Miscelánea Matemática **56** (2013) 1–25 SMM

Cien años de Turing

Carlos Torres Alcaraz
Departamento de Matemáticas,
Facultad de Ciencias, UNAM
carlos.torres.0505@gmail.com

Introducción

En el documental *La extraña vida y muerte del Dr. Turing (The Strange Life and Death of Dr. Turing*, BBC TV science series "Horizon", 1992), Marvin Minsky se refiere a Alan Turing con las siguientes palabras:

> He aquí una persona que descubrió la cosa más importante en la lógica, inventó el concepto de computadora con programa almacenado, hizo cosas maravillosas en biología y en criptología, e inició la inteligencia artificial y corrió maratones, rodó bicicletas y tuvo terribles problemas relacionados con su sexualidad. A pesar de que se trata de la figura clave de nuestro siglo, yo no sé nada acerca de esta persona; no sé nada acerca de él y quisiera saberlo[1].

Minsky toca un punto significativo en el caso de Turing. Para él no solo se trata de recordar la obra de un hombre que jugó un importante papel en los asuntos del siglo veinte, sino de comprender la vida de alguien que fue avasallado por el lado más oscuro del sistema legal británico. "De lo contrario, nos dice, corremos el peligro de que la celebración misma se convierta en el foco de nuestra atención y el hombre se vea opacado por el brillo de su obra." Si la vida de Turing hubiera sido distinta, pensemos por ejemplo en la de su mentor Alonzo Church, heterosexual, casado, no convicto por algún delito y con una

[1]Marvin Minsky es cofundador del laboratorio de inteligencia artificial del Instituto Tecnológico de Massachusetts (MIT) y uno de los padres de la inteligencia artificial en los Estados Unidos. El documental se puede ver en Internet; hacerlo es altamente recomendable.

vida que se extendió hasta los 92 años sin grandes sobresaltos, quizá los festejos pudieran ceñirse a otros cauces. No obstante, dados los tintes dramáticos que encierra la vida de Turing, resulta imperioso mirar no solo a su obra sino a su persona, es decir, reconocer las características y diferencias individuales que lo distinguieron como tal. Los trágicos acontecimientos al final de su vida refuerzan esta idea. Al respecto, muchos de los entresijos que aspira a desentrañar Minsky se develan en el libro *Alan Turing, The Enigma of Intelligence* de Andrew Hodges, cuyo interés en la vida y obra de Turing deriva no solo de su vocación matemática, sino de su participación en el movimiento de liberación gay en los años setenta.

1. Vida de Turing

Wikipedia (registro "Alan Turing" en inglés, marzo de 2012) se refiere a Turing como una persona con una constitución robusta, voz aguda, ingenioso y un tanto pedante; como alguien que mostraba muchas de las características indicativas del síndrome de Asperger[2]. Esto último lo colocaría en el rango de quienes padecen trastornos autísticos, una cuestión difícil de esclarecer en la actualidad dada la carencia de datos clínicos y el hecho de que ya casi nadie queda de su generación. Respecto a su vida y obra, algunos datos son los siguientes:

> Alan Mathison Turing nació el 23 de junio de 1912 en la ciudad de Londres. Su padre, Julius Mathison Turing, era funcionario del gobierno colonial británico en la India, por lo que debía viajar constantemente entre dicho lugar y el Reino Unido en compañía de su esposa Ethel Sara. Debido a ello, durante su infancia Alan y su hermano John debieron pasar largos periodos de tiempo con amigos de sus padres en Inglaterra.
>
> A los 19 años Turing ingresó al King's College como estudiante de matemáticas, donde se graduó con honores en

[2]Con relación al síndrome o trastorno de Asperger el lector podrá consultar la ficha correspondiente en Wikipedia. Grosso modo, se trata de un conjunto de condiciones mentales que afectan la interacción social y la capacidad de comunicación del individuo. El sujeto afectado suele mostrar intereses en áreas que resultan ser muy restringidas y en muchos casos estereotípicas. A diferencia del trastorno autista, quienes padecen el trastorno de Asperger no observan retraso en el desarrollo del lenguaje, ni nada que afecte el desarrollo de la inteligencia. Muchas características del síndrome se hacen notorias en las fases tardías del desarrollo, cuando las habilidades de contacto social comienzan a desempeñar un importante papel en la vida del sujeto. No es tratable.

1934. En 1935 ingresó como profesor en dicha institución, siendo ahí donde escribió su famoso artículo "On computable numbers with an application to the entscheidungsproblem" (Turing, 1936) en el que introdujo las ahora llamadas "máquinas de Turing", en su afán por investigar qué se puede y qué no se puede computar. En 1936 recibió el premio Smith por su trabajo en la teoría de la probabilidad, donde descubrió una variante del teorema del límite central[3]. En septiembre de ese año se trasladó a la Universidad estadounidense de Princeton, donde trabajó con el lógico Alonso Church. Allí desarrolló como disertación doctoral un trabajo titulado "Systems of logic Based on Ordinals" (Turing, 1939) en el que explora, por decirlo de alguna manera, la otra cara de la moneda: ya no la mente que computa siguiendo una regla, sino las posibilidades de la mente cuando no lo hace. Aquí la cuestión era, simplemente, la de ver qué tan lejos se podría llegar en la construcción axiomática de la aritmética si la mente tuviera la capacidad de reconocer la verdad de las proposiciones indecidibles (cuya existencia había demostrado Gödel en 1931), no mediante una prueba, sino mediante la intuición u otros caminos. Estas cuestiones habrían de reaparecer años más tarde como parte del debate mentes-máquinas con la participación, entre otros, de Kurt Gödel, John Lucas, Roger Penrose y Robin Gandy.

Durante la segunda guerra mundial Turing trabajó para el gobierno británico en el Centro Nacional de Códigos (National Codes Center) de Bletchley Park, al norte de Londres. Ahí logró aclarar, junto con un equipo de trabajo, la configuración de la máquina Enigma utilizada por los alemanes para cifrar y descifrar mensajes. Si bien sus contribuciones

[3]Este trabajo fue la disertación de Turing para ser admitido como profesor miembro en el King's College de la Universidad de Cambridge en 1934. Al respecto, (Zabell, 1995) ofrece algunos detalles de interés: "En 1933 Turing asistió a una serie de conferencias ofrecidas por el distinguido astrofísico Sir Arthur Stanley Eddington sobre la metodología de la ciencia. Un tema que Eddington abordó fue la tendencia de las mediciones experimentales sujetas a errores de observación a tener una distribución normal o gaussiana. El bosquejo heurístico de Eddington no dejó satisfecho a Turing, por lo que este último se propuso ofrecer una prueba matemática rigurosa de lo que hoy se conoce como el teorema del límite central para variables aleatorias independientes (aunque no necesariamente con una misma distribución)." (traducción mía). Dicho trabajo jamás se publicó, pues Turing, para su sorpresa, pronto descubrió que el teorema ya había sido demostrado en 1922 por el matemático finlandés Jarl Waldemar Lindeberg.

tuvieron una enorme importancia, en aquel tiempo Turing no recibió ningún reconocimiento por ello, pues tales actividades se mantuvieron en secreto y no fueron reveladas sino hasta los años setenta. Ni siquiera sus amigos más íntimos sabían de ello en su momento. Hoy sabemos que su participación no se limitó a romper el código de la máquina Enigma, sino el de los teletipos FISH (fabricados por Lorenz Electric y Siemens & Halseke). Las ideas desarrolladas en este dominio ayudarían posteriormente al desarrollo de Colossus, la primera computadora programable electrónica digital construida en Dollis Hill en 1943 y diseñada por Max Newman. Justamente, esta máquina se construyó para descifrar los códigos Fisch transmitidos por la máquina Lorenz. En cuanto a Enigma, Turing diseñó la Bombe, una máquina electromecánica que permitía eliminar una gran cantidad de claves candidatas para este aparato. El procedimiento tenía una base lógica: para cada combinación posible se ponía en marcha una cadena de deducciones lógicas, de modo que cuando la cadena concluía en una contradicción, la combinación se desechaba. Con algunas modificaciones, estas fueron las principales herramientas utilizadas por los aliados para descifrar los mensajes codificados con las máquinas Enigma y Lorenz.

Al finalizar la guerra Turing acarició la idea de crear una computadora electrónica en el sentido moderno del término. No obstante, sus planes, apoyados por el National Physical Laboratory de Londres, fueron opacados por los proyectos estadounidenses que contaban con un mayor financiamiento. El desencanto lo llevó a refugiarse en el atletismo, especialmente en la prueba maratón, donde estuvo a punto de formar parte del equipo británico que compitió en las olimpiadas de 1948.

Ese mismo año Turing se desplazó a la Universidad de Manchester, donde en 1949 fue nombrado director delegado del laboratorio de computación. Ahí trabajó en la elaboración del software de una de las primeras computadoras reales, la Manchester Mark I. Al respecto, las motivaciones de Turing, más que industriales o comerciales, seguían siendo científicas, por lo que pronto hubo de volver a la cuestión de las limitaciones teóricas de la computación. En esta ocasión su atención se centró en el poder de las computadoras y de la mente humana ¿Habría alguna diferencia en sus alcances? Su argumento era que las computadoras, programadas en forma adecuada, podrían rivalizar con la mente humana. Este fue el inicio de lo que más tarde se conocería como *inteligencia artificial*. En 1950 Turing publicó su famoso ensayo "Computing Machinery and Intelligence" en el que propuso una prueba de inteligencia para las máquinas hoy conocida como "test de Turing". Esta prueba sigue siendo motivo de tentativas y debates en la actualidad. Al respecto, la postura de Turing se adivina claramente en su reacción ante lo que él llama *objeción teológica* a la tesis de que las máquinas pueden pensar: "El pensamiento, dirían sus opositores, es una prerrogativa del alma inmortal del hombre: Dios ha dado un alma inmortal a todos los hombres y las mujeres, pero a ningún otro animal o a las máquinas, por lo que ningún animal o máquina puede pensar". La escueta respuesta de Turing encierra la clave de sus ulteriores esfuerzos en este domino: "No puedo aceptar ninguna parte de esta objeción." (Turing, 1950).

En 1951 Turing fue electo miembro de la Royal Society por sus logros de 1936. Esto sucedía justo en una época en la que su interés se había desplazado a una nueva zona: en esta ocasión a la morfogénesis biológica. Fue entonces que publicó en 1952 un trabajo titulado "The Chemical Basis of Morphogenesis" (Turing, 1952) en el que sostiene que las formas de todo organismo dependen de un sistema de substancias químicas denominadas *morfogenos*, las cuales reaccionan entre sí y se difunden por los tejidos, determinando de esta manera el desarrollo de tales formas. Este mecanismo explicaría, por ejemplo, cómo se forman las manchas en las pieles de los animales (v. gr., las rayas del tigre o las motas del leopardo). Dicho ensayo se considera pionero en este dominio. Al respecto, en 2012 unos investigadores del King's College pudieron confirmar experimentalmente la teoría de Turing al aclarar cómo se forma el patrón de crestas en el paladar de los mamíferos (Economou et al, 2012). El esquema es el predicho por las ecuaciones de reacción-difusión de Turing. Al respecto, los trabajos posteriores de Turing en este campo no fueron publicados sino hasta 1992 en el libro *Collected Works of A. M. Turing* (ver bibliografía)

En febrero de 1952 el trabajo de Turing se vio interrum-

> pido por su arresto a causa de una aventura sexual con un joven desempleado de Manchester llamado Arnold Murray. Este joven había ayudado a un cómplice a entrar en la casa de Turing para robarle. Cuando Turing acudió a la policía a denunciar el delito, el delincuente se defendió diciendo que Turing sostenía relaciones homosexuales con Murray. Esto dio lugar a un proceso penal. A Turing se le acusó de *indecencia grave* (Gross Indecency) en una época en que la homosexualidad constituía un delito en el Reino Unido[4]. Convencido de que no tenía de qué ser exonerado, durante el juicio Turing no se defendió de los cargos y fue condenado. Como alternativa a la prisión se le dio la opción de someterse a un tratamiento con hormonas femeninas para reducir su libido (castración química). El procedimiento consistió en inyecciones de estrógenos durante un año, lo cual le produjo importantes alteraciones físicas como la aparición de pechos, aumento de peso e impotencia. En 1954, justo dieciséis días antes de cumplir los 42 años, Turing murió envenenado por cianuro. Una investigación determinó que se trataba de un suicidio, aunque su madre y otros creyeron que su muerte fue accidental.

En cuanto al absurdo delito de “indecencia grave”, el 10 de diciembre de 2009, tras una campaña por Internet, el primer ministro Gordon Brown pidió una disculpa pública a nombre del gobierno británico por el modo en que Turing fue tratado después de la guerra. Las siguientes son algunas de las palabras con que se refiriera a este penoso incidente:

> No es exagerado decir que sin su sobresaliente contribución, la historia de la Segunda Guerra Mundial pudiera haber sido diferente. ... La deuda de gratitud que tenemos con él hace todavía más horrible la manera tan inhumana en que fue tratado. ... Si bien Turing fue juzgado conforme a la ley de su tiempo y no podemos retrasar el reloj, el trato que se le dio fue totalmente injusto y tengo la oportunidad de decir cuánto lamento y lamentamos lo que le sucedió. Alan y los muchos miles de gays que, al igual que él, fueron condenados bajo leyes homofóbicas, fueron tratados de manera terrible.

[4]El fundamento de la acusación contra Turing fue una ley dictada en la época victoriana, conforme a la cual cualquier persona del sexo masculino que sostuviera relaciones sexuales con otro individuo de su mismo género incurría en el delito de indecencia grave (sic), pudiendo ser encarcelado por un periodo de hasta dos años.

La administración de Gordon Brown pidió una disculpa pública, pero no ofreció perdón alguno ni a Turing ni a los miles de hombres que aún siguen vivos y que fueron condenados por la misma razón. Esto ha sido motivo de escarnio en contra de un gobierno que se ciñe a una política (que no a una ley) de no cuestionar a las anteriores administraciones. La negativa se dio sin siquiera examinar lo que en realidad era la susodicha *indecencia grave*, una noción indefinida destinada a encubrir el control y la represión de actos homosexuales entre los hombres, como si se tratara de algo delincuencial.

Este no es el trato que merecía una persona cuya única falta fue la de no ajustar sus prácticas sexuales a los prejuicios sociales establecidos en la ley. No obstante, por encima de tan deplorables circunstancias lo que tenemos ante nosotros es una de las figuras más importantes del siglo veinte. Tal como lo señala la revista *Time*, en la época actual todo aquél que teclea una computadora, abre una hoja de cálculo o se sirve de un procesador de palabras, está trabajando con la encarnación de una máquina de Turing. Es por ello que en 2012 muchas organizaciones y académicos decidieron rendir homenaje a este hombre que con su obra no solo enriqueció nuestros pensamientos, sino que ayudó a establecer las bases del mundo moderno, cambiando con ello nuestra vida en todos los sentidos[5].

En lo que sigue nos concentraremos en algunos aspectos de la obra de Turing relacionados con la computación, la lógica matemática y la inteligencia artificial.

2. El problema de la decisión y las máquinas de Turing

Como sabemos, Turing ejerció una influencia decisiva en el desarrollo de la computación moderna al definir los conceptos de *algoritmo y función computable* con la introducción de lo que hoy se conoce como *máquinas de Turing*. Asimismo, sabemos que él fue el creador de la idea de

[5]Las acciones se cuentan por cientos. Ha habido grandes eventos en países como Alemania, Australia, Brasil, China, España, Estados Unidos, Filipinas, Israel, Italia, Nueva Zelanda, Noruega, Portugal, la República Checa y, por supuesto, Inglaterra. Lo anterior involucra a más de cincuenta instituciones en la organización de coloquios, conferencias, exposiciones, publicaciones, competencias, eventos artísticos (v. gr., música inspirada en Turing) y videos. Además, como un homenaje a su memoria se cuenta desde 1966 con el premio Turing que anualmente otorga la Association for Computing Machinery, el cual se reconoce como el más alto honor en el dominio de la informática, algo semejante a un premio Nobel.

máquina con programa almacenado, razón por la cual en la actualidad se le considera como uno de los padres de la computación. No obstante, no todos saben que Turing ideó las máquinas que llevan su nombre para resolver un problema teórico en torno a los fundamentos de las matemáticas, una cuestión que raya en lo filosófico. A éste se le conoce como *problema de la decisión*.

¿Cómo pudo ser que una idea introducida con fines teóricos tuviera a la larga tal importancia práctica? Quizá la respuesta se halla en un señalamiento hecho por Jack Copeland: para Turing no había distinción entre lo *puro y lo aplicado* (Copeland, 2004). Esto se advierte en su interés por resolver problemas procedentes de muy distintos frentes, desde la lógica pura y la teoría de grupos hasta la biología, desde la filosofía hasta el análisis de las características de las componentes electrónicas de un circuito eléctrico; en verdad, una postura diametralmente opuesta al estrecho enfoque que se ofrece en la investigación y en la educación hoy en día.

La historia del problema de la decisión inicia con David Hilbert, quien a comienzos del siglo XX impulsó un nuevo punto de vista en torno al trato que deberíamos dar a las teorías matemáticas. Una parte de su proyecto incluía la completa formalización de la matemática clásica, es decir, su reducción a un sistema de signos en el que el razonamiento matemático se substituye con un conjunto de reglas formales relativas al manejo de éstos. Si bien este propósito se vio frustrado de alguna manera por los teoremas limitativos de Gödel, en otro sentido aportó una de las ideas clave para la programación y el cómputo formal: la de procesos mecánicos que no toman en cuenta el significado de las expresiones involucradas, sino solo su forma. Como señala Gregory Chaitin, se trata de una pieza olvidada de la historia intelectual de la computación.

La idea central es la siguiente: una vez reducida una teoría matemática a un lenguaje simbólico artificial provisto de axiomas y reglas de deducción formal, nos podemos olvidar del significado de sus fórmulas y verla como un juego que se practica con marcas en el papel. El propósito del juego es "deducir" teoremas desde los axiomas aplicando las reglas. Y si bien las razones por las que practicamos las matemáticas se hallan en su significado, cuando lo que se busca es estudiar la demostración matemática y el modo en que se estructuran sus teorías, conviene recurrir a este juego con símbolos perfectamente delimitado.

Con relación a esta clase de sistemas, cobra sentido la pregunta acerca del alcance deductivo de los axiomas: ¿Es tal o cual fórmula deducible de los axiomas con base en las reglas del sistema? ¿Podemos decidir estas cuestiones de manera mecánica (algorítmica)? Por ejemplo, ¿es deducible la fórmula $0 = 1$ a partir de los axiomas de Peano formalizados? Dada una fórmula cerrada A (sin variables no cuantificadas) y su negación $\neg A$, ¿es siempre deducible alguna de las dos a partir de los axiomas? Cuando la respuesta a ésta última pregunta es afirmativa, se dice que el sistema es *sintácticamente completo.* ¿Habrá un sistema de este tipo para la aritmética de los números naturales? En tal caso la teoría podría resolver todo problema expresable en su lenguaje (en un lenguaje más técnico: toda teoría sintácticamente completa es decidible[6]). Hilbert diseñó el juego de tal modo que las reglas son de aplicación mecánica y se pueden seguir sin tomar en cuenta el significado de las expresiones. Son tan precisas que con base en ellas es posible establecer cosas como las siguientes: "en la línea 3 hay un error de sintaxis", o bien "la fórmula en la línea 6 no es ni un axioma ni se sigue de fórmulas anteriores por la aplicación de las reglas de inferencia". Tales conclusiones dan fin a toda controversia acerca de si una sucesión propuesta es o no una prueba: ya no más llamados a la subjetividad.

2.1. La noción de procedimiento mecánico

Si bien Hilbert se apoyó en la idea de *procedimiento mecánico*, ni él ni sus seguidores intentaron clarificar esta noción. Turing fue el primero en abordar esta cuestión desde su justa perspectiva. Cuando se habla de un *procedimiento mecánico*, el adjetivo atribuye al sustantivo la cualidad de poder ser llevado a cabo por una máquina. Pero, ¿qué es una *máquina*? Aquí la pregunta no es acerca de un conjunto de piezas o elementos móviles cuyo funcionamiento permite realizar un trabajo o transformar energía. Más bien, se trata de una noción matemática: la de un procedimiento que puede ser realizado de manera automatizada

[6]Las nociones de completud y decidibilidad son de suma importancia en la lógica matemática moderna. La primera de ellas se refiere al hecho de que la teoría prueba o refuta cualquier cuestión que se pueda expresar en su lenguaje (v. gr., la conjetura de Goldbach), mientras que la segunda se refiere a la existencia de un procedimiento mediante el cual se puede determinar si una fórmula es demostrable o no en el sistema, sin necesidad de producir una prueba de ella. Un ejemplo de lo anterior lo ofrece el cálculo proposicional, en el que el conjunto de teoremas coincide con el conjunto de fórmulas que son tautología. En tal caso, para saber si una fórmula es o no es teorema basta con elaborar su "tabla de verdad" y ver si en todos los casos el valor computado es 1 (verdadero $= 1$, falso $= 0$). Así, sin necesidad de disponer de una prueba, podemos saber cuándo una fórmula es demostrable en el sistema. Sin entrar en detalles diremos que la completud de un sistema implica su decidibilidad, pero no a la inversa (*v. gr.*, el cálculo monádico de predicados es un caso de teoría decidible pero no completa).

por un dispositivo de cómputo. La forma que le da Turing a esta noción es muy precisa. No se trata de objetos físicos. Las máquinas de Turing son dispositivos de cómputo teóricos, los cuales no están sujetos a ningún tipo de limitación física[7].

Acorde a lo que sabemos y entendemos acerca de estas cuestiones, estos toscos mecanismos son capaces de realizar cualquier cómputo que pueda llevar a cabo un ser humano (o, para el caso, cualquier computadora moderna). Al respecto, la cuestión era reducir la noción de cómputo a algo en verdad simple y el punto es que Turing lo logró. En (Wang, 1996, p. 232), Gödel se refiere a este logro de Turing con las siguientes palabras: "Antes de Turing nadie había percibido con nitidez el concepto de procedimiento mecánico. Fue él quien nos colocó en la justa perspectiva, y ahora podemos comprender dicho concepto con claridad." Al respecto, fue la incorporación de este concepto matemático preciso (el de "procedimiento mecánico", caracterizado a través de las *máquina de Turing*) lo que permitió establecer resultados de indecidibilidad con toda generalidad. Y fue Turing quien, apoyándose en los trabajos de Herbrand, Church y el mismo Gödel, llenó este vacío.

Con este recurso a la mano, Turing atacó el problema de la decisión o *Entscheidungsproblem* (en palabras de Hilbert, "el problema fundamental de la lógica matemática"): ¿Habrá un procedimiento mecánico que permita decidir si una fórmula del cálculo puro de predicados es válida? Basándonos en el teorema de completud de Gödel de 1930 (no confundirlo con sus teoremas de incompletud de 1931), podemos reformular esta pregunta de la siguiente manera: ¿habrá un procedimiento mecánico para decidir si una fórmula del cálculo puro de predicados es demostrable en el cálculo funcional K de Hilbert y Ackermann?

Dada la caracterización que hace Turing de lo que es un procedimiento mecánico, la pregunta anterior se convierte en una pregunta acerca de la existencia o no de una máquina capaz de decidir la cues-

[7]La idea que ofrece Turing de las máquinas es muy simple y se halla expuesta en muchos lugares (v. gr., en Wikipedia). Se trata de dispositivos que en teoría trabajan desplazándose sobre una cinta infinita dividida en una sucesión de celdas contiguas, cada una de las cuales tiene escrito un único símbolo o está en blanco. El dispositivo en sí consiste de una cabeza o ventana lectora/escritora que escudriña una celda a la vez, y de un programa de operación. En cada paso, la máquina "lee" el símbolo en la ventana y, con base en él y una condición interna denominada estado (de los cuales hay un número finito), realiza el siguiente paso en la operación. En cada paso las posibles acciones se limitan a tres cosas: borrar y escribir en la celda un nuevo símbolo, cambiar de estado y desplazar la ventana a una celda contigua. Así de simple, francamente trivial. Lo sorprendente es que todos los mecanismo de cómputo conocidos en la actualidad son, en esencia, reducibles a máquinas de este tipo.

tión (es decir, de una máquina que tomando como dato a la fórmula en cuestión, se detiene habiendo impreso la palabra **no** cuando la fórmula no es demostrable y la palabra **sí** cuando sí los es). La respuesta que halló fue significativa: no hay una máquina que responda correctamente con un **sí** o un **no** a la pregunta anterior en todos los casos.

La demostración de este hecho implicó tres cosas extraordinarias: a) la introducción de la noción de *máquina universal*, b) la resolución del *problema de la detención* y c) un *argumento diagonal* inspirado en la paradoja de Russell y el teorema de incompletud de Gödel.

2.2. El problema de la detención y la máquina universal

El problema es muy simple: dada la descripción de una máquina y un dato de entrada, decidir si la máquina se detendrá al ser ejecutada con esa entrada[8].

La respuesta a esta interrogante la halló Turing mediante la introducción de lo que él llamó *máquina universal*. Escuchemos a Martin Davis: "Turing nos mostró cómo producir una máquina individual que, por sí misma, puede hacer cualquier cosa que pudiera ser hecha por una máquina de Turing, un modelo matemático de una computadora orientada a todos los usos." *Grosso modo*, se trata de una máquina capaz de imitar el comportamiento de cualquier máquina M a través de la lectura de un código numérico para M. Esta idea, de uso común hoy en día (las computadoras digitales actuales son versiones prácticas de este concepto) fue algo novedoso en su momento. La idea misma de codificar las máquinas en los números naturales (es decir, de representar su programa de funcionamiento mediante números) fue tomada de Gödel, quien a través de un procedimiento semejante logra expresar las reglas de la aritmética formal en ella misma (logrando de este modo construir enunciados autorreferentes). Dada la gran cantidad de información que hay al respecto, en esta ocasión no entraremos en demasiados detalles, limitándonos a introducir algunas convenciones, un poco de notación y un simple ejemplo.

Con relación a las máquinas de Turing, los números naturales se suelen representar de manera muy simple: mediante sucesiones de trazos verticales (palotes) llamadas *numerales*. Por ejemplo, el numeral de 5

[8]Obviamente, si se impone un tiempo límite a la operación de la máquina, el problema es muy fácil de resolver. Lo que Turing mostró es que el problema se complica cuando no se impone ningún límite de tiempo, es decir, cuando se intenta determinar si la máquina se detendrá o no sin observar su comportamiento.

es la sucesión |||||. Así, la cinta de una máquina a la que se le dan como datos de entrada los números 3 y 5 se vería como en la figura 1.

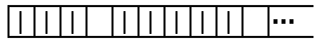

Figura 1.

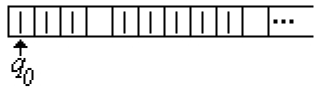

Figura 2.

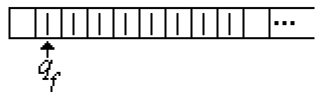

Figura 3.

Si la máquina, digamos M, está hecha para sumar números, al inicio y al final de la operación la cinta luciría como se indica en las figuras 2 y 3, respectivamente. En la figura 2 la flecha y la expresión q_0 indican la celda inspeccionada y el estado en que se halla la máquina en ese instante de operación. La convención es que q_0 siempre denota al estado inicial, es decir, el estado en que se halla la máquina al inicio del cómputo; asimismo, en la figura 3 q_f denota un estado para el que la máquina no cuenta con ninguna instrucción en combinación con el símbolo en la celda, (el símbolo "|" en este caso), lo cual equivale a la terminación del proceso (la máquina "se detiene"). Con cierta imprecisión, diremos que la palabra o expresión computada es simplemente aquella que se encuentra entre las primeras dos celdas en blanco al momento de detenerse ("||||||||" en este caso). Escribimos $M(3,5) = 8$. En general, cuando una máquina computa un numero k para los números $k_1, \ldots, k_n$ se escribe $M(k_1, \ldots, k_n) = k$.

Por ejemplo, si los únicos símbolos que podemos "escribir" en la cinta son el blanco "B" y el trazo "|" y las acciones "mover la ventana a la celda a la izquierda", "a la derecha" o "no moverla" las denotamos, respectivamente, con I, D, Λ, la siguiente es la descripción (función de transición) de una máquina de Turing M que computa a la función sucesor: $\delta(B, q_0) = (|, q_1, \Lambda)$; $\delta(|, q_0) = (|, q_0, D)$. La primera instrucción se lee "si la celda está en blanco y estás en estado q_0, pinta un trazo, cambia al estado q_1 y no te muevas"; la segunda en cambio se lee así: "si en la ventana se halla el símbolo "|" y estás en estado q_0, pinta un trazo (es decir, deja en la celda el símbolo que hay), permanece en el estado q_0 y muévete a la derecha". El lector imaginará cómo procede la máquina cuando se le da como dato un número cualquiera.

A fin de escribir un código para la máquina anterior, borremos en su descripción los símbolos "δ", "$=$" y los paréntesis. Lo que resulta es la siguiente sucesión: $B, q_0, |, q_1, \Lambda; |, q_0, |, q_0, D$. Como se ve, la descripción se basa en cinco símbolos especiales B (blanco), $|$ (trazo), I, D y Λ (movimientos de la ventana) y en los estados, de los que lo único relevante son los índices. Cambiemos ahora las expresiones $B, |, I, D$ y Λ por números, conforme a la siguiente correspondencia: $g(B) = 1$, $g(|) = 3$, $g(I) = 5$, $g(D) = 7$, $g(\Lambda) = 9$. Asimismo, a cada estado q_n le podemos asociar un número natural $g(q_n)$ cuya escritura solo incluye a los dígitos 0 y 2. Para ello, escribimos a n en base 2, y cambiamos en la expresión así obtenida todas las presencias del dígito 1 por el dígito 2. Ahora interpretamos la expresión obtenida de esta manera como el nombre de un número en base 10, y ese será el código $g(q_n)$ del estado q_n. V. gr., como $27 = (11011)_2$, entonces $g(q_{27}) = 22022$ (veintidós mil veintidós). Así, los estados quedan caracterizados por números cuya representación decimal solo incluye a los dígitos 0 y 2. Hagamos todo esto con la sucesión anterior: "$B, q_0, |, q_1, \Lambda; |, q_0, |, q_0, D$" $\to$ "$1, 0, 3, 2, 9; 3, 0, 3, 0, 7$". El *código* de la máquina es el número que resulta cuando eliminamos las comas en la expresión anterior: 1032930307 (mil treinta y dos millones, novecientos treinta mil trescientos siete). Nótese que la escritura de este número codifica (compendia) la descripción de la máquina, es decir, la clave de su funcionamiento, y se puede decodificar. V. gr., 1032730307121205323273201225 (del orden de los miles de tetrallones) se decodifica como la máquina $\delta(B, q_0) = (|, q_1, D)$; $\delta(|, q_0) = (|, q_0, D)$; $\delta(B, q_1) = (B, q_2, I)$; $\delta(|, q_1) = (|, q_1, D)$; $\delta(|, q_2) = (B, q_3, I)$, la cual computa la suma de dos números.

De esta manera, cada máquina M del tipo considerado tendría asignado un código i, el cual se puede usar como un índice para ella. Esto se indica con la notación M_i. Turing "construyó" entonces una máquina U que hace lo siguiente: $U(i, k_1, \ldots, k_n) = M_i(k_1, \ldots, k_n)$. En otras palabras, U aplicada a un índice i y una sucesión de números $k_1, \ldots, k_n$ computa lo mismo que la máquina M_i para los números $k_1, \ldots, k_n$. Turing la llamó "máquina universal".

Con estos recursos a la mano Turing imaginó la existencia de una máquina T (un *programa*) que siempre daría la respuesta correcta al problema de la detención (v. gr., computando un 1 o un 2). A ésta se le proporcionaría el código i de cualquier máquina M y los datos de entrada $x_1, \ldots, x_n$ y como resultado arrojaría una respuesta: "1" (equivalente a "Sí, la máquina M se detendrá al ser aplicada a los datos

$x_1, \ldots, x_n$"), o "2" (equivalente a "No, la máquina M nunca se detendrá al ser aplicada a los datos $x_1, \ldots, x_n$"). En seguida imaginó lo que sucedería: con base en T crearía, mediante una simple modificación, una segunda máquina T' que haría lo siguiente: si la M_i sometida a la prueba se detuviera tomando como dato su propio código i, T' entraría en un ciclo infinito y jamás se detendría. En notación moderna: Si $M_i(i) \downarrow$, entonces $T'(i) \uparrow$. De lo contrario, T' se detendría (al igual que T) para informar que M no se detendría. Este fue punto fino del argumento: ¿Qué pasaría si a T' se le proporcionara como dato su propio código k? Supongo que el lector ya habrá hallado la respuesta: Si T' aplicada a su propio código se detuviera, entonces T' aplicada a su propio código no se detendría (por definición de T'). Por tanto, T' no debería detenerse. Pero en tal caso T' debería detenerse para informar que T' aplicada a su propio código no se detendría. En símbolos: $T'(k) \downarrow \Leftrightarrow T'(k) \uparrow$, una contradicción que prueba la inexistencia de una máquina como la T propuesta.

El argumento anterior guarda una estrecha similitud con la paradoja de Russell, que, en el fondo, prueba la inexistencia de un conjunto cuyos únicos elementos son aquellos conjuntos que no se pertenecen a sí mismos. Y al igual que Russell, Turing se sirve de un argumento diagonal análogo al utilizado por Cantor en su demostración de la no numerabilidad del conjunto de los números reales.

2.3. El problema de la decisión

Turing dedujo de inmediato un corolario: si no hay un procedimiento para determinar de antemano (mediante un cálculo) si una máquina se detendrá, tampoco hay una manera de decidir la cuestión axiomáticamente ¿Por qué? Porque si se pudiera utilizar un sistema axiomático de esta manera, con base en él podríamos desarrollar un algoritmo que resolviera el problema de la detención y, por consiguiente, construir una máquina T como la ya referida. Producto de lo anterior Turing llegó a la misma conclusión que Gödel en sus investigaciones en torno a los fundamentos de la teoría de los números (en el sentido de que ninguna teoría axiomática que contenga a la aritmética puede ser consistente y completa a la vez) y a algo más: El problema de la decisión no se puede resolver de la manera en que Hilbert pretende. En efecto, si el cálculo funcional K de Hilbert y Ackermann (o cualquier otro) fuera decidible, el problema de la detención sería resoluble axiomáticamente, pues en K se pueden construir fórmulas que describen el funcionamiento de las máquinas de Turing. De hecho, así lo demuestra, para cada máquina M

hay una fórmula φ_M con la siguiente propiedad: si hubiera un método para demostrar que φ_M es demostrable o no, entonces habría un procedimiento para determinar si la máquina M eventualmente se detendría habiendo computado un cierto símbolo s, lo cual entraría en conflicto con la indecidibilidad del problema de la detención.

Todo esto lo hizo Turing en su trabajo "On computable numbers with an application to the entscheidungsproblem" de 1936, en el que el énfasis se pone en ciertas cuestiones teóricas y filosóficas, antes que en las cuestiones prácticas relativas a la computabilidad. Valga la insistencia: pensar que Turing escribió su trabajo como una mera anticipación de las computadoras modernas equivale a perder de vista el punto central de su trabajo de 1936.

No obstante, aunque de momento el interés de Turing no estuviera centrado en cuestiones prácticas, en su obra anticipó de un solo golpe:

a) los programas de interpretación, en los que un programa general decodifica a otros programas y los implementa, es decir, los imita;

b) las computadoras con programas almacenados, con lo que la distinción entre datos y programas se borra (en $U(i, k)$, el dato i equivale a un programa, es un programa), y

c) la versatilidad de las computadoras actuales, donde las soluciones de hardware se substituyen con el software equivalente.

Desde la perspectiva de las matemáticas puras, Turing inventó las computadoras. Su máquina universal suministra un modelo básico y preciso de cómo trabajan las computadoras actuales. Esto no significa que la arquitectura de éstas corresponda al diseño de las máquinas de Turing. En este sentido, escribir un programa para la máquina U e implementarla en diversos contextos lo único que produciría sería un dispositivo sumamente ineficiente como para que pudiera tener un uso práctico. Al respecto, Turing no ofrece en su ensayo ninguna idea de cómo poner en práctica las ideas que sugiere, dejando al futuro tal posibilidad.

En cuanto a la filosofía, la incomputabilidad de Turing se asemeja a la impredecibilidad cuántica, en el sentido de que ha permitido cuestionar viejas certezas como la de que todo problema matemático es resoluble. Al respecto, la máquina universal (el gran aporte de Turing a las ciencias de la computación) asoma como un mero subproducto en esta batalla en contra de previsibilidad laplaciana: en el dominio de lo computable hay certezas que no podemos alcanzar.

3. Los teoremas de Gödel y extensiones de la aritmética formal

En 1938 Turing dio un giro en sus investigaciones, orientándolas hacia los teoremas de Gödel. Fue entonces que escribió el ya mencionado trabajo “Sistems of logic based on ordinals” en el que considera el método de Gödel para construir enunciados indecidibles. El tema es el siguiente: Si la mente humana puede reconocer la corrección de las proposiciones gödelianas (las cuales, por otra parte, son indemostrables en el sistema original), al hacerlo estará yendo más allá de las reglas y los axiomas en que se basa el sistema. Dicha capacidad le permitiría crear sistemas cada vez más poderosos mediante la adición de tales proposiciones indecidibles como axiomas, dando lugar a un serie de “lógicas ordinales” (es decir, una serie de sistemas formales que se formarían siguiendo la sucesión ordinal) con la que se intentaría minimizar el efecto del teorema de Gödel. Así, Turing compone una familia de lógicas ordinales L_α, una para cada ordinal numerable α, que se forma añadiendo a los axiomas de L_α una fórmula indecidible para L_α (v. gr., el enunciado de Gödel G_α). Cuando α es un ordinal límite, L_α es la unión de las L_β para $\beta < \alpha$. Este procedimiento abre la posibilidad de hallar un sistema completo sin faltar al primer teorema de Gödel, pues L_α puede no tener una axiomática finita cuando α es un ordinal límite. Turing prueba que el proceso de completar la teoría debe terminar para algún ordinal $\alpha < \omega_1$. Esto lo llevó a explorar la teoría de ordinales construibles y a demostrar que el problema de decidir si una fórmula del cálculo λ de Church representa un ordinal es irresoluble. A la vez, muestra cómo se podría escapar de las limitaciones que establece Gödel mediante ciertas máquinas denominadas “máquinas con oráculo”. Si bien Turing no llega a ninguna conclusión respecto a si la mente tiene o no tal capacidad, la sola elaboración de este trabajo es una muestra de su versatilidad de pensamiento. Al final de la siguiente sección habremos de volver a este punto desde otra perspectiva. Mientras tanto, damos por terminado todo comentario al respecto en virtud de que en este mismo número de *Miscelánea matemática* figura un artículo en el que se aborda este tema en detalle.

4. La imitación de la mente

Durante su estancia en Bletchley Park Turing desarrolló algunas ideas acerca de la posibilidad de que las máquinas mostraran un comportamiento inteligente. Tales ideas las solía ilustrar refiriéndose al ajedrez. A la sazón el reto que enfrentaba era introducir métodos heurísticos para que las máquinas lograran un buen desempeño en este juego, pues el uso de la "fuerza bruta" resultaba imposible dado el descomunal número de variantes por analizar. Era imprescindible reducir lo que hoy se conoce como "espacio de búsqueda". La primera vez que Turing abordó en público la idea de la inteligencia artificial fue en una conferencia dictada en Londres en 1947, donde señaló la posibilidad de una máquina que aprendiera de la experiencia y pudiera alterar sus propias instrucciones. En 1948 Turing lanzó un reto a su amigo Donald Michie para ver quién escribiría primero un algoritmo que jugara al ajedrez[9]. Desde entonces este juego se ha considerado como un importante terreno de pruebas en la inteligencia artificial (IA). Ese mismo año escribió, aunque no publicó, lo que se puede considerar el primer manifiesto de la IA. Se trata de un reporte titulado "Intelligent Machinery" en el que introduce muchas ideas que más tarde serían centrales en esta disciplina como, por ejemplo, la de "entrenar" a una red neuronal para que desempeñe ciertas tareas específicas[10].

En 1950 Turing escribió un programa para computadora que jugaba al ajedrez (tal vez el primero en la historia) y propuso su famoso *test* de inteligencia. Su convicción era que con el tiempo se podría programar una máquina capaz de adquirir habilidades que rivalizaran con las de un ser humano en lo que al pensamiento se refiere. La idea es clara en la consideración misma del ajedrez: Si se programara una máquina para participar con cierta destreza en este juego, un humano que ignorara que su rival es un autómata no podría diferenciarlo de una persona. En tal caso la máquina competiría con los humanos en inteligencia, y se podría decir que en este dominio imita su comportamiento. El *test* de Turing nació al extender esta cuestión a otros dominios: ¿Sería posible diseñar una máquina que pudiera no solo jugar o resolver problemas, sino responder a las preguntas que se le hicieran de tal modo que éstas no se pudieran diferenciar de las de un ser humano? Aún más, ¿Sería

[9]Michie recuerda que Turing solía experimentar con dos estrategias que posteriormente se conocerían en la inteligencia artificial como minimax y best-first. La estrategia minimax se aplica en juegos con adversario y de información perfecta; ésta consiste en desarrollar un algoritmo recursivo que permita minimizar la máxima pérdida. La idea subyacente es elegir el mejor movimiento suponiendo que el adversario escogerá el peor para ti. En cambio la estrategia best-first consiste en explorar una gráfica (en este caso el árbol de continuaciones posibles) expandiendo primero el nodo más promisorio conforme a un criterio predeterminado (v. gr., movilidad de las piezas).

[10]Turing, 1948.

posible construir un máquina que tuviera sentimientos como los de una persona?

4.1. El test de Turing

El *test* de Turing se relaciona con la primera de estas interrogantes y tiene como base un antiguo juego de salón conocido como *juego de la imitación*. En éste intervienen tres participantes: Un hombre (A), una mujer (B) y un interrogador (C), cada uno de los cuales se halla en una habitación por separado. El interrogador se puede comunicar con A y B a través de textos escritos (v. gr., a través de un teletipo), y les puede preguntar cualquier cosa; los conoce como X e Y. Puede escribir, por ejemplo, "que X diga cuán largo tiene el pelo." El objetivo del juego para el interrogador es determinar, con base en las respuestas recibidas, quién es el hombre y quién la mujer. Al final del juego él debe decir "X es A y Y es B" o "X es B y Y es A". El propósito de B es ayudar a C a que haga la identificación correcta, mientras que el de A es que el interrogador haga la identificación incorrecta, haciéndole creer que él es la mujer. El test de Turing resulta cuando una máquina ocupa el lugar de A y un ser humano (no necesariamente una mujer) el lugar de B. La pregunta es, ¿habrá una máquina que induzca al interrogador a hacer la identificación incorrecta con la misma frecuencia que cuando el juego es practicado por un hombre y una mujer? Turing sostiene que si tal fuera el caso, ello constituiría una clara evidencia en favor de la tesis de que las máquinas pueden pensar.

4.2. El test en la actualidad

Desde su nacimiento, el *test* de Turing se estableció como un verdadero reto para la IA, como una herramienta para medir el avance de esta disciplina. Al respecto, Turing confiaba en que a la larga el objetivo de la simulación sería alcanzado. Todo dependería del desarrollo de la tecnología y del arte de la programación. Dice al respecto: "Creo que dentro de unos cincuenta años será posible programar computadoras, con un potencial de almacenamiento de unos 10^9, capaces de jugar el juego de la imitación suficientemente bien como para que el interrogador no tenga más de un setenta por ciento de posibilidades de hacer la identificación correcta después de cinco minutos de interrogatorio." (Turing, 1950). Esta proyección dio lugar a que en 1991 el empresario Hugh Loebner, en combinación con el Cambridge Center for Behavioral Studies, estableciera el llamado premio Loebner. Éste consiste en una competencia anual con un pago de 2000 dólares al programa que más se asemeje a un ser humano, y un premio de 100,000 dólares a la primera computadora cuyas respuestas sean indistinguibles de las de un ser humano[11].

A la fecha los resultados no han sido los esperados cuando la plantilla de interrogadores está integrada por gente versada en el tema. Una cuestión aún bajo discusión es la del tipo de preguntas que se pueden hacer. Al respecto, en la competencia no se han impuesto normas precisas, pudiéndose decir que el propósito es que el programa engañe al interrogador en el más amplio sentido de la palabra, es decir, sometiéndose a cualquier clase de preguntas. Esto trae a colación el problema de cuáles serían los términos razonables del diálogo, pues lo que Turing propuso en un principio era que la máquina mostrara un comportamiento inteligente, no un conocimiento de las costumbres humanas o de los usos del lenguaje. Por ejemplo, en la competencia de 2008 tres interrogadores familiarizados con el tema pudieron diferenciar rápidamente a las máquinas con preguntas como "Si nos damos la mano al saludarnos, ¿qué mano estoy sujetando?" (la cual tiene que ver con hábitos e interacciones corporales, no con la inteligencia), "¿A qué huele el color rojo?" (la cual es una pregunta capciosa), o "Si tengo un alhajero en la mano, ¿Cuántos CD puedo guardar en él?". Aún así, hay un rango de preguntas que moran en la zona de lo admisible y que siguen causando problemas a esta clase de programas. Por ejemplo, "Las cuatro capitales del Reino Unido son tres: Londres y Manchester. ¿Qué hay de erróneo en esta afirmación?", o bien, "Si Londres está al sur de Oxford, ¿Oxford está al norte de Londres?". Estas interrogantes, utilizadas en el test, atañen al sentido de expresiones como "dos" y "tres" o a cierto tipo de relaciones (que no experiencias) espaciales no asociadas a hábitos o costumbres.

En un reporte, los ya referidos participantes señalan algunas estrategias útiles para el examinador, y por lo tanto algunos de los escollos que las máquinas deberán superar (Floridi 2009). Un interrogador no experimentado suele hacer preguntas inútiles como, por ejemplo, "¿Eres una computadora?" o "¿Crees en Dios?", con lo que omiten el punto central del interrogatorio: determinar lo más pronto posible si se trata de un programa o no. Las respuestas deberían ser lo más informativas posible, lo cual significa que uno debería obtener con cada una de

---

[11] La competencia se organiza bajo la siguiente modalidad: para cada máquina, un grupo de personas se van turnando en el lugar de (B) y otro grupo de personas (el de los jueces o interrogadores) en el de (C). La idea es que si un número suficiente de estas últimas son incapaces de diferenciar a la computadora de los seres humanos que participan en la prueba se concluirá, conforme a las expectativas de Turing, que la computadora es una entidad pensante, inteligente.

ellas un máximo de evidencias. V. gr., con relación a la primera de las preguntas anteriores, un "no" nos dejará absolutamente ignorantes en torno a quién es nuestro interlocutor, y algo similar sucede con la segunda. Otro consejo es que las preguntas signifiquen un reto para la máquina sintáctica al otro lado del teletipo. No sirve preguntar cosas como "¿Qué tal, cómo te va?", sino cosas que requieren comprensión por parte del interlocutor, es decir, que éste se vea obligado a entender la pregunta, su contexto e implicaciones.

Lo anterior quizá explica el éxito relativo que han obtenido algunas máquinas. Por ejemplo, un programa denominado Cleverbot (véase la entrada con este nombre en Wikipedia) fue puesto a prueba en un festival celebrado en Guwahati, India, en septiembre de 2011. En la prueba participaron un total de treinta voluntarios, los cuales conversaron durante cuatro minutos con una entidad desconocida. La mitad de los voluntarios lo hizo con humanos, mientras que la otra mitad lo hizo con Cleverbot, una aplicación web IV creada por el británico Rollo Carpenter. El 59.3 % de las personas creyeron que Cleverbot era un humano. Esto resulta sorprendente si tomamos en cuenta que el 36.7 % de los otros participantes creyeron que su interlocutor humano era una máquina. Para Carpenter, estos logros aún no llevan a considerar que la máquina piensa; más bien, en su opinión lo correcto es decir que alguna inteligencia está involucrada en el proceso, si bien habría que diferenciar esta noción de lo que es pensar[12]. De hecho, considera que pasar el test no prueba inteligencia; solo prueba que la máquina puede imitar inteligencia (lo cual, por cierto, no significa que los intentos por pasar la prueba no tengan sentido).

Un punto fino en este debate es, ¿cómo aprenden los programas imitadores? Cleverbot lo hace de las conversaciones, mientras que los humanos contamos con una multitud de datos sensoriales enriquecedores[13]. Así, sostiene Carpenter, lo justo sería un robot con las mismas capacidades sensibles que los humanos (vista, tacto, etc.), junto con una buena estrategia de aprendizaje.

[12]Ver `http://www.newscientist.com/article/dn20865-software-tricks\people-into-thinking-it-is-human.html`

[13]Cleverbot conversa consultando una base de datos que contiene todas sus conversaciones anteriores, seleccionando una respuesta apropiada a la pregunta o comentario. La versión en línea ejecuta dicha búsqueda tres veces antes de decidir una respuesta, mientras que la versión utilizada en el test de Turing lo hace 42 veces. La eficiencia de este procedimiento se torna evidente en la siguiente conversación sostenida en línea entre Cleverbot y la revista New Scientist: NS: Do you think you have passed the Turing test? C: What is the Turing test? NS: It's a test to determine if a machine is intelligent. C: Have you taken the test? NS: No, but you have, do you think you passed? C: I am, but I'm too lazy to do anything with it.

Como se ve, la cuestión de si la mente humana se puede o no simular es algo que habrá de decidirse en la práctica, con la construcción de programas que lo hagan bien en distintos frentes (incluyendo el test de Turing). Aún así, esto no sería una prueba irrebatible de que la mente humana es equivalente a una máquina, pues podría suceder que ni la inteligencia artificial ni la fisiología lograran reunir los elementos necesarios para demostrar un hecho de tal naturaleza. En tal caso lo único que se tendría como prueba sería la evidencia empírica. Lo que sí podemos decir es que la labor de Turing en este dominio refuerza lo dicho anteriormente en este trabajo: Para él, la línea divisoria entre la teoría y la práctica es muy tenue. Los problemas que plantea Turing tienen a la vez un enorme sentido teórico (e incluso filosófico, v. gr., ¿funciona la mente como una máquina?) y un sentido práctico, (v. gr., elaboración de programas inteligentes, programas que aprenden, redes neuronales, etc.), marcando líneas de trabajo muy precisas en ambos frentes.

4.3. Algunas observaciones

Cerramos esta sección con algunos comentarios en torno a la obra de Turing en el dominio de la lógica y la IA. Los comentarios tiene como base un ensayo de Andrew Hodges titulado “Turing and the Turing Test”[14].

1) Las máquinas de Turing también están basadas en el principio de imitación. La diferencia es que en 1936 lo que Turing estaba ideando era una máquina que pudiera computar lo mismo que un humano cuando sigue un método bien definido y cuenta con el tiempo y los recursos necesarios (lápiz y papel, etc.). En este sentido, las celdas de la cinta corresponden a la cuadrícula de un cuaderno escolar, y las operaciones básicas de inspeccionar una celda, marcar, borrar y desplazarse a la izquierda o a la derecha se relacionan de una manera muy obvia con las acciones de un calculista. Asimismo, la finitud de cada configuración sobre la cinta (solo un número finito de celdas no en blanco) corresponde a la memoria finita de los humanos, y el número finito de estados en que se puede hallar una máquina se relaciona con el número finito de estados mentales relevantes para el cómputo (“ahora voy a sumar estos números”, “ahora los voy a multiplicar”). Esta idea

[14] El ensayo se puede leer en `http://www.turing.org.uk/publications/testbook.html`

de modelar los "estados mentales" mediante "estados de máquina" pareciera anticipar la tesis de 1950 de que el comportamiento de la mente también se puede modelar cuando ésta no sigue ninguna regla. Al menos, eso es lo que sostiene Hodges con sólidos argumentos.

2) Ciertamente, Turing estaba consciente de la objeción de que las máquinas no pueden hacer nada que no sea mecánico. Por ejemplo, aquello que requiere iniciativa, imaginación, juicio, comisión de errores, etc. No obstante, consideraba que las máquinas serían capaces de obtener resultados similares a los humanos simulando todas esas cosas no mecánicas. En otras palabras: Turing adoptó la idea de que las operaciones computables podrían abarcar todo lo que normalmente se llama "pensamiento", y no solo los métodos bien definidos, es decir, las máquinas se podrían programar de tal modo que sus resultados parecieran a los ojos de cualquier observador como algo creado con iniciativa, imaginación, etc. por un ser inteligente[15].

3) Lo anterior guarda un vínculo con lo hecho por Turing en Princeton entre 1936 y 1938. Ahí lo que investiga es la lógica de lo no computable. En (Turing, 1939), Turing se refiere a "obtener [un sistema] más completo mediante la adición de fórmulas [los enunciados indecidibles de Gödel] como axiomas, las cuales son vistas como correctas, aunque el teorema de Gödel muestre que son indemostrables en el sistema original". Con esto contempla la posibilidad de que la mente tenga el poder intuitivo de dar pasos no computables que quedan fuera del alcance de las máquinas creadas por ella (una tesis sostenida por Gödel). No obstante, el trabajo en Bletchley Park durante la Guerra le llevó a desechar este punto de vista, adoptando entonces la tesis de que las operaciones computables serían suficientes para dar cuenta de las operaciones mentales de apariencia no mecánica, incluyendo aquellas relacionadas con el reconocimiento de la verdad.

Lo anterior no significa que para Turing la estructura del cerebro debería parecerse a la de una máquina. Las soluciones que él propone están dirigidas a la programación de máquinas, no a la construcción de algo así como un cerebro.

[15]En (Gandy, 1996) El lector hallará una clara defensa de esta posibilidad sustentada por Robin Gandy, discípulo, amigo y seguidor de Turing.

5. El final

Por desgracia, la obra de Turing llegó a su fin a temprana edad. Su última publicación fue un artículo en la revista *Science News* en 1954. El trabajo, titulado "Solvable and unsolvable problems", trata con la cuestión de la no computabilidad en un sentido puramente matemático. Entre otras cosas, Turing ofrece una prueba original del teorema de incompletud de Gödel con base en la cuestión de la irresolubilidad de ciertos problemas matemáticos. En el último párrafo señala que los teoremas de Gödel (así como muchos otros resultados negativos acerca de la resolubilidad) establecen un límite a lo que podemos alcanzar mediante el razonamiento puro, pudiéndoseles considerar como una demostración, dentro de la matemática misma, de la inadecuación de la "razón" cuando no se le apoya con el sentido común. Esta última afirmación pareciera ir en contra de lo sugerido en el artículo de 1950, pues nada dice acerca de la posibilidad de que las máquinas pudieran simular tal sentido interno. Esto se puede considerar como un indicativo de que, quizás, Turing estaba reconsiderando su postura en cuanto a la inteligencia de las máquinas y el lugar que ocupa el sentido común en el trabajo matemático. No obstante, esto nunca lo sabremos, como tampoco sabremos en qué dirección se habría movido su pensamiento en caso de haber vivido más tiempo.

Bibliografía

1. C. T. Alcaraz, La lógica matemática en el siglo XX, *Miscelánea Matemática* **31** (2000) 61–105.
2. G. J. Chaitin, Computers, paradoxes and the foundations of mathematics, *American Scientist* **90** (2002) 164–171.
3. B. J. Copeland, The Turing test, *Minds Mach.* **10** (2000) 519–539.
4. ______, *B. Jack Copeland (ed), The Essential Turing*, Oxford University Press, Oxford, 2004.
5. A. D. Economou, A. Ohazama, T. Porntaveetus, P. T. Sharpe, S. Kondo, M. A. Basson, A. Gritli-Linde, M. T. Cobourne, y J. B. A. Green, Periodic stripe formation by a Turing mechanism operating at growth zones in the mammalian palate, *Nature Genetics* **44** (2012) 348–351.
6. L. Floridi, M. Taddeo, y M. Turilli, Turing's imitation game: Still an impossible challenge for all machines and some judges –an evaluation of the 2008 loebner contest, *Minds & Machines* **19** (2009) 145–150.
7. R. Gandy, *Human versus Mechanical Intelligence*, 125–136, Millican y Clark, 1996.
8. K. Gödel, Publications 1929-1936, en *Collected Works of Kurt Gödel*,

S. Feferman, J. W. D. Jr., G. H. Moore, G. H. Moore, R. M. Solovay, y J. van Heijenoort, eds., Nueva York y Oxford, Oxford University Press, 1986.

9. ______, Publications 1938-1947, en *Collected Works of Kurt Gödel*, S. Feferman, J. W. D. Jr., G. H. Moore, G. H. Moore, R. M. Solovay, y J. van Heijenoort, eds., Nueva York y Oxford, Oxford University Press, 1990.
10. ______, Unpublished essays and lectures, en *Collected Works of Kurt Gödel*, S. Feferman, J. W. Dawson, W. G. Jr., C. Parsons, y R. M. Solovay, eds., Nueva York y Oxford, Oxford University Press, 1998.
11. A. Hodges, *Alan Turing, the Enigma of Intelligence*, Unwin Paperbacks, London, 1983.
12. Loebner price homepage, `http://www.loebner.net/Prizef/loebner-prize.html`.
13. P. Millican y A. Clark, *Machines and Thought, The Legacy of Alan Turing*, Oxford, Universidad de Oxford, 1996.
14. E. Nagel y J. R. Newman, Gödel's proof, `http://es.scribd.com/doc/17462870/Ernest-Nagel-and-James-R-Newman-Godels-Proof`.
15. The Alan Turing internet scrapbook, `www.turing.org.uk/turing/scrapbook/machine.html`.
16. A. M. Turing, *On the Gaussian error function, Unpublished Fellowship Dissertation*, King's College Library, Cambridge, 1934.
17. ______, On computable numbers with an application to the entscheidungsproblem, en *Proceedings of the London Mathematical Society, ser. 2*, tomo 42, 1936, corrections, Ibid, vol. 43, pp. 544-546, 230–265.
18. ______, Systems of logic based on ordinals, en *Proceedings of the London Mathematical Society, ser. 2*, tomo 45, 1939, 161–228.
19. ______, Intelligent machinery, *Philosophia Mathemática* **4** (1948) 256–260.
20. ______, Computing machinery and intelligence, *Mind* **59** (1950) 433–460.
21. ______, Solvable and unsolvable problems, en copeland, en *Collected Works of A. M. Turing*, 1954, 582–595.
22. ______, *Mechanical Intelligence*, tomo 1, North Holland, 1992.
23. ______, *Morphogenesis*, tomo 3, North Holland, 1992.
24. ______, *Studies in Logic and the Foundations of Mathematics*, tomo 2, North Holland, 1992.
25. ______, *Mathematical Logic*, tomo 4, North Holland, 2001.
26. A. M. Turing, H. Putnam, y D. Davidson, *Mentes y Máquinas*, Alan Ross Anderson (compilador), Universidad Nacional Autónoma de México, 1970.
27. H. Wang, *A Logical Journey, From Gödel to Philosophy*, The MIT Press, Cambridge, Massachusetts, 1996.

28. Wikipedia, Turing machine, `http://en.wikipedia.org/wiki/Turing_machine`.
29. L. S. Zabell, 1995a, Alan Turing and the central limit theorem – an update of the 1995 american math.
30. ———, Alan Turing and the central limit theorem, *American Mathematical Monthly* **102** (1995) 483–494.