Mathesis 12 (1997) 365-379
Impreso en México. Derechos reservados © 1997 por UNAM. (ISSN 0185-6200)

Los límites de las matemáticas

Atocha Aliseda

Gregory J. Chaitin. *The Limits of Mathematics: A course on Information Theory and the Limits of Formal Reasoning*. Singapore. Springer. 1997. Pp. xi + 148. ISBN: 981308359

1 Introducción

La computadora ha permeado la investigación matemática. Su uso ha ayudado tanto a convertir conjeturas (aserciones matemáticas que no han sido probadas o refutadas) en teoremas, como a verificar conjeturas para números muy grandes de casos, que son prácticamente imposible de ejecutar manualmente. Como ejemplo de lo primero, el primer teorema que se probó exitosamente usando una computadora es el *teorema de los cuatro colores*, el cual dice que cuatro es el número mínimo necesario para colorear regiones (como las de un mapa) de tal forma que no sea el caso para ningún par de regiones adyacentes compartir el mismo color. Aunque al principio hubo algunas reservas acerca de la naturaleza de esta demostración, finalmente se aceptó como prueba válida la de la computadora. Ejemplo de lo segundo es la *conjetura de Goldbach*, a saber, que todo entero positivo mayor que tres es la suma de dos primos (no necesariamente distintos). Lo que sabemos hoy en día, gracias a cálculos computacionales, es que esta conjetura es verdadera para todos los enteros positivos menores de 4×10^{11}.

Este libro es otro ejemplo del uso de la computadora para la demostración, en particular, en la matemática de lo indecidible. El autor nos propone una *Teoría Algorítmica de la Información* (TAI, de ahora en adelante), una teoría que se caracteriza por ser un enfoque práctico computacional, en la que la incompletud de un sistema formal se prueba a través de programas desarrollados en una versión extendida del lenguaje de programación Lisp.

A diferencia de lo que sugiere el título de este libro, los artículos que lo constituyen no son las partes de un texto para un curso de teoría de la información. Mas bien son piezas sueltas que presentan tanto repeticiones como lagunas para un curso completo en este tema. El texto está dividido en cuatro partes. Las tres primeras son transcripciones editadas de conferencias que el autor impartió en diversas universidades entre los años 1992 y 1996. Estas han sido publicadas posteriormente en boletines, capítulos de libros y memorias de congresos. Sus títulos son: 'Aleatoriedad en aritmética y el declive y caída del reduccionismo en las matemáticas puras', 'Programas elegantes en Lisp' y 'Una invitación a la teoría algorítmica de la información'. El cuarto artículo, 'Los límites de las matemáticas', es un texto publicado en el *Journal of Universal Computer Science* que incluye al final el código en Lisp con ejemplos de ejecuciones del programa que describe en el texto. Finalmente, hay un apéndice que contiene un intérprete de Lisp en *Mathematica*.

Sin embargo, a pesar de la estructura de este libro, el primer y tercer artículo constituyen el corazón de unas notas para un curso sobre TAI. Mientras que el primero se centra en los antecedentes históricos y en la parte teórica y preliminar de la TAI, en el tercero se describe con más detalle la propuesta de esta teoría en el contexto de programación en Lisp. Estos dos textos incluyen además, una discusión sobre las implicaciones filosóficas de la TAI. Por otra parte, los artículos segundo y cuarto contienen algunas ideas de los otros dos pero también los complementan de la siguiente forma. El segundo se centra en motivar y explicar la primera de las contribuciones originales, a saber, que el tamaño de un programa (cantidad de bits de información que contiene) es medida para su complejidad y que esto puede usarse para demostrar la incompletud de un sistema formal en el contexto del lenguaje de programación Lisp. El artículo cuarto es por un lado, una versión compactada del tercero y por el otro lado, describe al igual que el segundo, el siguiente contenido para un curso sobre TAI:

- Introducción histórica al problema de incompletud.
- Introducción al lenguaje (extendido) de programación en Lisp.
- Maquina universal de Turing y definición de Ω, la *probabilidad* de parada.
- Prueba de que Ω es algorítmicamente irreducible en términos de Lisp.
- Implicaciones filosóficas.

A continuación describiremos el contenido del libro, aunque no seguiremos fielmente su estructura. Empezaremos por los antecedentes históricos de las pruebas de incompletud para luego describir la TAI desde los puntos de vista teórico y práctico. Terminaremos con la discusión que Chaitin nos ofrece sobre las implicaciones filosóficas. No entraremos en muchos detalles matemáticos ni en cuestiones técnicas de programación. De hecho, en muchos casos, Chaitin sólo nos ofrece la línea general de su argumentación matemática y de sus técnicas de programación. Finalmente, ofreceremos nuestra opinión y crítica sobre este libro.

II Antecedentes

La tradición axiomática en matemáticas se remonta al método deductivo que heredamos desde Euclides, en el cual a partir de ciertos axiomas básicos y reglas de inferencia, se deducen todas las verdades de la geometría elemental. Esto llega a su culminación en los años veinte con el proyecto de David Hilbert, a saber, el de construir un sistema formal axiomático completo y consistente para las matemáticas. Esto es, un sistema tal que para toda aserción matemática, se pueda derivar ella o su negación (completo) y del cual no se pueda derivar tanto una aserción como su negación (consistente). Una implicación inmediata del éxito de este proyecto es que no hay límites para el razonamiento matemático a través del método axiomático.

Sin embargo, en 1931 un matemático de origen austriaco, Kurt Gödel, probó que si se asume consistencia en un sistema axiomático como el de la aritmética, entonces necesariamente ese sistema es incompleto. Esto es, que si tenemos un sistema del cual no se derivan contradicciones, entonces hay por fuerza aserciones que no se pueden decidir matemáticamente, donde el aparato lógico axiomático no es suficiente para probarlas o refutarlas. La demostración de Gödel es un tanto elaborada y compleja pues introduce una representación aritmética para las fórmulas y con ello construye una aserción autoreferencial, una que dice de sí misma que no es demostrable. Este resultado marca un límite al tratamiento axiomático de las teorías matemáticas, a la tradición formal axiomática matemática iniciada por Euclides y seguida por matemáticos de la talla de Leibniz, Peano, Boole, Russell y Whitehead. Con este resultado de incompletud Gödel destruye el sueño matemático que culminó con Hilbert.

Posteriormente a este resultado, Alan Turing, matemático de origen inglés, probó en 1936 algo de lo que el resultado de Gödel se obtiene como corolario. Esto es, que no hay un procedimiento de decisión que decida si un programa se para. En el trabajo que Turing reporta este resultado, su objetivo es analizar a los números computables (aquellos para los que existe un algoritmo que calcule sus dígitos). Su demostración consiste en una construcción de una sucesión de programas que calculan los enésimos dígitos de números reales computables. Usa el argumento diagonal de Cantor para construir un número real que no es computable, probando con esto que no existe un procedimiento mecánico que pueda decidir si el enésimo programa de computadora calcula o no el enésimo dígito. Por lo tanto, Turing muestra que el 'problema de la parada' ('halting problem') no tiene solución.

III Probabilidad de Parada y Aleatoriedad en la Aritmética

El trabajo de Chaitin es una continuación de los resultados de Kurt Godel y de Alan Turing en materia de incompletud. El autor ve su proyecto como la *continuación y extensión del trabajo de Turing* [p.1]. Su trabajo está íntimamente relacionado con números *dramáticamente no computables*. En el plano teórico de su investigación, una de sus contribuciones es la de construir un número real no computable que es aún más no computable que el construido por Turing. Para tal efecto, Chaitin introduce la noción de *probabilidad de parada* ('halting probability') y su resultado fundamental es que dicha probabilidad es *algorítmicamente irreducible*. Veamos en detalle el significado de la probabilidad de parada y de la irreducibilidad en términos algorítmicos. La probabilidad de parada la expresa como el siguiente número Ω acotado entre los números reales 0 y 1

$$0 < \Omega = \sum_{p\ \text{para}} 2^{-|p|} < 1$$

Supongamos que tenemos todos los programas de computadora, representados por números, en una bolsa. El número Ω es la probabilidad de que al obtener un número al azar de esta bolsa, el programa se pare. Chaitin muestra que este número no sólo es no computable, sino que *es no computable de la peor manera posible* [p.13]

En el texto, sin embargo, el autor sólo nos da las pistas para probar esta aserción. En primer lugar, nos dice, este número es algorítmicamente irreducible (o incompresible). Esto quiere decir que si se quieren calcular los primeros N bits de Ω, el programa que lo calcula debe tener al menos N bits de largo.

En segundo lugar, el autor nos da las pautas para mostrar su resultado aritméticamente.[1] Esto es, muestra cómo Ω se puede vestir aritméticamente y con esto obtener total aleatoriedad en teoría de los números. Así, convierte Ω a una ecuación diofántina exponencial con 17,000 variables (la ecuación, dice, mide 200 páginas!). Esta es una ecuación en donde cada constante es un número entero y una de las variables es un parámetro que va cambiando. La pregunta entonces es: ¿esta ecuación tiene un número finito o infinito de soluciones? (nótese que esta pregunta es más fina que preguntar si hay o no solución). La ecuación está construida de tal forma que tiene un número finito de soluciones si un bit particular de Ω es un cero y tiene un número infinito de soluciones si ese bit es uno. Así, decidir si esta ecuación diofántina exponencial tiene un número finito o infinito de soluciones es exactamente igual a determinar cuál es un bit individual de la probabilidad de parada. Y esto es indecidible, como hemos visto, es información matemática irreducible. Chaitin [p. 21-22] concluye su primer artículo con lo siguiente:

> Por Gödel supimos que no podemos obtener un sistema formal axiomático que sea completo. Sabíamos que estábamos en problemas y Turing nos mostró que simple era esto, pero Ω es un caso extremo donde el razonamiento falla en su totalidad.

Hasta aquí las investigaciones de Chaitin tienen un estilo similar a las de Turing, aunque es claro que sus resultados toman formas más refinadas. Desde un punto de vista aritmético, el problema equivalente al problema de Turing nos muestra que hay N preguntas aritméticas con respuestas positivas y negativas que son solo Log N bits de información algorítmica. La ecuación de Chaitin obtiene N preguntas aritméticas con respuestas positivas y negativas que son irreducibles, información matemática *incompresible*. Lo que sigue de su propuesta, propiamente la TAI, puede considerarse como una *teoría de funciones aplicada*, esto es lo que describimos a continuación.

---

[1] La relación entre la aritmética y la computabilidad se da al menos desde Gödel, mucho trabajo posterior muestra que problemas que involucran cálculos computacionales son equivalentes a problemas aritméticos con números enteros. Por ejemplo, el problema décimo de Hilbert, el de encontrar si existe un procedimiento mecánico para resolver si una ecuación diofántina (aquella que sólo tiene números enteros), resulta ser equivalente al problema de la parada de Turing.

IV Teoría Algorítmica de la Información

El objetivo principal de la TAI, en palabras de Chaitin es el siguiente:

> Todo el punto de mi teoría algorítmica de la información, todo el punto de mi enfoque teórico-informacional a la incompletud, es que algunas veces para obtener más información a partir de un conjunto de axiomas, sólo hay que meter otros más [p. 83-84].

Pero para poder llegar a esta conclusión, hay detrás todo un aparato teórico y práctico. En primer lugar, Chaitin define la máquina universal de Turing en términos de Lisp. Después describe las funciones para calcular la probabilidad de parada (Ω), da las pautas para la prueba de que es algorítmicamente irreducible y termina con un teorema de incompletud para un sistema formal axiomático representado como una expresión en Lisp. Veamos primero el resultado de que Ω es algorítmicamente irreducible.

Si pensamos en Ω como un número real escrito en binario, representado como una cadena de bits, tendrá una forma como la siguiente:

$$\Omega = 0.011101$$

Esto es, un cero seguido de un punto binario y luego una lista de ceros y unos. Sea Ω_N la cadena que representa los primeros N bits de Ω. El teorema que señala la irreducibilidad de Ω está representado en la siguiente desigualdad:

$$H(\Omega_N) > N - 8000$$

Esto dice que el tamaño en bits del programa más pequeño para la máquina de Turing que calcula los primeros N bits de Ω es mayor que los N bits menos 8000. Dicho de otra forma, para obtener los primeros N bits de la probabilidad de parada, se necesita un programa con un tamaño al menos un bit mayor que N-8000. La razón es la siguiente. Sabiendo los primeros N bits de Ω, uno podría resolver el problema de la parada para todos los programas de tamaño menor o igual a N bits. Así, podríamos ejecutar cada uno de los programas de tamaño menor o igual a N bits que paran, ver lo que calculan e ir colocando su resultado en una lista. Esto daría un resultado total que no podría calcularse con un programa de tamaño menor o igual a N bits. En el texto se describe muy sumariamente el algoritmo para obtener este resultado, pero refiere a su página electrónica para el código Lisp e invita al lector a probar el programa con sus ejemplos.

Finalmente, el teorema de incompletud de un sistema formal axiomático (SFA) es el siguiente:

> **Teorema de Incompletud.** Si un SFA tiene como complejidad su tamaño N, entonces puede usarse para determinar el valor de a lo más N + 15328 bits de Ω [p. 80].

Este resultado dice que se pueden obtener algunos bits de Ω sin contradecir los resultados de irreducibilidad e incompletud pero que esto tiene un límite a partir del cual ya no se pueden determinar los siguientes bits y Ω se vuelve totalmente impredecible, determinar el siguiente bit es equivalente a echar un volado. Este límite esta directamente relacionado con la complejidad del SFA. Entre más axiomas tengamos, más bits de Ω podemos determinar.

Para probar este resultado, Chaitin hace una demostración análoga a la paradoja de Berry.[2] La idea central de la prueba es que si se pudieran determinar más bits de Ω, entonces Ω no sería algorítmicamente irreducible. Esto es:

> No puedes probar que una cadena de N bits es algorítmicamente irreducible si ésta tiene más bits que los axiomas que estás usando para hacer la demostración. Similarmente, si tienes N bits de axiomas obtienes a lo más N + 15328 bits de Ω [p. 77].

Para Chaitin no basta hablar de un algoritmo que haga uso de la máquina universal (idealizada) de Turing, sino que hay que poder trabajar con un programa real en el que se puedan ejecutar las pruebas y hacer ejemplos interesantes. El sistema formal axiomático se representa como una S-expresión y funciones en Lisp determinan su tamaño y así miden su complejidad. Con esto se hace un programa que determina la capacidad exacta del sistema para calcular los bits de Ω.

2. Esto es, que es imposible probar que cualquier programa es 'elegante' (en términos de Lisp una S-expresión es elegante si ninguna S-expresión menor tiene el mismo valor). El resultado general que Chaitin prueba es que dado un sistema formal cuya complejidad es N no puede demostrar si una S-expresión es elegante si el tamaño de la S-expresión es mayor que N más una cierta constante c.

V Reflexiones Filosóficas

Como Ω es algorítmicamente irreducible, la única forma de obtenerlo a partir del sistema axiomático es agregándolo como axioma o postulado, insertarlo tal cual. Es en este sentido que Ω es información matemáticamente irreducible, no hay estructura o patrón que se pueda utilizar para calcular el siguiente bit ya que el tamaño del sistema axiomático es insuficiente. No se puede calcular, ni siquiera probabilísticamente, cuál es la probabilidad del siguiente dígito. Además, la reducción aritmética de Ω indica que la aleatoriedad permea campos de las matemáticas como la teoría de números. Asimismo,

> lo que Ω les muestra, lo que he descubierto, es que algunos hechos matemáticos son verdaderos sin razón alguna, son verdaderos por accidente! Y consecuentemente escapan por siempre del poder del razonamiento matemático [p. 24].

Estos resultados llevan a Chaitin [p. 24] a sugerir que la matemática debiera comportarse como una ciencia experimental y empírica como lo es la física:

> Yo creo que la teoría de números elemental y el resto de la matemática deberían investigarse más en el espíritu de la ciencia experimental y que uno debiera estar dispuesto a aceptar nuevos principios. Creo que la aserción de Euclides de que un axioma es una verdad autoevidente es un grave error. La ecuación de Schrödinger ciertamente no es una verdad autoevidente! y la hipótesis de Riemann tampoco es autoevidente, aunque sí muy útil.

De la física Newtoniana no se pueden derivar muchas cosas, como las ecuaciones de Maxwell, por ejemplo. Para obtenerlas, se necesitan nuevos postulados. Lo mismo pasa con la hipótesis de Riemann; hay muchos resultados que involucran la distribución de los números primos que no se podrían probar sin asumirla. Todo esto sugiere que hay que agregar nuevos axiomas no porque sean verdades autoevidentes sino porque sean útiles.

Esta visión de la matemática como ciencia empírica se remonta al menos a las reflexiones de Lakatos sobre las matemáticas y la comparte ahora Chaitin con dos nuevas corrientes: con la llamada 'escuela cuasi-empiricista', un nuevo enfoque en los fundamentos de las matemáticas y con la 'matemática experimental', disciplina cuya práctica se basa en el uso de la computadora, como en los campos de la teoría del caos, fractales y dinámica no-lineal.

Sin embargo, en general la práctica matemática no procede empíricamente. Si no se puede probar una conjetura, no se adjudica este hecho a la (posible) incompletud del sistema axiomático y mucho menos se añade como un nuevo axioma (el riesgo de hacerlo, claro está, es que una conjetura que solo se asume, bien puede resultar falsa en un futuro y por tanto tendría que ser retractada del sistema). Así, a pesar de los resultados de incompletud, el quehacer matemático procede todavía muy en el estilo de Hilbert, en el cual los teoremas son verdaderos por alguna razón y ésta debe ser una prueba. La labor del matemático es justamente la de encontrar esas pruebas que justifiquen sus verdades.

En opinión de Chaitin, lo que está transformado (al menos en parte) la práctica matemática no ha sido ni el resultado de incompletud de Gödel ni de indecidibilidad de Turing ni sus propios resultados, sino el uso intenso de la computadora. Aunque los resultados de Gödel son devastadores para el proyecto matemático hilbertiano, la aserción gödeliana autoreferente no es paradigmática de lo que el matemático encuentra en su práctica diaria y tal vez por esto el resultado de incompletud no se ha tomado muy en serio. Pero hacer cálculos computacionales y probar experimentalmente muchos casos permite al matemático hacer conjeturas a partir de sus resultados empíricos y así proceder de una manera más pragmática. El ideal sigue estando en probar matemáticamente estas conjeturas (sobre todo si las pruebas son cortas y elegantes), pero cuando esto no es posible, en ciertos casos se usan hipótesis que se respaldan en resultados de experimentos computacionales.

VI Crítica y Conclusiones

La contribución de la teoría algorítmica de la información que hace Chaitin en este libro es doble: por un lado, el resultado de la irreducibilidad algorítmica de Ω es un resultado teórico más concreto y fino que el resultado de incompletud de la aritmética de Gödel y el de la indecidibilidad del problema de la parada de Turing.

En primer lugar, el resultado de incompletud de Chaitin no hace uso de la autoreferencia. Mientras que Gödel utiliza un enunciado autoreferente ("este enunciado es falso"), Chaitin adapta la paradoja de Berry ("el primer entero positivo que hace que este enunciado sea muy pequeño para poder nombrarlo") para el mismo efecto (la única señal de autoreferencia de la que hace uso es que un enunciado pueda expresar de él mismo cuánto mide). En segundo lugar, Turing muestra la indecidibilidad de que un programa pare y Chaitin muestra que si se quiere tener un programa que imprima los primeros N bits de Ω y que luego pare, entonces este programa debe tener al menos N bits de largo. Esto es, la *probabilidad de parada* es *algorítmicamente irreducible* en sistemas menores a N bits.

En este sentido, la noción de irreducibilidad algorítmica es un refinamiento de la noción de decidibilidad lógica ya que nos responde no sólo a la pregunta de si una aserción (o su negación) se puede probar, más aún, en caso contrario, nos dice que tanto más complejo debe ser nuestro sistema axiomático para que la aserción pueda decidirse.

Por otro lado, el enfoque de Chaitin rompe con el estilo puramente teórico y abstracto de sus antecesores. Su propuesta es una teoría de funciones recursivas aplicada, práctica y empírica en la cual el resultado de incompletud se prueba con un programa de computadora real y tangible, usando nociones muy concretas como la del tamaño de un programa para la medida de su complejidad. Este enfoque es un punto a favor del trabajo de llevar a cabo la programación computacional de un resultado teórico. El mismo autor señala que tenía sus resultados teóricos de incompletud desde los años setenta, pero que no fue sino hasta muy recientes fechas, esto es, veinticinco años después, que pudo programar el algoritmo de su prueba original [p. 34].

En un nivel más general, en el plano de una filosofía de la ciencia matemática, este libro pone un énfasis muy especial en el papel que juega la computadora en la naturaleza de la investigación matemática. Por un lado, Chaitin sugiere con sus resultados de incompletud que la práctica matemática debiera ser más en el espíritu de una ciencia empírica, en donde las conjeturas y aserciones indecidibles se puedan añadir al cálculo deductivo como axiomas o postulados. Por otro lado, aunque esto no se deriva directamente de sus resultados, propone que los resultados de experimentos computacionales tengan el estatus de hipótesis a considerar como nuevos axiomas en un sistema axiomático.

Sobre estas aserciones empiristas quiero hacer algunas observaciones. En primer lugar, en la discusión informal de Chaitin, se confunden dos aspectos, a mi juicio muy distintos, del uso de la computadora en la práctica matemática. Mientras que las matemáticas experimentales hacen uso de la computadora para hacer conjeturas, como guía en el descubrimiento de hipótesis; la teoría aplicada de funciones recursivas de Chaitin usa a la computadora para demostrar, para probar la incompletud de un sistema formal axiomático. En el primer caso, la computadora se propone como parte de la estrategia heurística de construcción de hipótesis y en la segunda como medio para construir la prueba.

En segundo lugar, la sugerencia de agregar las aserciones indecidibles como axiomas es muy interesante, pero en mi opinión, es muy apresurado generalizarla hasta sugerir que las conjeturas e hipótesis debieran añadirse igualmente como axiomas. Yo más bien compararía la aserción más débil, a saber, que las conjeturas se pudieran agregar como aserciones hipotéticas y con ellas pudiera probarse la verdad relativa de otras aserciones que dependen de ellas. Esto acercaría la práctica matemática a los hábitos de una ciencia experimental, pero las pruebas no perderían su certidumbre, simplemente serían relativas a la verdad de sus aserciones hipotéticas.

Por último, a diferencia del autor de este libro, a mí no me parece sorpresivo que los resultados de incompletud no hayan influenciado la práctica matemática. Mientras los resultados matemáticos puedan seguirse mostrando con pruebas matemáticas, no hay razón para dejarlo de hacer. Sin embargo, en lo que sí me parece que los resultados de incompletud deberían influir en un futuro, especialmente los de Chaitin es en hacer parte de la práctica matemática a las pruebas de indecidibilidad. Sin embargo, esto es todavía pedir mucho, pues a pesar de sus conexiones, los resultados de indecidibilidad conocidos hasta ahora, se han probado de manera diferente y son muy complicados.

Así, los resultados de incompletud, más que imponer un límite al razonamiento humano nos sugieren que *subsiste la posibilidad de descubrir nuevos principios de demostración* [Nagel E. y Newman 1970 124] y en esto el uso de la computadora está jugando un papel fundamental como lo muestra Chaitin en este texto.

El estilo informal de las pláticas editadas en este libro, hace de su lectura una experiencia agradable, pero no por ello fácil y accesible. De hecho, el tipo de auditorio que se presupone es uno que tenga conocimientos tanto de lógica como de programación. Está dirigido a lectores familiarizados con nociones metalógicas y computacionales como lo son completud, consistencia, decidibilidad y con conocimiento de Lisp. Este libro es altamente recomendable tanto para lógicos teóricos que quieran ver resultados de incompletud claros y concretos, como para estudiosos de la computación que quieran conocer los resultados lógicos de incompletud. Sin embargo, este libro no es más que una introducción al tema, para un curso completo, el lector debe remitirse a las referencias adicionales que menciona el autor.

Atocha Aliseda es doctora en Filosofía y Sistemas Simbólicos por la Universidad de Stanford, California en 1997. Su tesis *Seeking Explanations: Abduction in Logic, Philosophy of Science and Artificial Intelligence* está publicada por el Institute for Logic, Language and Computation (ILLC) de la Universidad de Amsterdam. Actualmente es investigadora en el Instituto de Investigaciones Filosóficas de la UNAM. Sus temas de investigación son las lógicas no-monotónicas, la representación y el descubrimiento del conocimiento en inteligencia artificial, y la conexión entre la filosofía de la ciencia y la inteligencia artificial.

Referencias

NAGEL, E. y NEWMAN, J. R. 1970. *El teorema de Gödel*. Madrid: Tecnos. Pp. 140. (Col. Estructura y función. El porvenir actual de la ciencia #11).