Ni computadoras ni sociedades de convivencia

Todavía no sabemos cómo construir una computadora cuántica. Pero sabemos que cuando se construya, podremos explotar las bizarras cualidades del mundo cuántico —superposición, paralelismo, indeterminación, colapso— para realizar proyectos de cálculo de una magnitud extraordinariamente mayor a todo lo que podemos hacer ahora con las computadoras convencionales. Por eso hay multitud de científicos e ingenieros averiguando cómo, cuando tengamos el aparato, lo programaremos para explotar de la mejor manera esas cualidades. Hace setenta años no teníamos computadoras —ni transistores, vaya, ni bulbos—, pero sí un puñado de científicos e ingenieros averiguando el equivalente a cómo inventar una computadora. Alan Turing destacó entre ese puñado de científicos e ingenieros y nos legó el arquetipo minimalista de la actividad de una computadora en una máquina imaginaria, la máquina de Turing. Hace setenta años, en un mundo sin computadoras y sin voto para las mujeres, Turing vivió su homosexualidad de la forma más abierta posible en Inglaterra, tolerado hasta un punto después del cual sufrió un proceso similar al que, cincuenta años antes, había acabado con Oscar Wilde. Su biografía, tremendamente moderna, reitera la idea de la increíble licuadora de usos y costumbres que fue el siglo XX.

Las matemáticas son un juego de jóvenes, dictaminó Hardy.[1] Después de su educación preuniversitaria en la típi-

[1] "Ningún matemático debería permitirse olvidar que las matemáticas, más que cualquier otra ciencia o arte es un juego para

ca escuela varonil del imperio tardío —donde las experiencias homosexuales eran comunes y comúnmente toleradas—, Alan Turing (nacido en 1912) entró al privilegiado mundo del King´s College en Cambridge —colegio "conocido por una actitud de tolerancia liberal"— a los 19 años. Tres años después había encontrado una segunda ruta al descubrimiento del teorema del límite central, lo que le valió un nombramiento de *fellow* del colegio y un estipendio de 300 libras esterlinas anuales (cerca de veinte mil dólares de ahora).[2] Ese resultado es un magnífico inicio pero, estrictamente, corresponde al de un matemático del montón. Su fama se debe a su segundo producto. En los primeros años de la década de los treinta se había asentado un poco el polvo levantado por la paradoja de Russell y se volvía a confiar en la posibilidad de establecer los cimientos de la matemática de forma completa —dentro de la matemática puede demostrarse la veracidad o falsedad de toda declaración—, consistente —no puede concluirse una declaración falsa a través de un proceso válido— y decidible —existe un método para demostrar la verdad o falsedad de toda declaración—. Dos personas se encargaron de garantizar que nunca se asentaría el polvo eliminando edificio y cimientos: Kurt Gödel demostró la imposibilidad de la completitud y de la consistencia, y Alan Turing la de la decidibilidad.

hombres jóvenes.... Galois murió a los veintiuno, Abel a los veintisiete, Ramanujan a los treinta y tres, Riemann a los cuarenta. Ha habido hombres que han hecho un gran trabajo mucho tiempo después; las memorias de Gauss sobre geometría diferencial se publicaron cuando tenía cincuenta años (aunque las ideas fundamentales las concibió diez años antes). No sé de ningún caso de un avance matemático de importancia que haya sido iniciado por un hombre de más de cincuenta ... Es muy difícil encontrar un ejemplo de un matemático de primera talla que haya abandonado las matemáticas y haya logrado distinguirse en primer plano en algún otro campo". *A Mathematician's Apology.* G. H. Hardy

[2] En 1934 una libra costaba 4.80 dólares (http://www.miketodd.net /encyc/dollhist.htm). O sea que las 300 eran unos 1350 dólares, o unos veinte mil de ahorita (http://www.measuringworth.com/ calculators/compare/result.php).

A todos los que no somos matemáticos nos cuesta mucho trabajo imaginarnos cómo se puede demostrar que no existe un método general para evaluar si una declaración cualquiera dentro de un sistema axiomático es verdadera o falsa. Pero sí podemos ver que si tal método existiera, podríamos hacer que una máquina lo aplique. El *Entscheidungsproblem* —¿cómo evitar poner esa palabrota?— tiene como consecuencia que los matemáticos se quedan sin chamba. La demostración de Turing es endiabladamente compleja, pero si es cierto que "todo hombre —o mujer— debe ser capaz de todas las ideas",[3] quizá valga la pena hacer una caricatura de ella con el fin de asomarnos al extraño mundo de los matemáticos. Además porque, en el camino hacia la demostración de la indecidibilidad, Turing inventó una máquina imaginaria, una computadora imaginaria, que, como precursora de las computadoras reales, modificó la realidad y determinó la forma que habría de tomar la modernidad más que ninguna idea filosófica, incluida la solución al *Entscheindungsproblem.*

Existe cierto orgullo matemático por la abstracción, que establece una jerarquía que pone en un lugar más alto el trabajo con funciones que el trabajo con números, por ejemplo. Pero, en las aplicaciones de las matemáticas —y veremos que a pesar de haber escalado las cimas más altas de la abstracción, el legado de Turing es esencialmente práctico— tarde o temprano hay que calcular, hay que computar. Para computar se necesita un método, un *algoritmo*, una receta que nos diga qué hay que hacer y cuándo se llega al resultado. Esos algoritmos pueden ser muy complicados, con muchísimas bifurcaciones que dependen del caso particular del que se trate, pero que, si son consistentes, tienen contemplados todos los posibles casos. Un matemático cabal se sentirá satisfecho después de haber encontrado el algoritmo y considerará el trabajo mismo de computar resultados como de segunda clase. Cuando empezamos a tener aplicaciones prácticas para las matemáticas, se creó el trabajo de computador(a)

[3] Borges, "Pierre Menard autor del Quijote".

y, aparentemente por las mismas razones de género que hacen que el índice de feminidad entre los estudiantes sobresalientes de la UNAM sea mayor que 100, la mayoría de las personas que realizaban ese trabajo eran mujeres —por eso entre angloparlantes se considera, al igual que a los barcos, del género femenino a las computadoras, y por eso habría que ponerle un hasta aquí a la costumbre española de España de llamarlas *ordenadores*, ¿no?—. Turing se pregunta ¿cuáles son las funciones mínimas que debe realizar una computadora para aplicar un algoritmo?[4]

La máquina de Turing consta de dos partes: una cinta dividida en cuadros, en cada uno de los cuales se puede escribir un solo símbolo, y una unidad lectora que puede leer o borrar el símbolo escrito en el cuadro, o escribir uno si el cuadro está en blanco, y mover la cinta un espacio en cualquier sentido. La máquina además está en cada momento en un estado —que Turing llama su *configuración m*— que determina su siguiente paso. En algún lado se debe especificar además una *tabla de conducta* que contiene la lista del configuraciones m de la máquina, es decir, el propio algoritmo.

No hay manera de entender el funcionamiento de una máquina con esta descripción formal, así que intentemos un ejemplo exageradamente sencillo.[5] Imaginemos que en la

[4] Y Leavitt nos regala una bonita imagen de la sustitución de computadoras por máquinas de Turing: "Hagamos ahora el brinco cinemático que constituye el corazón del artículo de Turing. De pronto desaparecen esos millones de mujeres que trabajan en la fábrica. Nuestras computadoras desaparecen, para, esperamos, dedicarse a una ocupación más reposada y estimulante. En su lugar, de pronto, está colocada una máquina de Turing. Una fábrica llena de máquinas de Turing, cada una de las cuales trabaja en un algoritmo específico. Permítasenos concentrarnos en la máquina que ha remplazado a nuestra computadora, que hacía sumas. Para simplificar las cosas —y ya que nuestra máquina acaba de empezar en este trabajo— démosle una tarea más simple que las que hacía anteriormente su contraparte humana. Pidámosle que sume 2 y 2". [p. 66]

[5] Este ejemplo aparece tal cual en el libro de Leavitt y, según refiere, en la biografía de Hodges, *Alan Turing: The Enigma* (Walker, Nueva York, 2000).

cinta que corre por la máquina, representamos un número entero como una sucesión de cuadros con un *uno* en cada uno. Por ejemplo, el pedazo de cinta que se representa debajo, muestra el número tres y el número cuatro, separados, en este caso, por tres espacios blancos. Nótese que se necesita cuando menos un espacio como separador, pero que la presencia de espacios adicionales no modifica el número.

			1	1	1				1	1	1	1		

El ejemplo junta dos números contiguos en una sola secuencia, es decir, *suma* dos números representados de esta manera. Para simplificarlo, supondremos que los dos números están separados solamente por un espacio en blanco.

			1	1	1		1	1	1	1				

Las instrucciones están mostradas en la tabla siguiente:[6]

Configuración m	Símbolo en la cinta	Acción de la máquina	Nueva configuración m
A	Ninguno	Desplaza un espacio a la derecha	A
A	1	Desplaza un espacio a la derecha	B
B	1	Desplaza un espacio a la derecha	B
B	Ninguno	Escribe 1, desplaza un espacio a la derecha	C
C	Ninguno	Desplaza un espacio a la derecha	C
C	1	Desplaza un espacio a la izquierda	D
D		Ninguna, la máquina se detiene	D
D		Borra el 1, la máquina se detiene	D

[6] Como tantos otros productos de la industria editorial estadounidense, el libro de Leavitt está pulcramente editado y es difícil encontrar una errata. Eso hace más sorprendente que casi todas las ecuaciones que aparecen en el libro estén grotescamente mal tipografiadas.

En el diagrama siguiente se muestran estados progresivos de la máquina, los cuadros sombreados son aquellos que recorre la lectora/escritora sin cambiar de configuración, la que se indica en el inicio de la cinta:

A			1	1	1		1	1	1	1		
B			1	1	1	1	1	1	1	1		
C			1	1	1	1	1	1	1	1		
D			1	1	1	1	1	1	1			

Nótese que la lectora lee todos y cada uno de los cuadros, pero la escritora sólo actúa dos veces, en la configuración B cuando escribe un cuarto uno sucesivo y en la configuración D cuando borra el octavo uno sucesivo. El resultado de este programita es la suma de las dos secuencias de números, 3+4=7.

La idea de este ejemplo simplísimo es justificar cómo una máquina con estos elementos es capaz de seguir una receta. El siguiente paso es demostrar que existen recetas para generar *secuencias*, listas infinitas de números que siguen una regla. Dos ejemplos sencillos son

0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

y

0	1	0	1	1	0	1	1	1	0	1	1	1	1	0

En seguida, Turing demuestra *que cualquier secuencia computable puede ser generada por una tabla de instrucciones,* una lista de operaciones similar a la que acabamos de analizar, la cual sirve para sumar dos números. La lista de las instrucciones está codificada, es decir, es una lista de símbolos que describen la acción de la máquina. Turing propone una modificación a esta codificación que emplee solamente números, así cada máquina de Turing se puede describir por una larga lista de dígitos, es decir como una secuencia computable. Llamaremos a esta secuencia el número descriptivo de la máquina. De acuerdo con ese código el número descriptivo 6170972567121067502159055525238695001415244844 1584753 corresponde a alguna máquina. No sabemos qué haga, pero podemos aplicar el código en reversa y escribir la tabla de instrucciones correspondiente. Desde luego, no todos los números descriptivos —secuencias computables— corresponden a una máquina que genere una secuencia computable, algunas corresponderán a máquinas inútiles que no generan ningún resultado, típicamente porque se quedan, como los robots de Asimov, atrapadas en ciclos infinitos de órdenes contradictorias.

¿Cómo distinguir los números descriptivos de las máquinas que generan secuencias computables de los de aquellas inútiles? Aquí aparece la propuesta *práctica* más importante del trabajo de Turing: pensemos en una máquina universal cuyo trabajo sea leer el número descriptivo de una máquina cualquiera, aplicar el código en reversa para obtener la tabla de instrucciones y aplicar esa tabla para producir el resultado que la máquina original produciría por sí sola. Esta máquina universal de Turing es a lo que aspiran las computadoras actuales, a las que se les puede alimentar una lista de instrucciones que describa la realización de algún algoritmo que es interpretado y reproducido. La computadora moderna se convierte momentáneamente en la máquina de Turing necesaria para resolver un problema dado.

Con respecto al *Entscheidungsproblem*, una vez que se cuenta con la máquina universal sólo hace falta un paso más. Ponderemos ahora la existencia de una máquina que pueda decidir si un número descriptivo cualquiera corresponde o no a una máquina útil. Llamemos a esta la máquina D. Alimentamos a D con el número descriptivo de otra máquina, la M por ejemplo, y D nos contesta con un cero si la máquina sirve para generar una secuencia computable y con un uno si no. ¿Todavía estás conmigo, estimada lectora? Ya nada más faltan dos pasos. Supongamos finalmente que podemos acoplar esta máquina *decididora,* D, con la máquina *universa*l, U. De tal manera que le podemos alimentar a este trabuco el número descriptivo de cualquier máquina, digamos la M. La máquina compuesta toma ese número y decide, primero, si corresponde a una máquina útil. Si es el caso, alimenta ese número a la máquina universal que lleva a cabo el algoritmo; pero, si no es el caso, la máquina decididora detiene el proceso porque no tiene caso alimentar a la máquina universal un número que no produce nada. Gracias a este mecanismo de control, la máquina UD es siempre una máquina útil: hace lo que M si M hace algo o nos informa que M no hace nada. Ahora viene el paso final: alimentemos con el número descriptivo correspondiente a la máquina UD a la propia máquina UD. ¡Catástrofe! Porque la parte decididora de la máquina decide primero que el número descriptivo de UD es el de una máquina útil y, por tanto, lo pasa a la parte universal. Pero la parte universal reproduce la máquina UD que primero decide que UD es válida y luego la pasa a U, que reproduce la máquina UD que primero decide que UD es válida y luego la pasa a U, que… Es decir, la máquina UD alimentada con el número descriptivo UD es inútil. Pero habíamos dicho que era útil. Y como UD no puede ser útil e inútil a la vez, D no puede existir.

Esta demostración de que no puede existir una máquina que decida si un número descriptivo corresponde a una máquina válida o no, equivale, después de unas cuantas piruetas matemáticas más, a que no hay algoritmo para decidir si una declaración matemática es falsa o verdadera. El *Entscheidungsproblem* se resuelve con un no rotundo.

Turing inició el largo proceso de publicación de sus resultados en 1936, a los 34 años. Como con tantos otros descubrimientos importantes, la apreciación cabal de sus méritos se fue estableciendo lentamente, no sin retrocesos o disputas de originalidad y paternidad de las ideas. Pronto Turing encontró que un colega, el profesor Church en Princeton, había llegado a la misma conclusión por un método alterno. Como Turing era el de menor jerarquía académica de los dos, tuvo que visitar a aquel en una estancia sabática. No fue esta visita la más exitosa desde el punto de vista académico y no constituyó el despegue de una carrera académica tradicional, pero sirvió para que Turing experimentara el ambiente mucho más igualitario[7] y mucho más tolerante con las preferencias sexuales. Esas ventajas no pesaron más que su nostalgia por la patria y pronto regresó a Inglaterra y a una situación que modificó su carrera definitivamente: en 1939 se unió al esfuerzo nacional de guerra.

¿Qué contribución podría tener un matemático que inventaba máquinas abstractas que analizan la utilidad de otras máquinas abstractas en la guerra contra el Reich? La vasta operación de la maquinaria de guerra moderna depende de la comunicación incesante de mensajes. Con la tecnología de la radio difundida igualmente entre las partes en conflicto, la única forma de proteger la información es la codificación. La codificación —y su desciframiento mediante la ruptura del código— son problemas matemáticos, tanto teóricos como prácticos y, en tiempos de guerra, extremadamente importantes. Turing encabezó la guerra contra las claves de codificación de los alemanes. Durante todo el tiempo que duró la guerra hubo una carrera entre los métodos y las máquinas de codificación de los alemanes —la célebre *Enigma*— y los métodos y las máquinas de descodificación de los aliados. Algoritmos y máquinas para llevarlos a cabo, la especialidad de Turing. Dice Leavitt,

> El alcance de su contribución al esfuerzo de la guerra —del que nunca habló en vida— no debe subestimarse, y si bien es probablemente una exageración decir que sin Turing los aliados no hubieran ganado la guerra, es razonable suponer que sin él hubiera tomado muchos años más ganarla. Al mismo tiempo, si las autoridades británicas hubieran sabido de la homosexualidad de Turing, muy probablemente le hubieran prohibido acercarse a Bletchley [el laboratorio de desciframiento], en cuyo caso, como dijera su amigo Jack Good, "hubiéramos podido perder la guerra" [p. 188].

[7] La mamá de Turing escribió, "la familiaridad de los comerciantes lo sorprendía; alguna vez mencionó el caso extremo del tintorero que, al explicarle cómo atendería un pedido especial de Alan, le puso una mano en un hombro. 'Esto hubiera sido impensable en Inglaterra".

Después de la guerra Turing trató de reiniciar su carrera académica sin obtener, ni remotamente, resultados como los que lo habrían de situar en el firmamento de los matemáticos. Combinó su trabajo teórico con proyectos prácticos de construcción de prototipos de computadoras. Pero la personalidad que hace falta para reunir fondos y lidiar con la realidad de las cosas y de las gentes es muy distinta de la que hace falta para lidiar con la realidad de la abstracción matemática, y Turing no tenía esa personalidad. El aislamiento propio del genio y del homosexual más o menos enclosetado aumentó con el tiempo. Su bienestar y su satisfacción con el mundo disminuyó —quizá, en parte, porque ya no estaba en la vorágine de resolver un problema central de las matemáticas ni de ganar la guerra—:

> El problema en parte, era la soledad. A pesar de la aceptación propia de su homosexualidad, que a menudo rozaba el orgullo, Turing nunca había experimentado una relación realmente satisfactoria con otro hombre. En su lugar, su vida erótica había consistido en ataques de amor irrefrenable, usualmente hacia hombres heterosexuales que no tenían ningún interés en él, alternando con ocasionales "amistades cariñosas" con otros homosexuales en los que tenía un mínimo interés sexual, y de quienes no estaba ni remotamente enamorado. [...] Más tarde declararía "a veces estás platicando cómodamente con alguien sabiendo que en tres cuartos de hora iniciarás una noche maravillosa o habrás sido corrido a patadas de la habitación" [p. 197].

Aceptó una plaza de profesor en la Universidad de Manchester. En esos años se convirtió en una figura pública en el contexto del debate acerca de la posibilidad de la inteligencia artificial. Su contribución más recordada a este debate es la prueba que propuso para demostrar la existencia de inteligencia artificial: pongamos a platicar a una persona con dos interlocutores, uno humano y uno artificial; si la persona no puede distinguir quién es quién, tenemos inteligencia artificial. Estos años de la posguerra hasta 1951 son los de la mayor normalidad en la vida de Turing: tenía trabajo, casa, vecinos, rutinas:

> Su vida erótica, si no floreciente, parece, al menos, un asunto menos depresivo de lo que había sido hasta entonces.

> [...V]iajaba a menudo a Europa —en una ocasión a Noruega y en varias a Francia—. El Código Napoleónico —que *no* criminalizaba el sexo entre hombres— significaba que en el "extranjero", los ingleses podían disfrutar un muy necesitado respiro del aura de preocupación y culpa que todavía se asociaba con el sexo homosexual en Inglaterra. El continente le daba a Turing la oportunidad de disfrutar andanzas eróticas sin temor a las repercusiones. Al buscar esas libertades, por unos pocos días a la vez, seguía el comportamiento típico de su generación [p. 265].

En Manchester, en 1952, ligó con un joven de 19 años en las afueras del cine. Arnold Murray no aceptó el dinero que le ofreció Turing, sino que lo tomó a escondidas de su cartera. En encuentros sucesivos le pidió dinero prestado. Finalmente, junto con un amigo, asaltó la casa de Turing. Todavía discutieron un posible arreglo, pero finalmente Turing hizo intervenir a la policía. Para su sorpresa, la policía presentó cargos contra Turing mismo: *magna indecencia con otro hombre.* El mismo delito que había acabado con Oscar Wilde cincuenta años atrás. Pero los tiempos *habían* cambiado. Turing no fue a la cárcel. Sólo fue condenado a un tratamiento con estrógenos que habría de "curarlo". El atleta, corredor de fondo, engordó y le crecieron los pechos.

Un día de 1954, la asistente doméstica encontró el cadáver de Alan Turing en su cama. Junto a él, había una manzana mordisqueada, impregnada con cianuro como se supo después. Su madre y algún amigo cercano trataron de descartar la hipótesis del suicidio.

En internet circula el rumor de que la manzana del logo de las computadoras Apple es un homenaje a Turing. La compañía lo niega, insiste en que está relacionado con Newton. Entonces, ¿por qué está mordida? •

Carlos Amador Bedolla

David Leavitt: *The man who knew too much: Alan Turing and the invention of the computer,* Atlas Books, W. W. Norton & Co., 2006