Kurt Gödel y Alan Turing: una nueva mirada a los límites de lo humano

Claudio Gutiérrez

*Profesor Asociado Departamento de Ciencias de la Computación, Universidad de Chile.
Ph.D. Computer Science, Wesleyan University; Magíster en Lógica Matemática, Pontificia Universidad Católica de Chile; Licenciatura en Matemáticas, Universidad de Chile. Áreas de investigación: Fundamentos de la Computación, Lógica aplicada a la Computación, Bases de Datos, Semántica de la Web, Máquinas Sociales.*
cgutierr@dcc.uchile.cl

¿Cuál es la contribución intelectual que Gödel y Turing hicieron a la humanidad? Usualmente se sostiene que Gödel demostró que los sistemas formales son febles, esto es, los programas totalizadores de formalización del razonamiento están inevitablemente destinados a fracasar. Y en la misma línea, usualmente se sostiene que el gran aporte de Turing fue clarificar el fundamento teórico que subyace a los computadores y sus relaciones con los sistemas formales. En otras palabras, sus aportes habrían sido mostrar los alcances y las limitaciones intrínsecas del uso de lenguajes formales para representar el mundo que nos rodea.

En lo que sigue argumentaré que Gödel y Turing fueron mucho más allá de eso: revolucionaron nuestra manera de conceptualizar el entendimiento de lo humano. Cabe a Kurt Gödel haber llevado los argumentos sobre el *razonamiento humano* hasta sus últimas consecuencias, haber disecado esa noción hasta entender su esencia misma, haber desentrañado su mecanismo escondido. El gran aporte de Gödel fue demostrar –usando la lógica matemática para ello– que el razonamiento humano está indisolublemente ligado a una noción material, mecánica, la de *procedimiento*. Un procedimiento es una serie de acciones materiales que conducen a un resultado dado. Casi en paralelo, Alan Turing analiza la noción de procedimiento y, a partir de un análisis profundo de las acciones que normalmente lleva a cabo un humano para calcular, enlaza esa noción material con la noción abstracta de *función recursiva*, que el mismo Gödel había formulado en el intertanto.

Así, entre ambos, enlazan la noción de acción humana, de hacer humano, con una familia de formalismos, con un lenguaje de especificación. Podríamos decir que enlazan la materia con el espíritu, o para entendernos mejor, enlazan el *hardware* y el *software* humanos.

Como ocurre con todos los procesos verdaderamente revolucionarios, quien hace un forado al dique que sujeta el viejo sistema, inesperadamente es arrastrado por el subsiguiente aluvión sin que generalmente alcance a entender qué cambió. Es lo que le ocurrió a Gödel y a Turing. Entre ambos –sin proponérselo explícitamente– trenzaron sus ideas estableciendo los fundamentos y los alcances del razonamiento humano y una de las hipótesis más relevantes que tenemos hoy sobre las relaciones entre la materia y el espíritu. De esta historia hablaremos en lo que sigue.

GÖDEL Y TURING: CERCANÍAS Y DESENCUENTROS

Partamos presentando a nuestros dos protagonistas. Kurt Gödel (1906-1978) fue un matemático austríaco que hoy es considerado el lógico más relevante desde Aristóteles. Muy joven se interesó por la lógica y los fundamentos de las matemáticas. En 1931, cuando tenía 25 años, publicó su trabajo sobre la incompletitud de los sistemas formales que cambiaría para siempre nuestro entendimiento de los sistemas formales y la lógica. Gödel era una persona muy reservada, profunda y obsesiva (en su infancia su familia le apodaba “Señor Por qué”). Hay una anécdota que muestra muy bien su carácter. Al llegar a Estados Unidos arrancando del nazismo que había anexado Austria, para nacionalizarse debió estudiar la constitución de los Estados Unidos. Detallista, descubrió que este entramado lógico-legal permitiría una dictadura como la nazi. Sus amigos Einstein y Morgenstern intentaron desesperadamente convencerlo que no mencione eso en público, que era un caso muy poco probable de darse. Durante el examen de inmigración, apareció el tema de las formas de gobierno y Gödel orgulloso le indica al examinador que tiene una demostración de la posibilidad de una dictadura. Por suerte el funcionario conocía a Einstein y las extravagancias de estos personajes, y rápidamente le cambió de tema. Gödel vivió tranquilo el resto de su vida en Princeton y al final de sus días se obsesionó con que alguien lo podía envenenar; así murió de malnutrición crónica. Una biografía completa de Gödel es la de Dawson [4]. Los recuerdos de su amigo Hao Wang [14] sobre él son muy interesantes también.

Alan Turing (1912-1954) fue un matemático inglés, de amplios intereses científicos. También desde muy joven mostró signos de genio en el estudio de las matemáticas y motivado por los resultados de Gödel se interesó en la lógica matemática. Captó rápidamente la esencia misma del resultado de Gödel y por su cuenta, sin guía formal, dio el siguiente paso (que no vislumbró Gödel) desarrollando su famoso formalismo (conocido hoy como “máquina de Turing”) que captura la noción de procedimiento “efectivo” (esto es, que efectivamente puede realizarse). Con esos antecedentes se va a Princeton donde continúa estudiando con Alonzo Church y se doctora en 1938. Vuelve a Inglaterra en los años iniciales de la Segunda Guerra y se enrola entre los científicos que trabajaban en asuntos militares, en su caso, descifrando códigos enemigos. Después de la guerra, se dedica a construir el primer computador inglés. Escribe sobre diversas áreas. El Gobierno inglés lo persiguió por su condición homosexual y debido a ello, también muere trágicamente. Una excelente biografía es [10]. Su obra está en línea en http://www.turing.org.uk/.

Aunque Turing y Gödel nunca se encontraron físicamente (al parecer tampoco intercambiaron correspondencia), cada uno "descubrió" al otro a su manera. Cuando estaba en Princeton, Turing escribía a su madre: "*Aquí hay un gran número de los más distinguidos matemáticos: J. v. Neumann, Weyl, Courant, Hardy, Einstein, Lefschetz y otros. Desafortunadamente no hay muchos lógicos. Church está aquí, pero Gödel, Kleene, Rosser y Bernays que estuvieron el año pasado, se fueron. No creo que me importe mucho ninguno excepto Gödel*". En la constelación de genios de Princeton, Einstein y Von Neumann incluídos, a Turing sólo le interesó Gödel.

La admiración (secreta) era recíproca. También a Gödel lo que más le llamó la atención de la impresionante obra lógica de los años treinta, fue la obra de Turing. Ese formalismo logró convencerlo que finalmente se había encontrado la representación de la noción de procedimiento efectivo que los matemáticos andaban buscando hace mucho tiempo. En particular, Gödel le vió una consecuencia muy relevante que se deducía de él y escribió: "*Debido al trabajo de A. M. Turing, es posible dar ahora una definición precisa e incuestionable del concepto general de sistema formal*". Ahora es Gödel quien ve más allá del propio Turing. Cuando Turing se adentraba en otros campos dejando atrás sus investigaciones teóricas sobre lo computable, Gödel continuó la reflexión sobre ello.

ANTECEDENTES

Debemos fijar un comienzo, y lo haremos en el cambio del siglo XIX al XX. Probablemente se trata del período fundacional de lo que será el gran salto conceptual que darán Gödel y Turing en los años treinta del siglo XX.

El origen de esta revolución conceptual puede remontarse a la concepción del mundo que se inicia con la filosofía y la ciencia moderna, y a su programa para instalar el análisis, la descomposición de un fenómeno complejo en sus partes simples constituyentes, como metodología para evitar errores en el razonamiento y profundizar el conocimiento. Los pensadores y científicos advierten sobre los problemas que trae al conocimiento la vaguedad del lenguaje natural y la falta de precisión en los conceptos.

En el área de la lógica, el fondo de estas inquietudes podemos presentarlo hoy de la siguiente forma. Si el razonamiento no es más que una sucesión finita de argumentos muy precisos, cada uno encadenado al siguiente por mecanismos también precisos, lo que los matemáticos llaman una demostración; ¿es posible llevar cada uno de esos encadenamientos, cada uno de esos eslabones, a un extremo de claridad y precisión de tal forma que sean "evidentes" (esto es, no se necesite "inteligencia" para verificarlos)? De esta pregunta surgieron dos temas que desarrollaremos en lo que sigue: por una parte, *los alcances de la formalización tanto del lenguaje como de la lógica* y, por otro, *los mecanismos de descripción de conceptos y objetos matemáticos* de forma que sean susceptibles de ser "construidos" a partir de otros básicos y evidentes.

La formalización del lenguaje

A comienzos del siglo XX la comunidad lógico-matemática trabajaba febrilmente en un programa propuesto por el renombrado matemático David Hilbert, cuyos antecedentes se pueden rastrear en los escritos de lógicos medievales como Raimundo Lulio, luego en Leibniz, y en quienes soñaron con "mecanizar" el razonamiento. Este programa consistía en la *formalización total del razonamiento matemático* y la reducción del razonamiento a sus partes atómicas evidentes. El siguiente párrafo de Hilbert ilustra muy bien sus motivaciones:

> "Si se quiere que la inferencia lógica sea confiable, debe ser posible inspeccionar esos objetos [concretos matemáticos] completamente en todas sus partes, y que junto con esos objetos, se determine inmediata e intuitivamente el hecho de que estén allí, que difieren unos de otros, y que uno sigue a otro, o que estén concatenados, como algo que no puede ser reducido a ninguna otra cosa que requiera reducción. Ésta es la posición filosófica básica que considero requisito para las matemáticas, y en general, para todo pensamiento, entendimiento y comunicación científica [9]".

Como vemos, el programa tenía fuertes motivaciones prácticas. En particular, los fundamentos del análisis matemático, especialmente el tratamiento del sospechoso concepto de infinito y de número infinitesimal hacía imperiosa la necesidad de contar con algún sistema formal que permita razonar sobre estos objetos sin incurrir en errores (por ejemplo, las llamadas paradojas). Los fundamentos mismos de la ciencia "más segura", la ciencia "exacta" por antonomasia, se veían temblorosos.

A comienzos del siglo XX se habían conseguido grandes avances. El monumental tratado *Principia Mathematica* de Whitehead y Russell por un lado, y la emergente *teoría de conjuntos* de Zermelo-Fraenkel por otro, entregaban sistemas formales en los cuales se podía expresar formalmente toda la matemática conocida. Sólo quedaba entonces demostrar que esos sistemas eran coherentes, esto es, no tenían inconsistencias internas. Es ésta la tarea que se propuso el joven Kurt Gödel a comienzos de los años treinta, y cuyo sorprendente resultado analizaremos más adelante.

Transformaciones efectivamente calculables

Junto al infinito, hay otra noción que transformó las matemáticas modernas: el concepto de función. Aunque ahora nos parezca trivial, la noción de función, el tratarla como un objeto en sí, no es nada sencillo. Lo que se está abstrayendo realmente es la noción de transformación, que tanto dolor de cabeza ha producido a los filósofos.

Y cuando se junta la noción de función con la de infinito, aparecen problemas serios. Una función (en matemáticas) transforma números (en general objetos matemáticos) en otros. ¿Cómo especificar una tal transformación, especialmente si la cantidad de objetos sobre los que se aplica es infinita? Fue Richard Dedekind en 1888 el primero que propuso un método –muy elemental por lo demás– que pasaría a la historia como *definición inductiva o recursiva*. Dedekind ya hablaba así: "Por una transformación T entenderemos una ley según la cual a cada elemento determinado s de un sistema S le corresponde una cosa determinada" y propone que para transformaciones sobre los números naturales, se especifique su valor para 1 y se reduzca su cómputo para un número cualquiera al cómputo del anterior mediante otra función Θ de la siguiente manera[1]:

$$T(1) = \omega,$$
$$T(n+1) = \Theta(T(n)).$$

La definición es algo limitada. Por ejemplo, no es posible especificar en este formalismo la función factorial $f(n+1) = (n+1)\,f(n)$. Peano generaliza esta definición y establece en 1889 una noción más amplia conocida posteriormente como *recursión primitiva*. Una función f de varias variables se especifica así:

$$(1)\ f(0, y_1,\ldots, y_n) = h(y_1,\ldots, y_n)$$
$$(2)\ f(x+1, y_1,\ldots, y_n) = g(x, f(x, y_1,\ldots,y_n), y_n,\ldots, y_n)$$

donde g y h son funciones definidas previamente por recursión también. Nótese que esta definición es mucho más general y poderosa. En particular, la función factorial se puede especificar ahora poniendo $f(0) = 1$ y $f(x+1) = mult(x, f(x))$, donde *mult* es la multiplicación usual de enteros que se puede definir a su vez por las fórmulas (1) y (2)[2].

La gracia de todas estas definiciones es que dadas la entradas (el *input*), es posible *mecánicamente*, esto es, sin gran esfuerzo mental y sin necesidad de "ingenio", calcular el valor de la función para esa entrada en un número finito de pasos. Esto es lo que en la época le llamaban una *función efectivamente calculable*. ¿Cuánto es posible generalizar esta idea? ¿Es posible tener un formalismo en el que se pueda especificar todas las transformaciones que son efectivamente calculables? Este problema está estrechamente ligado al anterior sobre los *procedimientos* efectivos en un razonamiento. La transformación (función) aparece allí como el eslabón que permitiría enganchar un argumento con otro en una demostración o razonamiento.

EL COMIENZO: GÖDEL 1931

Hacia 1930 el joven Kurt Gödel trabajaba en el programa de Hilbert. En particular, intentaba demostrar la consistencia de la aritmética, que se consideraba el fundamento del análisis matemático clásico. Fue él quien se tomó más en serio ese programa y fue también quien terminó por destruirlo. Esto es, intentando demostrar la consistencia de la aritmética, descubrió que ese era un objetivo imposible.

El trabajo de Gödel, titulado *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme* ("Sobre sentencias formalmente indecidibles de Principia Mathematica y Sistemas afines"), de modestas 25 páginas, fue escrito el año 1930 y publicado en 1931 en la revista *Monatshefte für Mathematik und Physik*. Allí Gödel se propone como objetivo principal demostrar lo que hoy se conoce como el "Teorema de incompletitud de Gödel". Dejemos que él mismo nos explique, con las palabras introductorias de su artículo, en qué consiste su contribución:

> "Como es sabido, el progreso de la matemática hacia una exactitud cada vez mayor ha llevado a la formalización de amplias partes de ella, de tal modo que las deducciones pueden

[1] No es difícil mostrar que es suficiente contar con una buena definición sólo para los números naturales. Para quienes quieran conocer estas ideas fundacionales de Dedekind, recomiendo el excelente librito de R. Chuaqui: "¿Qué son los números? El Método Axiomático". Edit. Univ. 1980. Allí traduce y comenta la obra central de Dedekind sobre este tema. La cita es la Definición 21, p. 45, y la definición de función recursiva es el Teorema de la Definición por Inducción, p. 88.

[2] Ejercicio para el lector: esto último no es totalmente cierto. Hay una linda historia asociada a este detalle. Resulta que con la formulación de Peano no es posible definir la multiplicación pues necesita recursión doble. Nadie lo había advertido, hasta que Karl Grandjot –la historia transcurre en Göttingen– puso la observación en los márgenes de un libro que su profesor Edmund Landau le prestó para hacer las clases. El genial Karl Grandjot luego se vino a Chile y fue profesor de la Universidad de Chile durante el resto de su vida. Ver [8].

llevarse a cabo según unas pocas reglas mecánicas. Los sistemas formales más amplios construídos hasta ahora son el sistema de *Principia Mathematica* (PM) y la teoría de conjuntos de Zermelo-Fraenkel (desarrollada aún más por J. von Neumann).

Estos dos sistemas son tan amplios que todos los métodos usados hoy día en la matemática pueden ser formalizados en ellos, es decir, pueden ser reducidos a unos pocos axiomas y reglas de inferencia. Resulta por tanto natural la conjetura de que estos axiomas y reglas basten para decidir *todas* las cuestiones matemáticas que puedan ser formuladas en dichos sistemas[3]. En lo que sigue se muestra que esto no es así, sino que por el contrario, en ambos sistemas hay problemas relativamente simples de la teoría de los números naturales que no pueden ser decididos con sus axiomas (y reglas)".

Su resultado formalmente dice así:

Teorema (de Incompletitud). Sea K un conjunto recursivo primitivo de fórmulas, y S un sistema deductivo para K que cumple ciertos requisitos mínimos[4], y sea $dem(G)$ el conjunto de proposiciones que se pueden demostrar en el sistema formal a partir del conjunto de fórmulas G. Entonces hay una proposición A en K tal que $\sim A$ pertenecen a $dem(K)$.

Expresemos el significado profundo, las consecuencias, de este resultado con sus propias palabras:

"Todos ustedes conocen las famosas palabras de Hilbert sobre que todo matemático está convencido de que para cada pregunta matemática precisamente formulada es posible encontrar una única respuesta y que es exactamente esta convicción el estímulo fundamental del trabajo de investigación matemático. Hilbert mismo estaba tan convencido de esto que aún pensaba que era posible que se diera una demostración matemática de esto, al menos en el dominio de la teoría de los números.

¿Cómo podemos imaginar que una tal demostración pueda ser obtenida? Para buscarla primero tenemos que analizar el significado del teorema a ser demostrado. Para cada hombre desprejuiciado esto sólo puede significar lo siguiente: dada una proposición matemática cualquiera *A,* existe una demostración para *A* o para no-*A*, donde por "demostración" se entiende algo que parta de axiomas evidentes y proceda por inferencias evidentes. Ahora, formulado de esta manera, el problema no es susceptible de tratamiento matemático pues involucra las nociones no-matemáticas de evidencia. Luego lo primero que hay que hacer es explicitar esta noción a través del análisis de las demostraciones matemáticas reales. Si eso se hace –y ha sido hecho por la lógica matemática y por la *Teoría de la Demostración* de Hilbert– entonces nuestro problema puede ser tratado matemáticamente y la respuesta resulta ser negativa aún en el dominio de la teoría de los números.

Pero es claro que esta respuesta negativa tiene dos significados diferentes: (1) puede significar que el problema en su formulación original tiene una respuesta negativa, o (2) puede significar que algo se perdió en el proceso de transición de la evidencia al formalismo. Es fácil ver que realmente ocurrió lo segundo, puesto que las preguntas de la teoría de los números que son indecidibles en un formalismo dado son siempre decidibles por inferencias evidentes que no son expresables en el formalismo dado. Respecto de la evidencia de estas nuevas inferencias, ellas resultan ser tan evidentes como las del formalismo dado. Luego el resultado es más bien que no es posible formalizar la evidencia matemática aún en el dominio de la teoría de los números, pero la convicción acerca de la cual Hilbert hablaba permanece enteramente intocada".

Es importante destacar los asuntos cruciales que están involucrados en la argumentación anterior: la noción de sistema formal y la de inferencia. Está implícito aquí el hecho que esta inferencia y este sistema formal deben expresarse por medio de transformaciones "efectivas".

INTERMEDIO: FUNCIONES RECURSIVAS Y TESIS DE CHURCH

En paralelo al espectacular resultado de Gödel, continuaba la búsqueda de formalismos que especificaran la noción de función efectiva (ver los

[3] Un sistema que cumpla esto, o sea, que pueda decidir todas las cuestiones que puedan ser formuladas en él se dice *completo*. En 1930 el mismo Gödel había demostrado en otro célebre teorema, que la lógica (pura) de primer orden (la lógica "usual") era completa. Esto significa que los axiomas y reglas de deducción usuales son suficientes para decidir todas las cuestiones lógicas (puras) que pueden ser formuladas allí. Cuando se le agrega la teoría de números eso ya deja de ser cierto, y el sistema se torna incompleto.

[4] Estoy evitando detalles técnicos engorrosos. "Requisito mínimo" refiere esencialmente a la aritmética. Para una exposición simple de la idea central de la demostración ver [7]. Para la demostración misma, ver [6].

detalles de esta entretenida historia en [1]). Volvamos a la definición (algo ingenua) de Peano, y veamos qué más necesitamos para especificar completamente el sistema. Después de un análisis simple, surge la necesidad de definir las siguientes nociones:

(3) La función sucesor
$s(x) = x + 1$,
que aparece como noción primitiva (no definida);
(4) La función constante
$f(x_1, \ldots, x_n) = 0$
(que también aparece como noción primitiva; no es difícil ver que cualquier otra función constante se puede definir a partir de ésta);
(5) La función proyección
$\pi_k(x_1, \ldots, x_n) = x_k$
(para pasar subconjuntos de argumentos a otra función);
(6) La composición de funciones u operador de substitución (que se usa en varias partes).

Agregando las nociones (3) a (6) a la definición de Peano (ecuaciones (1) y (2) más arriba), queda completamente definida la noción de *función recursiva primitiva.*

Pronto quedó en evidencia que aún este formalismo no lograba capturar todo lo que era efectivamente calculable. Uno de los contraejemplos más conocidos es la función de Ackermann (un alumno de Hilbert). Esta extraña función de tres argumentos $\varphi(m,n,p)$ no se puede especificar con el formalismo de funciones recursivas primitivas, y declara, por iteración, operaciones aritméticas cada vez más complejas:

$$\begin{aligned}
\varphi(m, n, o) &= m + 1 + \cdots + 1 \ (\, n \text{ veces} \,) \\
&= m + n \ (\textit{suma}) \\
\varphi(m, n, 1) &= m + \cdots + m \ (\, n \text{ veces} \,) \\
&= m \cdot n \ (\textit{multiplicación}) \\
\varphi(m, n, 2) &= m \cdots m \ (\, n \text{ veces} \,) \\
&= m^n \ (\, \textit{exponenciación} \,) \\
\varphi(m, n, 3) &= m^{m^{\cdot^{\cdot^{m}}}} \ (\, n \text{ veces} \,) \\
&= \cdots \ (\, \textit{sin nombre conocido} \,) \\
\varphi(m, n, 4) &= \textit{ejercicio para el lector}
\end{aligned}$$

Una vez más Gödel, esta vez inspirado por sugerencias del lógico francés Jacques Herbrand, consigue desarrollar un formalismo (basado en ecuaciones funcionales) que logra extender la noción de función efectiva para incluir funciones más generales como la de Ackermann; formalismo que hoy se conoce como *funciones recursivas.*

El lógico Stephen Kleene simplificó la formulación de Gödel, agregando a la especificación que tenemos (ecuaciones (1) - (6)) el siguiente operador, al que bautizó como operador μ, y que está definido para una función primitiva recursiva $f(x, y_1, \ldots, y_n)$ de la siguiente manera:

$$(7)\ \mu(x) f(x, y_1, \ldots, y_n) = \min_{z \geq 0} \exists y_0 \ldots \exists y_n f(z, y_1, \ldots, y_n) = 0$$

En castellano: el valor de la función μf se obtiene "buscando", a partir de $z = 0$ números $y_1, \ldots, y_n$ que satisfacen $f(z_0, y_1, \ldots, y_n) = 0$. Si no existen tales números para $z = 0$, seguimos buscando con $z = 1$, y así sucesivamente, hasta encontrar los primeros $z_0, y_1, \ldots, y_n$ que hacen $f(z_0, y_1, \ldots, y_n) = 0$. Allí la búsqueda se detiene y se retorna z_0. Nótese que con esto se introduce la posibilidad de que una función no retorne ningún valor; esto es, que no sea total[mente] definida para todos los valores de sus argumentos. Por ello muchas veces se llama a este conjunto funciones recursivas *parciales.*

Tesis de Church. En la búsqueda de formalismos que capturaran la noción de transformación, Alonzo Church, de Princeton, propuso poco después que Gödel y Kleene formalizaran la noción anterior, el *cálculo lambda*. Este cálculo captura la noción abstracta de función a partir de un análisis de la noción de evaluación [2].

Consideremos la función que saca la raíz cuadrada de un número. Cuando se escribe $\sqrt{3}$, lo que el común de los matemáticos entiende es que éste es el valor que retorna la función al evaluarla en 3. Pero, ¿cómo escribir –especificar– entonces la sola aplicación de la función "sacar raíz cuadrada" aplicada a a, pero sin evaluarla aún? Para ello, en primer lugar, hay que tener un formalismo para describir funciones (sin que necesariamente se apliquen a nada aún). Church usa la notación $\lambda x[M]$, que interpretada como programa computacional, consta de λx, que indica una "variable" x, y M que especifica el "código" o "programa" que usa esa variable. En segundo lugar, es necesario definir la noción de aplicación de la función (nuestro $\lambda x[M]$) sobre un argumento (que denotaremos A, y que es otra expresión cualquiera del cálculo lambda), y que Church denotó $\{\lambda x[M]\}(A)$. La evaluación entonces sera el "procedimiento" (¡aquí aparece la noción de procedimiento mecánico!) que consiste en reemplazar cada ocurrencia de la variable x en M por el argumento A. Por ejemplo al evaluar $\{\lambda x \sqrt{x^2+2x}\}(3)$ resulta, $\sqrt{3^2+2 \bullet 3}$, y si evaluamos las operaciones aritméticas básicas en un segundo paso llegamos a $\sqrt{15}$. Nótese que con este formalismo se pueden aplicar funciones a funciones, etc[5].

En un artículo publicado en 1936, *An unsolvable problem of elementary number theory* ("Un problema insoluble de la teoría elemental de los números"), basado en ideas presentadas informalmente el año anterior, Church demuestra que este forma-

[5] Y aparecen también expresiones "infinitamente recursivas", esto es, que no terminan nunca de evaluarse. Por ejemplo {λx[{x}(x)]}(λx[{x}(x)]) en la notación horrible de Church, o (λx(xx))(λx(xx)) en notación actual. Nótese que al evaluar esta expresión, esto es al reemplazar la variable x en la expresión de la izquierda por el argumento λx(xx), queda la misma expresión. Luego hay que evaluarla de nuevo y así sucesivamente.

lismo es equivalente a (puede especificar las mismas funciones que) el conjunto de las funciones recursivas parciales. Lo que quedó para la historia es que a partir de esta "evidencia", Church avanza la audaz tesis de que esta noción captura precisamente el concepto de función efectivamente calculable (Gödel tiempo antes había comentado en privado sobre las relaciones entre el formalismo de las funciones recursivas y lo efectivamente calculable, pero consideró que la evidencia recogida hasta ese momento no era suficiente para asociarlas). Church formula así su tesis:

"El propósito del presente artículo es proponer una definición de calculabilidad efectiva que se piensa que corresponde satisfactoriamente con la algo vaga intuición en términos de los cuales los problemas [para los cuales se requiere una función efectivamente calculable] son formulados, y mostrar, por medio de un ejemplo, que no todos los problemas de esta clase son solubles".

La afirmación anterior es lo que se conoce como Tesis de Church. "Tesis" porque esta afirmación no es susceptible de demostración, pues relaciona un formalismo matemático con una noción ("efectividad") que es, digamos, filosófica. Como indicamos, aunque Gödel la intuyó antes, seguía considerando que no existía suficiente evidencia para sostenerla. Esto cambió cuando apareció el artículo de Turing de 1936.

FINALE: TURING 1936

En 1936 Alan Turing –de sólo 25 años– publicó, casi simultáneamente con el artículo de Church mencionado anteriormente, su trabajo titulado *On Computable Numbers with an Application to the Entscheidungsproblem* ("Sobre números computables con una aplicación al problema de decisión"). En él analiza en detalle el proceso de razonamiento (de "cómputo") humano y en base a ello propone una máquina que simula ese mismo proceso. A continuación, demuestra que ese formalismo es equivalente con la noción de función recursiva.

Consideramos interesante reproducir ese análisis de Turing, que hoy sorprende por su sencillez (el artículo original está reproducido en [3], pero es fácil encontrarlo en la Web):

"Un cómputo se hace normalmente escribiendo ciertos símbolos en papel. Podemos suponer que este papel está normalmente dividido en cuadrados, como los cuadernos de aritmética de los niños. En aritmética elemental se usa a veces el carácter bidimensional del paper. Pero tal uso es evitable, y creo que se puede convenir que este carácter bidimensional no es esencial para el cómputo. Asumo entonces que el cómputo se lleva a cabo en un papel de una dimensión, esto es, en una cinta dividida en cuadrados.

Supondré también que el número de símbolos que se puede imprimir es finito. Si permitiéramos un número infinito de símbolos, entonces habría símbolos que difieren en un grado arbitrariamente pequeño. El efecto de esta restricción sobre el número de símbolos no es muy seria. Siempre es posible usar secuencias de símbolos en lugar de símbolos individuales. Luego un numeral arábico como *17 o 999999999999999* se tratará normalmente como un símbolo. Similarmente en cualquier lengua europea las palabras son tratadas como símbolos individuales (el chino, sin embargo, intenta tener una cantidad infinitamente enumerable de símbolos). Las diferencias, desde nuestro punto de vista, entre símbolos individuales y compuestos es que los compuestos, si son muy largos, no pueden ser observados con una sola mirada. Esto es de acuerdo a mi experiencia. No podemos decir con una mirada si *9999999999999999 o 999999999999999* son iguales.

El comportamiento de quien computa en cada momento está determinado por los símbolos que está observando, y su "estado mental" en ese momento. Podemos suponer que hay una cota B al número de símbolos o cuadrados que puede observar en cada momento quien computa. Si quiere observar más, debe usar observaciones sucesivas. Podemos suponer también que el número de estados mentales que necesitan ser considerados es finito. Las razones para ello son del mismo carácter que restringen el número de símbolos. Si admitiéramos infinitos estados mentales, algunos de ellos estarían "arbitrariamente cercanos" y se confundirían. De nuevo, esta restricción no afecta seriamente el cómputo, puesto que el uso de estados más complejos se puede evitar escribiendo más símbolos en la cinta.

Imaginemos ahora las operaciones que hace quien computa descompuestas en "operaciones simples" que son tan elementales que no sea fácil imaginarlas divididas más aún. Cada tal operación consiste en algún cambio de estado del sistema, el cual se compone de quien computa y su cinta. Conocemos el estado del sistema si conocemos la secuencia de símbolos sobre la cinta, cuáles de ésos son observados por quien computa (posiblemente en algún orden especial), y el estado mental de quien computa. Podemos suponer que en una operación simple no se altera más de un símbolo. Cualquier otro cambio puede descomponerse en cambios simples de este tipo. La situación respecto a los cuadrados cuyos símbolos pueden ser alterados de esta manera, es la misma que la de aquellos cuadrados que sólo se observan. Podemos, por consiguiente, asumir sin pérdida de generalidad que los cuadrados cuyos símbolos se cambian son siempre cuadrados "observados".

Aparte de estos cambios de símbolos, las operaciones simples deben incluir cambios de distribución de los cuadrados observados. Los nuevos cuadrados observados deben ser inmediatamente reconocibles por quien computa. Pienso que es razonable suponer que ellos pueden ser sólo cuadrados cuya distancia desde el más cercano de los cuadrados previamente observados no exceda cierta cantidad fija. Digamos que cada uno de los nuevos cuadrados observados está a L cuadrados [de distancia] de un cuadrado observado inmediatamente antes. [...]

Podemos ahora construir una máquina que haga el trabajo de quien computa. Para cada estado mental de quien computa corresponde una "m-configuración" de la máquina ["m" por máquina]. La máquina escudriña [escanea] B cuadrados correspondientes a los B cuadrados observados por quien computa. En cualquier movimiento la máquina puede cambiar un símbolo sobre un cuadrado escudriñado o puede cambiar cualquiera de los cuadrados escudriñados por otro cuadrado distante no más de L cuadrados de alguno de los otros cuadrados escudriñados. El movimiento que se hace, y la subsecuente configuración, están determinados por el símbolo leído y la m-configuración".

Luego de hecho este análisis, Turing propone el formalismo conocido hoy como "máquina de Turing", que es simplemente la abstracción de cada una de las nociones mencionadas anteriormente. Una cinta cuadriculada infinita de una dimensión, un alfabeto finito de símbolos, un cabezal que puede "leer" y "escribir" símbolos en cada uno de esos cuadrados, y que puede moverse de un cuadrado a su vecino a izquierda o derecha, y un sistema de "estados". La operación de la máquina está codificada esencialmente por una función que a cada par (símbolo-leído, estado) le asocia: un movimiento a izquierda o derecha en la cinta; la escritura de un símbolo en el nuevo cuadrado de la cinta; y un nuevo estado. Formalmente:

Una *máquina de Turing* es un cuádruple (K,S,e,δ) donde:

1. K es un conjunto finito de estados. Hay un estado especial h que se llama estado de detención *(halt state)*.
2. S es un conjunto de símbolos (el alfabeto de la máquina). Hay además tres símbolos extras: el símbolo # (que simula el cuadrado blanco), y otros dos, D e I (que indican el movimiento a derecha e izquierda respectivamente).
3. e es el estado inicial con el cual parte la máquina.
4. δ es una función de $K \times S \rightarrow (K \cup \{h\}) \times (S \cup \{D,I\})$.

La máquina funciona así: si está en el estado q y el cabezal leyendo el símbolo a, entonces el valor de la función $\delta(q,a)$ indica qué hacer de la siguiente manera: si $\delta(q,a)=(p,b)$, entonces se sobrescribe el símbolo a en el cuadrado que se está leyendo con el símbolo b y se cambia el estado a p. Si $\delta(q,a)=(p,D)$, se mueve el cabezal a la derecha y cambia el estado a p. Análogamente con $\delta(q,a)=(p,I)$. Finalmente, si $\delta(q,a)=(h,-)$, donde $-$ indica cualquier símbolo, la máquina se detiene.

Turing muestra –entre muchos otros resultados– que este formalismo es equivalente a los formalismos de Church y de Gödel, esto es, describe exactamente las funciones recursivas parciales.

Es posible mirar esta contribución de Turing al menos desde dos perspectivas. La primera, como la presentación de un formalismo más (esta vez muy intuitivo y sencillo, tanto, que es el que hoy se usa para enseñar la teoría de computabilidad) equivalente a la noción de funciones recursivas parciales. La segunda, la que a nosotros nos interesa aquí, como un análisis cuidadoso de los límites de la noción de "procedimiento efectivo". Gödel expresa muy bien esto último:

"El trabajo de Turing da un análisis del concepto de "procedimiento mecánico" (alias "algoritmo" o "procedimiento computacional" o "procedimiento combinatorial finito"). Y *demuestra* que este concepto es equivalente con el de Máquina de Turing".

Destaqué el "demuestra" que usa Gödel. Esto es, Gödel –quien es el arquetipo de la precisión y rigurosidad lógica– considera que el análisis de Turing (particularmente la sección 9 de su artículo) *demuestra* que los procedimientos mecánicos son equivalentes a la noción de función recursiva. En otras palabras, que la noción de función recursiva captura la noción de procedimiento efectivo.

Muy poco después de conocer el resultado de Turing, Gödel reformula –generaliza– de la siguiente manera su resultado de incompletitud de 1931:

"No es posible mecanizar el razonamiento matemático, esto es, nunca será posible reemplazar los matemáticos por una máquina, aún si nos confinamos a los problemas de la teoría de números. Hay por supuesto, porciones de las matemáticas que pueden ser completamente mecanizadas y automatizadas; por ejemplo, la geometría elemental es una de ellas, pero la teoría de los números enteros ya no lo es".

¿Por qué no se podía afirmar esto antes del resultado de Turing de 1936? El mismo Gödel lo explica:

"Cuando publiqué mi artículo sobre proposiciones indecidibles el resultado no podía ser expresado en esta generalidad, debido a que en ese tiempo no se habían dado definiciones matemáticamente satisfactorias de las nociones de procedimiento mecánico y sistema formal. Esta brecha ha sido llenada por Herbrand, Church y Turing. El punto esencial es definir qué es un procedimiento. Entonces la noción de sistema formal se sigue fácilmente puesto que una teoría formal está dada por tres cosas:

1. Un número finito de símbolos primitivos cuyas combinaciones finitas son llamadas expresiones;
2. Un conjunto finito de expresiones (llamadas axiomas);
3. Ciertas, así llamadas, "reglas de inferencia";

Ahora, una regla de inferencia no es nada más que un procedimiento mecánico que permite a uno determinar si a partir de conjunto finito de expresiones es posible inferir algo por medio de la regla de inferencia usada, y así escribir la conclusión".

Pero la historia no termina todavía...

¿MÁS ALLÁ DE TURING?

Todos los esfuerzos posteriores por generalizar y ampliar la definición de función recursiva han confirmado la intuición de Church y los argumentos de Turing y Gödel que se había capturado –en la expresión de Gödel– la *noción epistemológica* de procedimiento efectivo. Aparte de los sistemas ecuacionales de Gödel-Herbrand, el cálculo lambda de Church, la *mu*-recursividad de Kleene, que hemos visto, se han propuesto muchos otros. Entre los más conocidos están los sistemas de Post (sistemas combinatoriales de símbolos); los sistemas de Markov (una suerte de gramáticas formales que capturan las funciones recursivas); los diagramas de flujo; la familia de máquinas con contadores *(counting machines)* que esencialmente agregan un registro para almacenar números enteros; las máquinas de acceso aleatorio (RAM) que pueden considerarse máquinas con contadores con la posibilidad de accesar registros de manera indirecta con instrucciones almacenadas.

En particular, la noción de máquina de Turing ha sido analizada desde múltiples perspectivas, intentando encontrar extensiones que pudieran capturar formas hoy desconocidas de "efectividad". Ninguna ha tenido éxito. Destaquemos aquí algunos de ellos. El más sistemático es probablemente el intento de Kolmogorov y su escuela, lo que se conoce como las máquinas de Kolmogorov-Uspensky. Las ideas esenciales que están detrás del análisis de Turing son (ver análisis en [15]):

1. Principio de localidad. La única información relevante para decidir qué hacer en el siguiente paso de un procedimiento es la información local que se tiene en ese momento (en las palabras de Turing, la región bajo "escaneo" de la máquina y el "estado" mental).

2. Principio de finitud. En cada momento, el estado (mente) es sólo capaz de almacenar y procesar un número finito de ítems (unidades atómicas de información).

Kolmogorov observa que hay otro supuesto en el análisis de Turing y es que la topología del espacio de información permanece invariable. Por lo tanto analiza la situación donde el espacio de trabajo es dinámico, esto es, la topología misma de la cinta cambia a medida que la máquina trabaja. Como toda investigación relevante, obtiene una serie de profundos resultados adicionales, pero no logra ir más allá de la máquina de Turing. Esto es, demuestra que este nuevo formalismo extendido de máquinas con espacio de trabajo dinámico tiene un poder expresivo igual a las máquinas de Turing [13].

El alumno y amigo de Turing, Robin Gandy, es quien probablemente ha llevado el análisis de la máquina de Turing más lejos. Propone la noción de *aparato mecánico discreto*, y axiomatiza la noción misma para luego estudiar (esta vez sí que formalmente, en contraposición a la tesis de Church) los límites y alcances de estos sistemas. El lógico alemán W. Sieg simplificó y mejoró esa idea haciéndola más entendible [12].

La historia no termina aquí. Más bien sólo está comenzando. En 1972, Gödel publicó una observación que denominó "Un error filosófico en el trabajo de Turing", donde expresa:

"Turing en su artículo de 1937 [el texto largo reproducido aquí al comienzo de la sección "*Finale:* Turing 1936"] presenta un argumento que se supone que muestra que los procedimientos/procesos mentales no pueden ir más allá de los procedimientos/procesos mecánicos. Sin embargo, este argumento es inconclusivo. Lo que Turing deja completamente de lado es el hecho que la mente, en su uso, no es estática, sino está desarrollándose constantemente, esto es, que nosotros entendemos más y más los términos abstractos a medida que los vamos usando, y que más y más términos abstractos entran en la esfera de nuestro entendimiento. Pudieran existir métodos sistemáticos que actualizan este desarrollo, los que pudieran formar parte de este procedimiento/proceso. Por lo tanto, aunque en cada etapa el número y precisión de los términos abstractos a nuestra disposición es finito, ambos (y, por consiguiente, también el número de estados indistinguibles de la mente que habla Turing) puede converger hacia el infinito en el curso de la aplicación de este procedimiento".

La profundidad de la observación de Gödel no se puede pasar por alto (aunque sea algo injusta con lo que Turing realmente quiere expresar). Lo que está indicando es la distorsión, la complejidad, los desafíos que introduce el tiempo, la dinámica al modelo clásico de Turing. De aquí surgen preguntas relevantes como las siguientes:

I. ¿Modela la máquina de Turing al individuo? Sí en un momento dado, fijo. No en su desarrollo existencial, en su desarrollo como persona. Gödel indica que falta un parámetro que es la dinámica de lo que en el modelo de Turing está fijo: el lenguaje, los símbolos, el conjunto de estados, etc.

II. ¿Modela la máquina de Turing el "espíritu" de la humanidad? No, contesta Gödel. Los argumentos son similares.

El debate que abrieron Gödel y Turing recién comienza. BITS

*Agradezco los comentarios de T. Carrasco, P. Alvarado, A. Martínez; el apoyo de P. Barceló, y al Proyecto Fondecyt 1110287.

Referencias

[1] R. Adams, An Early History of Recursive Functions and Computability. From Gödel to Turing. Docent Press, 2011.

[2] F. Cardone, J. R. Hindley, History of Lambda-calculus and Combinatory Logic. Swansea University Mathematics Department Research Report No. MRRS-05-06, 2006.

[3] M. Davis (Edit.) The Undecidable, Basic Papers on Undecidability Propositions, Unsolvable Problems and Computable Functions. Dover, 2004.

[4] J. W. Dawson Jr., Logical Dilemmas: The Life and Work of Kurt Gödel, A. K. Peters Ltd., 1997.

[5] K. Gödel, Collected Works, 3 volúmenes, 1986-1995. Oxford Univ. Press, 1995.

[6] K. Gödel, Obras Completas. Edición y traducción de J. Mosterín. Alianza Edit. 2006.

[7] C. Gutiérrez. "El Teorema de Gödel para no iniciados". Revista Cubo, Univ. La Frontera, Vol. 1, 1999 (dcc.uchile.cl/cgutierr/otros/godel.pdf).

[8] C. Gutiérrez, F. Gutiérrez, "Carlos Grandjot: tres décadas de matemáticas en Chile", Boletín Asoc. Mat. Venezolana, Vol. XI, No. 1 (2004), pp. 55 - 84.

[9] D. Hilbert,1926, "Über das Unendliche", Mathematische Annalen, 95: 161-90, 1926.

[10] A. Hodges, Alan Turing; the enigma. Vintage 1992.

[11] P. Odifreddi, Classical Recursion Theory. The Theory of Functions and Sets of Natural Numbers. Elsevier, 1989.

[12] W. Sieg, "Church without dogma: axioms for computability". New Computational Paradigms (B. Lowe, A. Sorbi, B. Cooper, editors); Springer Verlag 2008, 139-152.

[13] V. A. Uspensky, A. L. Semenov. Algorithms: Main Ideas and Applications. Kluwer Academic Publishers, 1993.

[14] H. Wang, Reflections on Kurt Gödel. MIT Press, 1987.

[15] H. Wang, Popular Lectures in Mathematical Logic. Dover Publ., 1993.