Ordenadores, paradojas y fundamentos de las matemáticas

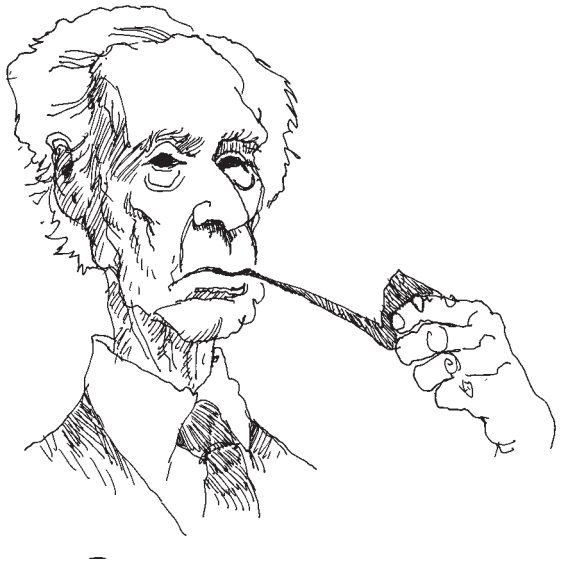
Grandes pensadores del siglo XX han demostrado que la incompletitud y la aleatoriedad medran incluso en el mundo austero de la matemática

Gregory J. Chaitin

Todos saben que los ordenadores son aparatos muy prácticos. Tanto, que se han vuelto indispensables en el funcionamiento de una sociedad moderna. Pero hasta los informáticos han olvidado —exagero, pero sólo un poco— que fueron inventados para que ayudasen a aclarar una cuestión filosófica concerniente a los fundamentos de la matemática. ¿Sorprendente? Sí, en verdad.

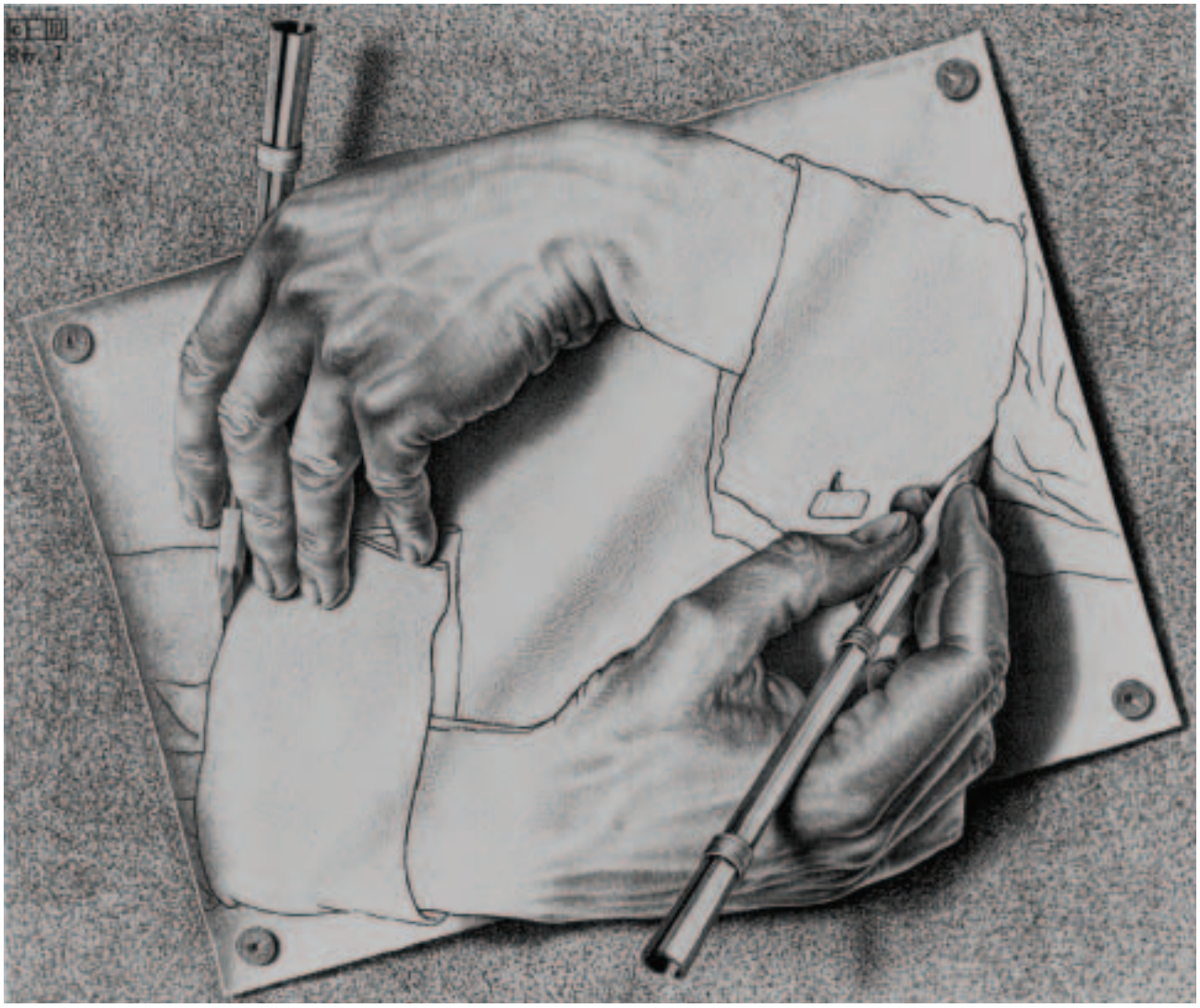
Comienza esta asombrosa historia con David Hilbert, un célebre matemático alemán, que a principios del siglo XX propuso la formalización completa de todo el razonamiento matemático. Pero resultó que era imposible formalizar el razonamiento matemático, por lo que, en cierto sentido, su idea fue un tremendo fracaso. Mas, en otro sentido, tuvo un gran éxito, porque el formalismo ha sido uno de los grandes dones que nos ha hecho el siglo XX. No para el razonamiento o la deducción matemática, sino para la programación, para el cálculo, para la computación. Una pieza olvidada de la historia intelectual.

Me propongo referir aquí esa historia sin detenerme en los detalles de índole matemática. Será, pues, imposible explicar plenamente la obra de quienes hicieron las aportaciones fundamentales, entre ellos Bertrand Russell, Kurt Gödel y Alan Turing. Aun así, el lector paciente debería poder captar la esencia de sus argumentos y comprender en qué se inspiraron algunas de mis propias ideas sobre la aleatoriedad inherente a la matemática.



Las paradojas lógicas de Russell

Voy a empezar con Bertrand Russell, matemático que al pasar el tiempo se tornaría filósofo, primero, y por último, humanista. Russell constituye una figura clave porque descubrió algunas paradojas muy perturbadoras en la lógica misma. Es decir, halló casos en los que razonamientos en apariencia impecables conducen a contradicciones. Las aportaciones de Russell fueron fundamentales para que se difundiese la idea de que estas contradicciones causaban



1. *MANOS QUE DIBUJAN*, obra creada por M. C. Escher en 1948, proporciona una analogía visual de la "paradoja de Russell", así llamada en recuerdo del matemático británico Bertrand Russell. Planteó a sus coetáneos de principios del siglo XX este problema lógico, que más tarde inspiraría los trabajos de Kurt Gödel, de Alan Turing y del autor sobre los límites de las matemáticas. Una de las formas que toma la paradoja de Russell es el par de enunciados: "La oración siguiente es falsa. La oración anterior es verdadera." Cada aserto, por separado, parece razonable (es decir, puede ser verdadero o falso); en cambio no es posible evaluar su verdad o falsedad al tomarlos conjuntamente. Es su combinación la que origina la paradoja, lo mismo que las dos manos del dibujo de Escher.

una crisis grave y habían de ser resueltas de algún modo.

Las paradojas que Russell descubrió atrajeron mucho la atención en los círculos matemáticos, pero, curiosamente, tan sólo una de ellas acabó llevando su nombre. Consideremos el conjunto de todos los conjuntos que no son un elemento de sí mismos. Preguntemos entonces: "¿Es este conjunto elemento de sí mismo?". Si fuera elemento de sí mismo, no lo sería, y recíprocamente.

El conjunto de todos los conjuntos mencionados en la paradoja de Russell encuentra un símil en el barbero de un pueblo pequeño y apartado: el barbero rasura a todos los hombres que no se afeitan ellos mismos. Tal descripción parece francamente razonable hasta que se pregunta: "¿Se afeita el barbero a sí mismo?". Se afeita a sí mismo si, y solamente si, no se afeita a sí mismo. Desde luego, se podría decir: "¿Y a quién le importa ese hipotético barbero? ¡Todo eso no es más que un absurdo juego de palabras!". Pero cuando lo que se está dilucidando es el concepto matemático de conjunto, no resulta tan fácil dejar de lado un problema lógico.

La paradoja de Russell es un eco, en la teoría de conjuntos, de otra paradoja muy anterior, ya conocida por los antiguos griegos. A menudo se la llama paradoja de Epiménides, o paradoja del mentiroso. Se dice que Epiménides exclamó: "¡Esta aseveración es falsa!". ¿Lo es? Si su aseveración es falsa, ha de ser verdadera. Pero, si es verdadera, es falsa. Así que, cual-

quiera que sea la hipótesis sobre su veracidad, estamos en conflicto. Otra versión de la paradoja, en dos enunciados, reza: "El enunciado siguiente es verdadero. El enunciado precedente es falso". Cada enunciado, individualmente, parece estar claro, pero, combinados, crean un sinsentido. Es posible desdeñar tales paradojas, considerándolas juegos de palabras sin significado, pero algunas de las más grandes inteligencias del siglo XX se las tomaron muy en serio.

Una de las reacciones a la crisis de la lógica fue la tentativa de Hilbert, que trató de eludirla por medio del formalismo. Si encontramos conflictos al seguir razonamientos que parecen correctos, la solución consiste en utilizar la lógica simbólica para crear un lenguaje artificial y ser muy cuidadosos al especificar sus reglas, de modo que no surjan contradicciones. Después de todo, el lenguaje cotidiano es ambiguo: no siempre se sabe de cierto cuál es el antecedente de un pronombre.

El plan de rescate de Hilbert

La idea de Hilbert consistía en crear para el razonamiento, para la deducción y para la matemática un lenguaje artificial perfecto. Hizo, por tanto, hincapié en la importancia del método axiomático, donde se parte de un conjunto de postulados básicos (axiomas) y reglas bien definidas para efectuar deducciones y derivar teoremas válidos. La idea de trabajar matemáticamente de este modo se remonta a los antiguos griegos, y en particular, a Euclides y su geometría, un sistema de hermosa claridad matemática.

Dicho de otro modo, era intención de Hilbert ser absolutamente riguroso en lo que se refería a las reglas del juego —las definiciones, los conceptos elementales, la gramática y el lenguaje—, de modo que hubiera un general acuerdo sobre la forma en que se había de hacer la matemática. En la práctica resultaría excesivamente laborioso utilizar un sistema axiomático tal para desarrollar nuevos resultados o teorías matemáticas, pero su importancia desde el punto de vista filosófico sería grande.

La propuesta de Hilbert no parecía demasiado espinosa. Después de todo, no hacía sino seguir las tradiciones de formalización de la matemática; bebía de una larga historia de trabajos de Leibniz, Boole, Frege, y Peano. Pero lo que él deseaba era recorrer el camino completo, hasta el mismísimo fin, y formalizar la *totalidad* de la matemática. La gran sorpresa fue que tal cosa no resultase posible. Hilbert estaba equivocado, aunque su error fue tremendamente fructífero porque había planteado una pregunta muy acertada. Al formularla creó una disciplina del todo nueva, la *metamatemática*, un campo introspectivo de la matemática en el que se estudia lo que la matemática puede, o no puede, conseguir.

La noción fundamental es la siguiente: en cuanto se entierra la matemática en un lenguaje artificial *à la* Hilbert, en cuanto se establece un sistema axiomático completamente formal, podemos olvidarnos de que posee algún significado y limitarnos a considerarla un juego; sus piezas serían marcas trazadas en un papel, y consistiría en deducir teoremas de los axiomas. Claro está, si se hace matemática es porque tiene significado. Pero si se desea estudiar la matemática utilizando métodos matemáticos, es necesario destilar el significado y limitarnos a examinar un lenguaje artificial con reglas absolutamente precisas.

¿Qué clase de cuestiones podríamos plantear? Por ejemplo, si se puede demostrar que 0 = 1. (Podemos esperar que no.) A decir verdad, dada una proposición cualquiera, llamémosla A, podemos preguntarnos si es posible demostrar, o bien A, o bien la contraria de A. Se considera que un sistema axiomático formal es completo si se puede demostrar, bien que A es verdadera, bien que A es falsa.

Hilbert perseguía la creación de reglas tan precisas, que toda demostración pudiera siempre someterse a un arbitraje imparcial, a un procedimiento mecánico capaz de afirmar "esta demostración se atiene a las re-

El autor

GREGORY J. CHAITIN es matemático del Centro de Investigación Thomas J. Watson de IBM, en Yorktown Heights, Nueva York. Es también profesor visitante en las universidades de Buenos Aires y de Auckland. Ha sido, a lo largo de los últimos 35 años, el principal arquitecto de la teoría algorítimica de la información, que inventó cuando todavía no contaba 20 años. Su logro más reciente ha consistido en transformar la teoría algorítmica de la información de modo que sea aplicable a los programas informáticos reales. El presente artículo es resumen de una conferencia pronunciada en 1999 en la Universidad de Massachusetts en Lowell, cuya versión completa está publicada en el libro *Conversations with a Mathematician*. Se reproduce con permiso de Springer-Verlag.

glas", o tal vez "hay un error tipográfico en la línea 4", o "eso que en la línea 4 se supone es consecuencia de la línea 3, en realidad no lo es". Ese veredicto sería el final. Sin apelación.

No pensaba Hilbert que la creación matemática hubiera de llevarse a cabo de ese modo, sino que, si se pudiera hacer matemática de ese modo, se podría utilizarla para estudiar su propio poder. Y Hilbert pensó que él mismo iba a ser capaz de ejecutar tal empresa. Podemos, pues, imaginar la enormidad del desconcierto cuando en 1931 un matemático austríaco, Kurt Gödel, demostró que el plan de rescate de Hilbert no era en modo alguno razonable. Jamás podría ser llevado a efecto, ni siquiera en principio.

La incompletitud de Gödel

Gödel dinamitó la visión de Hilbert en 1931. Por entonces era docente en la Universidad de Viena, si bien procedía de la hoy llamada República Checa, de la ciudad de Brno en concreto, que en aquella época formaba parte del Imperio Austrohúngaro. Posteriormente, pasaría, como Einstein, al Instituto de Estudios Avanzados de Princeton.

El descubrimiento de Gödel fue pasmoso: Hilbert estaba totalmente equivocado; no hay modo de que exista un sistema axiomático para la totalidad de la matemática en el que quede claro como el agua si un enunciado es verdadero o no. Con mayor precisión: Gödel descubrió que el plan falla aun limitándose a la aritmética elemental, es decir, a los números 0, 1, 2, 3... la adición y la multiplicación.

Cualquier sistema formal que trate de contener toda la verdad y nada más que la verdad al respecto de la adición, la multiplicación y los números 0, 1, 2, 3,... tendrá que ser incompleto. O más bien: será, ora incoherente, ora incompleto. Por tanto, si se supone que solamente dice la verdad, entonces no dirá toda la verdad. En particular, si se supone que los axiomas y las reglas de deducción no permiten la demostración de teoremas falsos, habrá teoremas verdaderos que no podrán ser demostrados.

La demostración de la incompletitud dada por Gödel es muy ingeniosa. Muy paradójica. Una locura casi. Gödel empieza, en efecto, con la paradoja del mentiroso, a saber, la afirmación "¡soy falsa!", que no es ni verdadera ni falsa. En realidad, lo que Gödel hace es construir una aseveración que dice de sí misma: "¡Soy indemostrable!". Desde luego, hará falta muchísimo ingenio para poder construir en la teoría elemental de números —en la aritmética— un enunciado matemático que se describa a sí mismo y diga semejante cosa, pero *si fuéramos capaces* de lograrlo, enseguida comprenderíamos que estaríamos en un brete. ¿Por qué? Porque si el enunciado es demostrable, entonces es necesariamente falso; estaríamos demostrando resultados falsos. Si es indemostrable, como dice de sí mismo, entonces es verdadero, y la matemática, incompleta.

Hay en la demostración de Gödel muchos detalles técnicos complicados. Pero al consultar su artículo original, encontramos en él algo que se parece mucho a la programación en LISP. Es debido a que la demostración de Gödel comporta la definición recursiva de una gran cantidad de funciones que operan sobre listas, y eso es precisamente lo que hace LISP. Así pues, aunque en 1931 no existían los ordenadores ni los lenguajes de programación, una mirada retrospectiva deja ver claramente un lenguaje de programación en el núcleo del artículo original de Gödel.

John von Neumann, otro famoso matemático de aquellos tiempos (que, dicho sea de paso, tuvo un importante papel en la promoción y la creación de la tecnología informática en los Estados Unidos), apreció inmediatamente el hallazgo de Gödel. Von Neumann jamás se había planteado que el proyecto de Hilbert pudiera ser erróneo. Así pues, Gödel no sólo había demostrado una inteligencia apabullante, sino que tuvo la valentía de presumir que Hilbert podría estar equivocado.

Muchos consideraron que el artículo de Gödel era absolutamente devastador. Toda la filosofía matemática tradicional acababa de quedar reducida a escombros. En 1931, sin embargo, había en Europa algunos otros problemas de los que preocuparse: una gran depresión económica y una guerra en ciernes.

La máquina de Turing

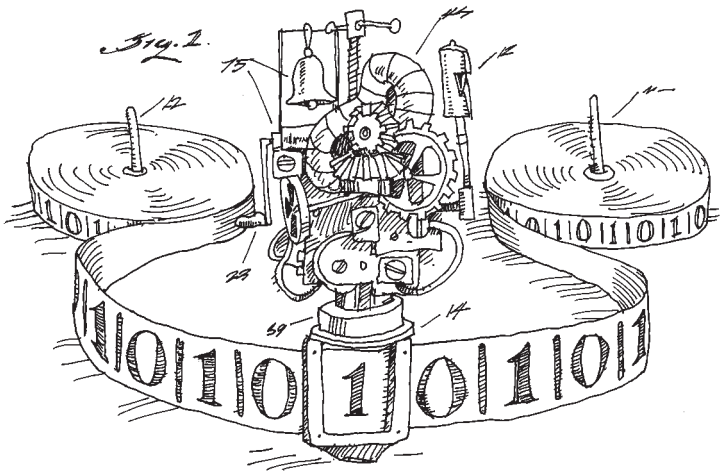
El siguiente avance de importancia tuvo lugar cinco años después, en Inglaterra, cuando Alan Turing descubrió la no-computabilidad. Recordemos que, según Hilbert, debía existir "un procedimiento mecánico" que decidiese si una demostración se atenía a las reglas o no. Hilbert no aclaró nunca qué entendía por procedimiento mecánico. Turing, en esencia, vino a decir que se trataba de una máquina (una máquina de un tipo que ahora llamamos máquina de Turing).

El artículo original de Turing contiene un lenguaje de programación, lo mismo que el artículo de Gödel,

o mejor dicho, lo que hoy denominaríamos un lenguaje de programación. Pero esos dos lenguajes de programación son muy diferentes. El de Turing no es un lenguaje de alto nivel, como el LISP; se trata más bien de lenguaje de máquina, el código "en crudo" formado por unos y ceros que se le suministra al procesador central de un ordenador. El invento de Turing de 1936 es, de hecho, un lenguaje de máquina horrible, que nadie querría utilizar hoy, porque es demasiado rudimentario.

Pero aunque las máquinas computadoras hipotéticas de Turing sean muy sencillas, y su lenguaje de máquina bastante primitivo, no carecen precisamente de versatilidad. En su artículo de 1936, Turing afirma que una máquina tal debería ser capaz de efectuar cualquier cómputo que un ser humano pudiese llevar a cabo.

En este punto, el curso del razonamiento de Turing experimenta un violento giro. ¿Qué le sería *imposible* a semejante máquina? ¿Qué es lo que no podría hacer? Y Turing encuentra inmediatamente un problema que ninguna máquina de las que llevan su nombre podría resolver: el problema de la detención, es decir, decidir de antemano si una máquina de Turing (o un programa de ordenador) acabará por hallar su solución deseada y, por tanto, se detendrá.



Si se impone un límite de tiempo, este problema tiene muy fácil solución. Supongamos que deseamos saber si un programa dado llegará a detenerse en el plazo de un año. En tal caso, basta hacerlo funcionar durante un año y observar si se detiene o no. Lo que Turing hizo ver es que podemos encontrarnos en una dificultad muy seria si no se impone límite de tiempo, si tratamos de deducir si un programa se detendrá o no sin limitarnos meramente a hacerlo funcionar.

Trataré de esbozar el razonamiento de Turing. Supongamos posible la creación de un programa de ordenador capaz de averiguar si un programa, cualquiera que sea, llegará a detenerse. Llamémoslo, por comodidad, un "verificador de terminación". En teoría, le suministraríamos un programa y emitiría una respuesta: "sí, este programa terminará," o bien, "no, este programa seguirá haciendo girar sus ruedas en un bucle infinito y nunca llegará a detenerse". Preparemos ahora un segundo programa basado en el verificador de terminación. Consistirá en modificar el verificador de modo que, cuando se le entregue para examen un programa que termine, entre en un bucle infinito. Y aquí viene la parte sutil: suministre a su nuevo programa una copia de sí mismo. ¿Qué hará?

No olvide que ha preparado el nuevo programa de verificación de manera que entre en un bucle infinito si el programa sometido a prueba termina. Pero ahora el programa objeto de verificación es el propio programa verificador modificado. Por consiguiente, si terminase, habría de entrar en un bucle infinito, lo que significa que no termina: una contradicción. Tampoco sirve de nada suponer lo contrario. Si el programa no terminase, el verificador de terminación indicaría tal hecho, y el programa no entraría en bucle infinito, llegando, pues, a término. Esta paradoja llevó a Turing a considerar que sería imposible idear un verificador de terminación universal.

Lo más interesante es que Turing dedujo un corolario inmediato: Si no hay forma de determinar de antemano mediante cálculos si un programa va a dete-

2. EL ARTICULO DE ALAN TURING de 1936 introdujo la noción de "máquina capaz de realizar, casilla a casilla, operaciones sobre una cinta infinitamente larga". Esta construcción mental recibe desde entonces el nombre de "máquina de Turing". Este artilugio imaginario puede leer lo que está escrito en una casilla de la cinta. En función del estado interno de la máquina, deja tal cual o modifica esa casilla, desplaza la cinta un espacio hacia la izquierda o hacia la derecha, y repite el proceso. Turing demostró que un autómata así podría servirse de este sencillo procedimiento para llevar a cabo cualquier cálculo concebible, con tal de que se le proporcionara el conjunto adecuado de instrucciones básicas.

nerse o no, tampoco puede haber ningún modo de averiguarlo mediante razonamientos. Ningún sistema axiomático formal puede facultarnos para decidir si un programa acabará por detenerse. ¿Por qué? Porque si fuera posible utilizar a tal fin un sistema axiomático, éste nos proporcionaría los medios para calcular por adelantado si un programa se detendrá o no. Lo cual es imposible, pues se obtendría una paradoja del estilo de "¡esta aseveración es falsa!". Se puede crear un programa que se detiene si y solamente si no se detiene. La paradoja es similar a la descubierta por Gödel en sus investigaciones sobre la teoría de números. (Recordemos que no había dificultades mayores en el sistema que Gödel examinó que las que 0, 1, 2, 3..., la adición y la multiplicación ofrecen.) La proeza de Turing consistió en demostrar que *ningún* sistema axiomático formal puede ser completo.

Al desencadenarse la Segunda Guerra Mundial, Turing comenzó a trabajar en criptografía y von Neumann en el cálculo de detonaciones de bombas atómicas. El mundo dejó de lado durante un tiempo el problema de la incompletitud de los sistemas axiomáticos.

La aleatoriedad en la matemática

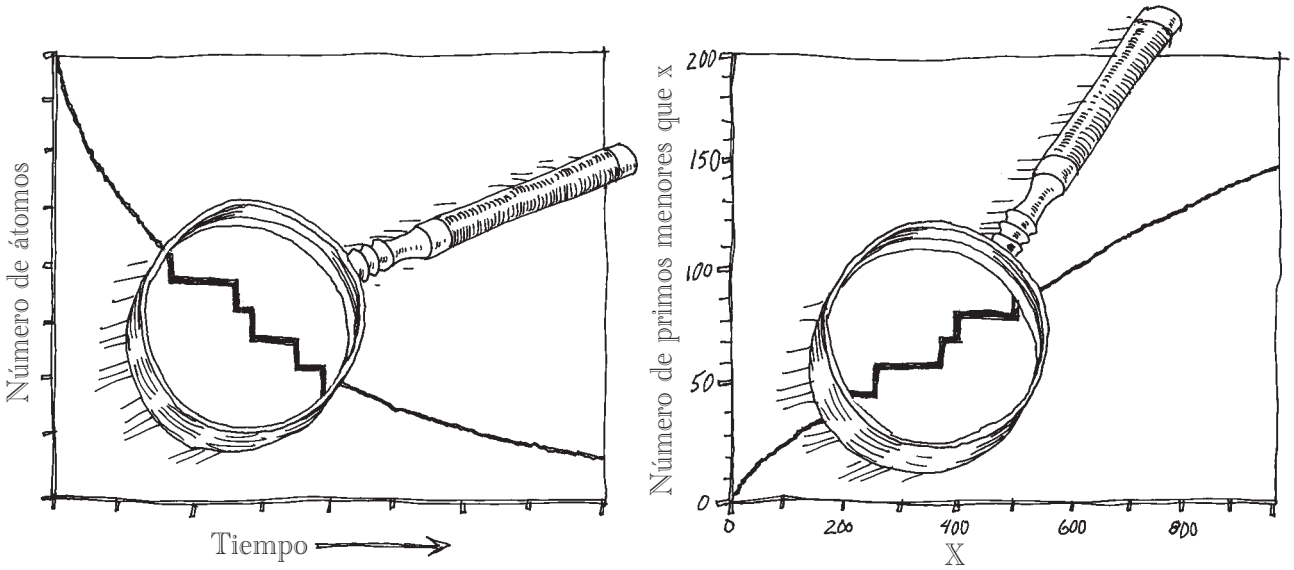
La generación de matemáticos preocupados por estas profundas cuestiones filosóficas quedó prácticamente extinta con la Segunda Guerra Mundial. Luego vine yo.

A finales de los años cincuenta, yo era casi un niño, leí en *Scientific American* un artículo sobre Gödel y la incompletitud. El resultado de Gödel me dejó fascinado, aunque en realidad no pude comprenderlo del todo; me pareció que había en él algo de dudoso. En cuanto al método de Turing, consideré que profundizaba mucho más, pero todavía no me sentía satisfecho. Fue por entonces cuando se me ocurrió una curiosa idea sobre la aleatoriedad.

De muchacho también leía mucho acerca de otra famosa cuestión intelectual, no la de los fundamentos de la matemática, sino la de los fundamentos de la física —sobre la teoría de la relatividad y la cosmología, e incluso más frecuentemente sobre la mecánica cuántica—. Aprendí que cuando las cosas son muy pequeñas, el mundo físico se comporta de una forma descabellada; en realidad, es aleatorio; es intrínsecamente impredictible. Estaba yo leyendo acerca de todo esto, y empecé a considerar si no habría también aleatoriedad en la matemática pura. Empecé a sospechar que ésa pudiera ser la verdadera causa de la incompletitud.

Hay un ejemplo que viene al caso en la teoría elemental de números, donde se plantean ciertas cuestiones muy difíciles. Tomemos los números primos. Si estamos interesados en su estructura de detalle, resulta que los números primos se comportan de forma muy impredictible. Es cierto que existen en ellos regularidades estadísticas. Se tiene, sea por caso, el llamado teorema de los números primos, que pronostica con muy buena precisión la distribución media de los números primos. Pero en lo que toca a la distribución detallada de cada número primo, parece a las claras aleatoria.

Empecé, pues, a pensar que pudiera ser que la aleatoriedad inherente a la matemática proporcionase una razón más profunda de toda esta incompletitud. A mediados de los años sesenta, A. N. Kolmogoroff, en la Unión Soviética, y yo, cada por su lado, aportamos nuevas ideas, a las que me gusta llamar "teoría al-



3. LA MECANICA CUANTICA refleja el papel de la aleatoriedad en la física. La desintegración de una sustancia radiactiva, en apariencia progresiva y regular, está compuesta en realidad por una serie de pasos discretos, siendo imposible predecir el momento exacto en que se desintegrará el átomo siguiente (*a la izquierda*). El trabajo del autor pone de relieve una aleatoriedad similar en la matemática, observable, por ejemplo, en la distribución de los números primos. Aunque el número de primos menores o iguales que *x* sigue una tendencia bien conocida, la curva está formada por una serie de pasos erráticos, no siendo posible predecir el valor exacto del número primo siguiente a partir de ninguna teoría general (*a la derecha*).

4. LAS INVESTIGACIONES DE KURT GÖDEL condujeron a la concepción moderna de la aleatoriedad como propiedad tan inherente a la matemática como a la física. Albert Einstein se resistía a aceptarlo. A pesar de ello, los dos fueron íntimos amigos en sus días de Princeton.

gorítmica de la información". Aunque el nombre parezca rimbombante, la idea fundamental es muy sencilla: se trata, simplemente, de medir la complejidad computacional.

Encontré una de las primeras referencias a la complejidad algorítmica de que yo tuviese noticia en un trabajo de von Neumann. Turing consideraba a la computadora como mero concepto matemático —una computadora perfecta, que jamás comete errores, que dispone de tanto espacio y tiempo como necesite—. Después de que Turing diese a conocer esta idea, el paso lógico siguiente para un matemático consistía en calcular el tiempo necesario para efectuar un cálculo; sería una medida de la complejidad de éste. Hacia 1950, von Neumann hizo resaltar la importancia de la complejidad temporal de los cálculos; hoy es una especialidad bien desarrollada.

Mi idea no era estudiar el tiempo, a pesar de que, desde un punto de vista práctico, sea muy importante, sino el *tamaño* de los programas informáticos, la cantidad de información que es necesario proporcionar a un ordenador para que realice una determinada tarea. ¿Por qué es interesante? Porque la noción de complejidad asociada al tamaño del programa se liga con la noción de entropía de la física.

Recordemos que la entropía desempeñó un papel crucial en los trabajos de un famoso físico del siglo XIX, Ludwig Boltzmann, y ocupa un lugar central en la mecánica estadística y en la termodinámica. La entropía mide el grado de desorden, caos, aleatoriedad de un sistema físico. La entropía de un cristal es pequeña; en un gas a temperatura ambiente, alta.

La entropía guarda relación con una cuestión filosófica de la mayor importancia, a saber: ¿por qué corre el tiempo en un solo sentido? En la vida ordinaria existe, desde luego, una gran diferencia entre la retrogradación y la progresión en el tiempo. Un vaso se rompe, pero no se recompone espontáneamente. De igual modo, en la teoría de Boltzmann la entropía tiene necesariamente que aumentar: el sistema ha de adquirir cada vez mayor desorden. Tal principio se denomina Segundo Principio de la Termodinámica.

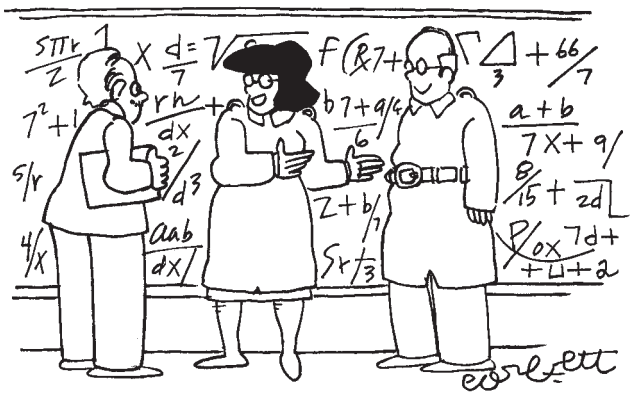
Los contemporáneos de Boltzmann no conseguían ver la forma de deducir este resultado a partir de la física newtoniana. Después de todo, en un gas, donde los átomos chocan y rebotan como si fueran bolas de billar, cada interacción es reversible. Si tuviéramos algún modo de filmar una pequeña porción de gas durante un breve tiempo, no podríamos saber, al ver la película, si estaba siendo pasada hacia delante o hacia atrás. Pero la teoría de los gases de Boltzmann afirma que existe una flecha del tiempo, que un sistema partirá de un estado ordenado y acabará en un estado muy desordenado y mezclado. Existe incluso un nombre amedrentador para la situación final: la "muerte térmica".

La relación entre mis ideas y la teoría de Boltzmann se debe a que el tamaño de un programa de ordenador es análogo al grado de desorden de un sistema físico. El programa necesario para especificar dónde se encuentran todos los átomos de un gas tendría que ser enorme; en cambio, para la descripción de un cristal no haría falta un programa tan grande, a causa de la regularidad de su estructura. La entropía y el tamaño de programa se encuentran, pues, íntimamente relacionados.

La noción de complejidad medida por el tamaño de programa guarda relación también con la filosofía del método científico. Ray Solomonoff (un científico informático que entonces trabajaba en Zator Company, en Cambridge, Massachusetts) propuso esa idea en 1960, en un congreso profesional; yo no tuve noticia de su trabajo hasta después de haber llegado por mí mismo, varios años después, a ideas muy parecidas. Basta pensar en el principio de "la navaja de Occam": la teoría más sencilla es la mejor. Ahora bien, ¿qué es una teoría? Es un programa de ordenador para la predicción de observaciones. Y el aserto de que la mejor teoría es la más sencilla se traduce en la afirmación de que un programa informático conciso constituye la teoría óptima.

¿Y si no existe una teoría concisa? ¿Y si el programa más breve capaz de reproducir un conjunto de datos experimentales es del mismo tamaño que el conjunto de datos? En este caso, la teoría no sirve de nada —es un amaño—; los datos resultarían incomprensibles, aleatorios. Una teoría sólo es buena en la medida en que comprime los datos hasta crear un sistema, mucho menor, de hipótesis teóricas y de reglas de deducción.

Así pues, podríamos definir lo aleatorio como lo que no puede ser comprimido. La única forma de describirle a alguien un objeto o un número que es com-



Somos un equipo: yo acentúo lo positivo y él elimina lo negativo.

pletamente aleatorio consiste en exhibírselo y decirle: "Aquí lo tienes". Dado que carece de estructura o de regularidad, no existe otra descripción más concisa. En el otro extremo se encuentran los objetos o los números que poseen una gran regularidad. Podría describirse uno de ellos diciendo, por ejemplo, que consiste en un millón de repeticiones de 01. He aquí un objeto muy grande que admite una descripción muy breve.

Mi idea consistía en utilizar la complejidad, medida por el tamaño de programa, para definir la aleatoriedad. Y en cuanto se empieza a examinar el tamaño de los programas de ordenador —en cuanto se toma en cuenta la noción de tamaño de programa o de complejidad de información en lugar de la de complejidad determinada por el tiempo de ejecución—, se produce un fenómeno interesante: allí donde miremos, encontraremos incompletitud. ¿Por qué? Porque la primera pregunta que se hace en mi teoría ya nos crea un conflicto. La complejidad de algo se mide por el tamaño del mínimo programa de ordenador que permite calcularlo. Pero, ¿cómo podremos estar seguros de que tenemos el mínimo programa? La respuesta es que no podremos. No es poco sorprendente: esa tarea escapa del alcance del razonamiento matemático.

La demostración de por qué es así resulta un tanto prolija, por lo que me limitaré a mencionar el resultado, que es uno de mis enunciados de incompletitud favoritos: Si tenemos *n* bits de axiomas, nunca será posible demostrar que un programa es el más breve posible si su tamaño supera *n* bits. Es decir, tendremos dificultades con un programa en cuanto sea de mayor tamaño que una versión computarizada de los axiomas; o más exactamente, si es mayor que el tamaño del programa de comprobación-demostración de los axiomas y de las reglas de deducción asociadas.

Resulta así que no es posible, en general, calcular la complejidad medida por el tamaño de programa, porque determinar la complejidad medida por el tamaño de programa de algo equivale a conocer el tamaño del más conciso de todos los programas que la calculan. Tal conocimiento no es posible si el programa es más extenso que los axiomas. Si hay *n* bits de axiomas, nunca se podrá determinar la complejidad medida por el tamaño de programa de nada que tenga más de *n* bits de complejidad, que es casi todo.

Explicaré por qué afirmo tal cosa. Los conjuntos de axiomas que normalmente utilizan los matemáticos son bastante concisos, pues de no serlo nadie creería en ellos. En la práctica, hay un vasto mundo de verdades matemáticas —una cantidad infinita de información— mientras que, por otra parte, cualquier conjunto dado de axiomas solamente abarca una cantidad finita, diminuta, de esa información. Tal es la razón, en pocas palabras, de que el teorema de incompletitud de Gödel sea, no misterioso y complicado, sino natural e inevitable.

¿Adónde, ahora?

Esta conclusión es muy impresionante. En sólo tres pasos se va desde Gödel, donde tan chocante resultaba ya que existiesen límites al razonamiento, a Turing, donde ya parecía mucho más razonable, y de Turing a una consideración de la complejidad medida por el tamaño de programa en la que la incompletitud, los límites de la matemática, no pueden sernos ya más notorios.

Suelen decirme: "Todo eso está muy bien. La teoría de información algorítmica es una bonita teoría; pero ahora no estaría de más un resultado concreto que escape, a su juicio, del alcance del razonamiento matemático". Durante muchos años, una de mis respuestas favoritas era: "Tal vez el Ultimo Teorema de Fermat". Pero en 1993, Andrew Wiles se presentó con una demostración. Contenía un paso en falso, pero ahora nadie duda de que sea correcta. La teoría algorítmica de la información hace ver que existen montones de cosas que no es posible demostrar, pero no permite llegar a una conclusión sobre cuestiones matemáticas sueltas.

¿Cómo es posible que, a pesar de la incompletitud, los matemáticos estén logrando tantísimos progresos? Sin duda, estos resultados de incompletitud parecen llevar consigo sentimientos pesimistas. Tomados sin más, podría parecer que no hay forma de avanzar, que la matemática es imposible. Felizmente para quienes nos dedicamos a la matemática, no parece que se cumpla esa condenación. Quién sabe: tal vez algún joven metamatemático de la próxima generación nos haga ver por qué ha de ser así.

Bibliografía complementaria

LA DEMOSTRACIÓN DE GÖDEL. E. Nagel y J. R. Newman, en *Sigma, el mundo de las matemáticas*, vol. 5, págs. 57-84. Editorial Grijalbo. Barcelona. 1958.

RANDOMNESS AND MATHEMATICAL PROOF. G. J. Chaitin, en *Scientific American*, vol. 232, n.º 5, págs. 47-52; 1975.

GÖDEL, ESCHER, BACH: AN ETERNAL GOLDEN BRAID. D. R. Hofstadter. Basic Books; Nueva York, 1979.

ARITMÉTICA Y AZAR. G. J. Chaitin, en *Investigación y Ciencia*, n.º 144, págs. 44-50; septiembre de 1988.

GÖDEL: A LIFE OF LOGIC. H. L. Casti W. DePauli. Cambridge, Massachusetts, Perseus Publishing, 2000.