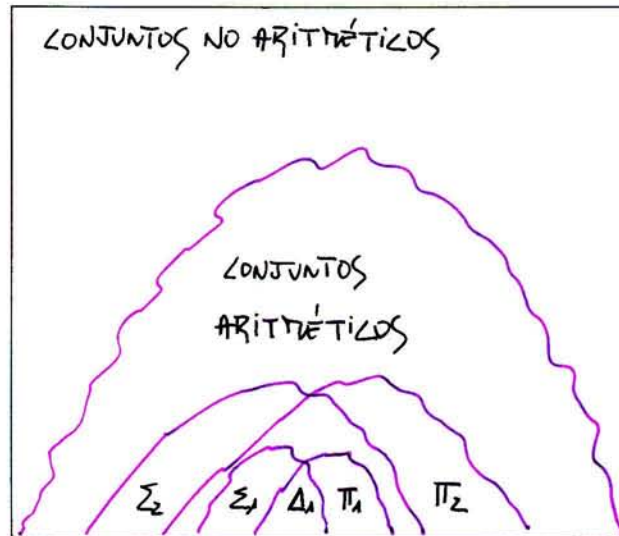
$$\Sigma_2 = \{ \vec{a} \mid \exists c \forall b \langle \vec{a}, b, c \rangle \in \mathbf{R} \} \quad \mathbf{R} \text{ es recursiva}$$

$$\Pi_2 = \{ \vec{a} \mid \forall c \exists b \langle \vec{a}, b, c \rangle \in \mathbf{R} \} \quad \mathbf{R} \text{ es recursiva}$$

Gráficamente la situación es así (ver figura: 3.1)

3.10. Apéndice: Indecidibilidad de *FOL*.

La indecidibilidad de la lógica de primer orden no se demuestra directamente, se suele utilizar algún sistema finitamente axiomatizable del que ya sabemos que es indecidible. Church en su demostración de indecidibilidad hace uso de

Figura 3.1: Elementos de $\wp(\mathbb{N})$

los resultados de otro artículo suyo, normalmente ahora solemos usar el sistema \mathbf{Q} —denominada la *aritmética de Robinson*—, una subteoría de la aritmética que tiene sólo siete axiomas; a saber:

1. $\mathbf{Q}_1 := \forall xy(\sigma x = \sigma y \rightarrow x = y)$
2. $\mathbf{Q}_2 := \forall x(c \neq \sigma x)$
3. $\mathbf{Q}_3 := \forall x(x \neq c \rightarrow \exists y(x = \sigma y))$
4. $\mathbf{Q}_4 := \forall x(x + c = x)$
5. $\mathbf{Q}_5 := \forall xy(x + \sigma y = \sigma(x + y))$
6. $\mathbf{Q}_6 := \forall x(x \cdot c = c)$
7. $\mathbf{Q}_7 := \forall xy(x \cdot \sigma y = (x \cdot y) + x)$

El aparato conceptual introducido por Gödel nos permite decir que un conjunto de fórmulas es decidable, si el conjunto de los números de Gödel de sus elementos es recursivo. Así, una teoría es decidable si y sólo si el conjunto de los números de Gödel de sus teoremas es recursivo o, de forma equivalente, es recursiva su función característica. Dada una teoría decidable hay un método efectivo para decidir si una sentencia es un teorema: se halla su número de Gödel y se usa ese número como argumento en la función característica. La sentencia será un teorema si y sólo si el valor es 1. Por el contrario, si una teoría no es decidable, y si aceptamos la tesis de Church, no habrá un método efectivo para

decidir si una sentencia es un teorema. Si lo hubiera, la función característica de los números de Gödel de sus teoremas sería efectivamente computable y, por lo tanto, recursiva, usando la tesis de Church.

La prueba de que \mathbf{Q} no es decidible¹³ usa los recursos de Gödel —es decir, gödelización y diagonalización— y ciertas propiedades de \mathbf{Q} —en particular, que todas las funciones recursivas son representables en \mathbf{Q} —.

El argumento de Church se puede presentar como sigue:

1. Sea

$$\theta := \bigwedge_{1 \leq i \leq 7} \mathbf{Q}_i$$

la conjunción de los axiomas de \mathbf{Q}

2. Por el teorema de la deducción tenemos que:

$$\mathbf{Q} \vdash C \text{ sys } \vdash \theta \rightarrow C$$

(Haciendo uso de los teoremas ya disponibles de completud y corrección, podemos parafrasear este resultado diciendo que un test de validez equivale a un test de pertenencia a la teoría \mathbf{Q} : para decidir si C es un teorema, hagamos el test a $\theta \rightarrow C$.)

3. Designemos mediante $\#A$ al número de Gödel de la fórmula A . Sea f una función que a cada número natural n le asigna $\#(\theta \rightarrow D_n)$ —donde, $\#D_n = n$ —. Esta función es claramente recursiva. Para cada fórmula C , de número de Gödel n ,

$$f(n) = \#(\theta \rightarrow C)$$

4. Sea

$$P_0 = \{\#G \mid G\}$$

es decir, el conjunto de los números de Gödel de los teoremas lógicos de la lógica de primer orden

5. Sea

$$P = \{\#C \mid \mathbf{Q} \vdash C\}$$

Por lo tanto,

$$P = \{\#C \mid \theta \rightarrow C\} = \{n \mid f(n) \in P_0\}$$

Pero P no es recursivo, puesto que es el conjunto de los números de Gödel de los teoremas de \mathbf{Q} , y sabemos que \mathbf{Q} no es decidible

6. Por consiguiente, P_0 tampoco es recursivo
7. La conclusión es que la lógica de primer orden no es decidible; no hay test de validez.

¹³Una demostración de este teorema puede leerse en Boolos [?], en la página 175, basada en resultados demostrados en el libro. Desgraciadamente, este gran lógico también murió recientemente, este mismo año de 1996.

Bibliografía

- [1] Alonso, E. [1996]. *Curso de Teoría de la Computación*. Serie Cuadernos de Trabajo. Ediciones de la UAM. Madrid.
- [2] Alonso, E. [2002]. *Teoría de la Computación*. Summa Logicae en el siglo XXI en <http://logicae.usal.es>
- [3] Barwise, J. [1977]. *Handbook of Mathematical Logic*. North Holland Publishing Company. Amsterdam. Holanda.
- [4] Beth, E. [1965]. *The foundations of mathematics*. Amsterdam: North Holland.
- [5] Boolos, G. [1989]. *Computability and Logic*. Cambridge University Press. Cambridge. U.K.
- [6] van Dalen, D. [2001]. “*Algorithms and Decision Problems: A Crash Course in Recursion Theory*” en [11]
- [7] Davis, M. (ed.) [1965]. *The Undecidable. Basic Papers on Undecidable Propositions, Unsolvability Problems, and Computable Functions*. Raven Press, Hewlett, Nueva York.
- [8] Davis, M. [1977]. “*Unsolvability Problems*”, en [3].
- [9] Enderton, H. [1972]. *A Mathematical Introduction to Logic*. Academic Press. New York. USA
- [10] Enderton, H. [1977]. “*Elements of Recursion Theory*” en [3]
- [11] Gabbay, D y Guenther, F. [2001]. *Handbook of Philosophical Logic 2nd edition*. Kluwer Academic Publishers. Dordrecht. Holanda.
- [12] Gandy, R. [1988] . “*The Confluence of Ideas in 1936*”. (en Herken [14]).
- [13] Heijenoort, J. ed. [1967]. *From Frege to Gödel*. Harvard University Press. Harvard. USA
- [14] Herken, R. ed. [1988]. *The Universal Turing Machine: A Half-Century Survey*. Oxford University Press. Oxford. U.K.

- [15] Hermes, F. [1969]. *Enumerability, Decidability, Computability*. North Holland Publishing Company. Amsterdam. Holanda.
- [16] Nagel, E. y Newman, J.R. [1958]. *El Teorema de Gödel*. Ed. castellana de Adolfo Martín. Estructura y Función. Tecnos. Madrid, 1979.
- [17] Odifreddi, P. [1989]. *Classical Recursion Theory*. North Holland Publishing Company. Amsterdam. Holanda.
- [18] Rogers, H. [1967]. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill. New York. USA
- [19] Shoenfield, J. R. [1967]. *Mathematical Logic*. Reading: Addison-Wesley Publishing Company.
- [20] Shoenfield, J. R. [1995]. "The mathematical work of S. C. Kleene". **Bulletin of Symbolic Logic**. vol. 1. Núm 1. pp 9- 43.
- [21] Smullyan, R. [1993]. *Recursion Theory for Metamathematics*. Oxford Logic Guides 22. Oxford: Oxford University Press.
- [22] Tarski, A., Mostowski, A. y Robinson, R. M. [1953]. *Undecidable Theories*. Amsterdam: North Holland Publishing Company.