ISSN 0718-9788
Artículos / Articles
Santiago de Chile

ALAN TURING: MÁQUINAS E INTELIGENCIA. EN CONMEMORACIÓN DE LOS 100 AÑOS DE SU NACIMIENTO

Celso Vargas[1]
Instituto Tecnológico de Costa Rica

Abstract: This article was written in commemoration of the 100 hundred year of Allan Turing's birthday. It divides into two main parts. First the context in which Allan Turing proposes his automatic machines (1936-37) as a way to solve what has been known as "Entscheidungsproblem" (decision problem), that roots back to Hilbert efforts to give a new foundation to mathematics based on the logic, at the beginning of the 20th century. It was shown that Turing solution is equivalent to that proposed by Church (lambda calculus) the same year, since there, Turing-Church Thesis. The second part analyses the evolution of the concept "machine", both a(utomatic)-machines and o(racle)-machines and intelligence. It is shown that there exist continuity between his approach in 1936-1937 and that of the paper "Computing Machinery and Intelligence" published in *Mind* in 1950. Several common elements are found in these papers, even though the general strategy used in the last paper is different.

Descriptores: Turing · Church · Entscheidungsproblem · Computabilidad · Inteligencia

Resumen: Este es un artículo escrito en conmemoración del 100 aniversario del nacimiento de Allan Turing. Se divide en dos partes. En la primera se establece el contexto histórico y conceptual que lleva a Turing a proponer en 1936-37 sus máquinas automáticas como solución al problema conocido como "Entscheidungsproblem" o problema de la decisión. Este tiene su origen en el programa de Hilbert de proporcionar un fundamento nuevo a las matemáticas sobre la base de la lógica; programa éste que inicia con el siglo XX. Se probó que esta propuesta de Turing es equivalente a la propuesta por Church de ese mismo año basada en su cálculo lambda, de ahí, la Tesis Turing-Church. La segunda parte, analiza la evolución del concepto de "máquina" y de "inteligencia" que concluye con la publicación de su famoso artículo en la revista *Mind* en 1950 "Computing Machinery and Intelligence". Se muestra que en estos trabajos existe más bien continuidad más que ruptura. Varios elementos comunes se encuentran en estos trabajos, aun cuando la estrategia general utilizada en el último es bastante diferente.

Keywords: Turing · Church · Entscheidungsproblem · Computability · Intelligence

Recibido: 24/09/2012. Aceptado: 06/11/2012.

[1] Escuela de Ciencias Sociales. celvargas@itcr.ac.cr

No es extraño celebrar el aniversario del nacimiento del matemático Allan Turing en una escuela de filosofía, y no en la de matemáticas o informática como correspondería, porque, a diferencia de otros periodos de la historia, la primera mitad del siglo XX es un periodo de un gran desarrollo del pensamiento científico y en el que un grupo de filósofos comienza a trabajar muy cerca de los científicos para extraer las consecuencias filosóficas, para formular problemas a los científicos y para proponer soluciones. De hecho, muchos de los científicos de este periodo, tales como Einstein, Hilbert, Gödel, Turing, Church y otros, dedicaron buena parte de su vida académica a extraer consecuencias filosóficas de su trabajo. En el caso que nos concierne, como sabemos, el trabajo de Alan Turing "Computing machinery and intelligence" fue publicado en 1950 en *Mind*, una revista especializada en temas filosóficos. La amplia formación de estos intelectuales es una de las características más sobresalientes y de la cual debemos nosotros aprender.

Alan Turing nació en 1912 y murió en 1954. En su corta vida se ocupó de una gran diversidad de temas, todos ellos de gran relevancia en los campos de la ciencia, la tecnología y la estrategia militar. También sobresalió en el deporte. Participó como maratonista, representando a Gran Bretaña, en los Juegos Olímpicos de 1948 (Hodges, 2007). Debido a su muerte prematura, dejó varios trabajos inconclusos, en particular, en el campo de la teoría biológica y en física. Sus contribuciones fueron de un extraordinario impacto en prácticamente todas las áreas en las que incursionó.

En este trabajo nos ocupamos de dos temas exclusivamente: su concepción de los procedimientos mecánicos que son "efectivamente computables" y su perspectiva sobre la inteligencia. Como han señalado varios autores, entre ellos Hodges (2007), ya en la concepción misma de las llamadas "máquinas de Turing", Turing incluye consideraciones sobre la mente humana. De hecho, estas máquinas están pensadas teniendo en consideración la manera en la que procede el ser humano. Este trabajo tiene dos partes importantes. Primera, ubicar la propuesta de Turing sobre los dispositivos mecánicos (máquinas de Turing) y, segundo, rastrear brevemente el concepto de inteligencia en nuestro autor de referencia.

De igual importancia es dar una breve noticia sobre el contexto intelectual dentro del cual ubicamos las contribuciones tan decisivas de Turing, por lo menos, las que nos interesa analizar, y que se ubican entre los años 1936 y 1950. En 1936-37 Turing introduce su perspectiva de los procedimientos mecánicos para lo "efectivamente calculable". Este contexto está marcado por tres elementos principales: la nueva forma que adquiere el programa de Hilbert a partir de los resultados obtenidos por Paul Bernays, Wilhelm Ackermann, John von Neumann y Jacques Herbrand durante la década de los 20, los dos teoremas de Gödel en relación con la axiomatización de las aritmética y sistemas de complejidad comparable, y los trabajos de respuesta a los problemas planteados por Gödel en relación con lo que es "efectivamente calculable", específicamente, los trabajos de Alonzo Church, Kleene, Post y von Neumann.

El programa hilbertiano

El desarrollo de las geometrías no euclidianas tuvo un enorme impacto en las matemáticas y planteó la urgencia de fundamentar las matemáticas sobre bases firmes. En efecto, dichas geometrías, en el contexto en el que surgen, abren una "enorme grieta" en el concepto de "verdad" que la cultura occidental heredó de los griegos. Uno de los enormes logros de las matemáticas griegas fue la axiomatización de la geometría en la que ciertas propiedades importantes se mantenían: la transmisión de la verdad de los axiomas a los teoremas y la correspondencia de las propiedades geométricas con el espacio que nos es más cercano. De esta manera, se captaban aspectos sustantivos de la verdad como única, inmutable y necesaria. Esta visión no era exclusiva de las matemáticas, sino también de la ciencia y la filosofía en general. Como sabemos, para Aristóteles la ciencia es el conocimiento de lo necesario y de lo verdadero. Visión esta que podemos rastrear en Descartes, Leibniz y Kant, entre otros.

Las dudas en torno el postulado V de las paralelas (aquel que afirma que dadas una recta *a* y un punto exterior a ella *B*, se puede trazar una y solo una recta *b* paralela a *a*) llevó al desarrollo de estas nuevas geometrías. Su negación, el que no puede trazarse ninguna o que pueden trazarse al menos dos rectas paralelas, es decir, se pueden encontrar al menos tres rectas que no se cortan en dicho plano, está a la base de estas nuevas geometrías. Dos geometrías principales surgieron tomando una u otra de estas alternativas: la geometría de Lobachevski y la de Riemann (Aleksandrov, 1982). Lobachevski adopta en lugar del quinto postulado euclidiano la segunda alternativa, mientras que Riemann, la primera.

El desarrollo de estas dos geometrías durante la segunda mitad del siglo XIX cuestionó profundamente el concepto de verdad y planteó la urgencia de desarrollar métodos para probar la consistencia tanto de la geometría, la aritmética y de los nuevos desarrollos de las matemáticas. Hilbert (1862-1943) será uno de los que más se ocupará de estos problemas y de la urgente necesidad de fundamentar las matemáticas sobre una base nueva. Durante la última década del siglo XIX, Hilbert realizará enormes esfuerzos por poner en marcha un programa de fundamentación de las matemáticas. Trabajo este que culmina con el gran éxito de su obra *Foundations of Geometry* (1899). En dicha obra, Hilbert utiliza un método de consistencia indirecta, es decir, deriva la geometría de la aritmética, mostrando que las verdades geométricas son también verdades aritméticas. Si la geometría euclidiana es consistente, también lo son las otras geometrías. Pero la gran interrogante se cierne en la consistencia de la aritmética.

El tiempo pronto mostrará los enormes problemas asociados con el establecimiento de pruebas de consistencia. Esta primera estrategia de Hilbert de probar consistencia de manera indirecta, es decir, expresando cierto tipo de resultados por reducción a otro campo (como en el caso de la geometría a la aritmética) llevó pronto a problemas. En 1902 Russell comunica a Frege que su sistema de aritmética presenta una paradoja, lo que cuestiona los fundamentos de la axiomatización de Frege. Pronto se verá que las pruebas de consistencia desarrolladas por Hilbert y sus seguidores, en particular el trabajo que realiza Dedekind, descansa en supuestos tan problemáticos como aquellos seguidos por Frege.

Esto obliga a que el programa se mueva hacia la búsqueda de pruebas de consistencia directa, es decir, debemos comenzar por construir un sistema desde cero y de manera incremental ir creando una base cada vez más amplia en la que podamos expresar los principales resultados de las matemáticas: primero, la aritmética ordinaria; luego, los números enteros, los racionales, los reales y finalmente, los complejos. Así pues, el desarrollo de sistemas axiomáticos abstractos, sin ninguna suposición de partida o de la intuición, y el desarrollo de métodos de consistencia interna, constituyen los dos elementos principales de este programa. Un fuerte impulso al programa de Hilbert vendrá de los tres volúmenes de la monumental obra *Principia Mathematica* de Russell y Whitehead (1910-1913). Esto hizo que Hilbert, que había dejado de lado sus trabajos sobre los fundamentos de las matemáticas para ocuparse, entre otros, de temas físicos relacionados con la teoría especial y general de la relatividad, volviera en 1917 y retomara el tema a partir del impulso proporcionado por Russell y Whitehead.

Como señala Zach (2003), p. 2 "Hilbert mismo volvió a trabajar sobre temas de fundamentación en 1917. En Septiembre 1917, dio una conferencia a la Sociedad Matemática Suiza, titulada "Pensamiento Axiomático". Es la primera contribución publicada sobre los fundamentos de las matemáticas desde 1905. En ella de nuevo enfatiza el requisito de pruebas de consistencia para sistemas axiomáticos. El principal requisito de la teoría de axiomas debe ir más allá (que meramente evitar las paradojas conocidas), es decir, a mostrar que dentro de cada campo del conocimiento construido a partir sistemas de axiomas, las contradicciones son absolutamente imposibles. Plantea la prueba de la consistencia de la aritmética (y de la teoría de conjuntos) de nuevo como los principales problemas a resolver (*open problems*). En ambos casos, no parece haber nada disponible más fundamental que reducir la consistencia de éstos a la lógica misma. Y Hilbert pensó, entonces, que el problema había sido resuelto esencialmente por Russell en su trabajo de los *Principia*. Sin embargo, otros problemas fundamentales de la axiomática permanecen sin resolver, incluyendo el de la decidibilidad de cada cuestión matemática".

Así pues, Hilbert y sus colaboradores, en particular Bernays y Behmann, comienzan a avanzar en los nuevos desarrollos de la lógica con el objetivo de resolver estos problemas. Esto da un nuevo aliento al programa hilbertiano. Sin embargo, pronto los problemas surgirán de nuevo en el seno mismo del equipo. En 1921, Weyl, uno de los discípulos de Hilbert, abandona el formalismo y adhiere el intuicionismo en lógica. Como se recordará, el intuicionismo se basa en dos tesis muy fuertes: la matemática "es una creación no lingüística (*languageless*) de la mente" y la aceptación del continuum (infinito) a partir de conjuntos seleccionados con antelación: lemhoff (2008). Las consecuencias para el programa formalista obliga a que Hilbert defienda el formalismo contra esta nueva visión de la lógica y las matemáticas y que busque construir las matemáticas a partir de una visión finitista a partir de 1921, en la que se acepta como dado cierto conjunto de marcas (*signs*) que son la base de la construcción de la teoría de los números (algo ligeramente similar a las clases de Russell).

Esta constituye la última fase en la evolución del programa hilbertiano: fundamentar las matemáticas desde una perspectiva finitista. Dentro de esta nueva forma es que Hilbert plantea los tres problemas fundamentales (1928), respecto de los cuales comienza a trabajar Gödel:

A. "¿Son las matemáticas completas, en el sentido de que cualquier postulado pueda ser probado o rechazado?"

B. "¿Son las matemáticas consistentes, en el sentido de que nunca se pueda demostrar algo que sea manifiestamente falso?"

C. "¿Son las matemáticas decidibles, en el sentido de que se puede crear un sistema de deducción paso a paso que aplicado a cualquier postulado permita determinar si es cierto o falso?"

Los dos teoremas de Gödel

Los tres problemas, mencionados anteriormente, son de naturaleza muy general. Pero Gödel comienza a trabajar en los dos primeros en la forma en que fueron presentados por Hilbert y Ackermann en su trabajo de 1928, titulado *Grundzüge der theoretischen Logik* (Hilbert and Ackermann, 1928), utilizando la lógica de predicados de primer orden como el cálculo dentro del cual presentar una forma más específica. Como señala Heijenoort (1967), el problema de la completitud tal y como se presenta en esta obra consiste en probar si dado un sistema de axiomas para la lógica de predicados de primer orden "… es completo en el sentido de que de él pueden ser derivadas todas las fórmulas lógicas que son correctas para cada dominio de individuos".

Gödel aborda en su tesis doctoral este problema y prueba la completitud del cálculo de predicados de primer orden para un sub-lenguaje de fórmulas con ciertas características específicas, en particular, para un lenguaje que no exprese identidad. Estos resultados fueron realizados durante 1929 y publicados en el año siguiente.

De esta manera, Gödel estaba muy familiarizado con el programa de Hilbert y con los logros que se habían alcanzado y de los enormes retos que se tenían por delante. Observamos que Gödel obtiene nuevos resultados ya en ese mismo año de 1930. En este mismo año presentará sus primeros resultados de incompletitud en un congreso sobre epistemología de las matemáticas celebrado en Königsberg en Octubre de 1930. Es un año muy particular, debido a lo siguiente: fue en ese año que Hilbert se había retirado y fue invitado como conferencista distinguido en este congreso de Octubre que se realizó en su ciudad natal. Como señala Soare (2009), la disertación de Hilbert versó sobre la importancia de las matemáticas en la ciencia y de la lógica en las matemáticas y reiteró la idea de que no hay problemas irresolubles, bajo el principio "Debemos conocer y conoceremos". Este lema, Hilbert lo había señalado en varias ocasiones, como una visión optimista sobre las posibilidades de fundamentar las matemáticas sobre una base nueva. Le había precedido en el uso de la palabra, en ese congreso, un joven matemático, prácticamente desconocido, que había concluido sus estudios doctorales un año antes: Kurt Gödel. En esta conferencia, Gödel presenta su primer teorema de incompletitud.

En dicha prueba, Gödel introduce una serie de técnicas matemáticas nuevas, comenzando con las funciones recursivas primitivas las cuales serán fundamentales tanto en el trabajo de Turing como de Church, Kleene, Post y Von Neumann, como vamos a ver. Ha sido tradicional distinguir entre lenguaje objeto (el lenguaje al *que nos estamos refiriendo*) y metalenguaje (en *el que estamos hablando*). El metalenguaje debe ser suficientemente rico para permitir expresar tanto el lenguaje objeto como otro tipo de propiedades que son de interés en el proceso de demostración. Pues bien, y ésta es la segunda técnica, Gödel introdujo un nuevo sistema de codificación o numeración que le permite expresar cualquier expresión metalingüística en términos aritméticos, es decir, "hacer que la aritmética hable sobre sí misma". Este sistema de codificación es tal que cada expresión recibe uno y solo un número, lo que permite que podamos pasar de la expresión aritmética a la expresión metalinguística y viceversa. Esto le posibilitó construir en el mismo lenguaje aritmético una expresión aritmética que expresa su propia imposibilidad de ser demostrada, es decir, como lo denomina Soare (2009), encontró una nueva forma de formalizar la *paradoja del mentiroso*.

Antes de enunciar esta primera prueba es necesario hacer dos aclaraciones. Gödel construye su prueba de incompletitud sobre un lenguaje lógico (lógica de predicados de primer orden, enriquecido con identidad (tres en total) y con los nueve axiomas de Peano para la aritmética elemental). Este sistema es denominado P (la artimética de Peano). Segundo, Gödel utiliza "ω-consistente" para referirse a que los números naturales satisfacen los axiomas de Peano. Así pues, el primer teorema establece que, "Si P es ω-consistente, entonces, existe una oración la cual no es probada ni refutada en P".

Una oración de este tipo, como hemos mencionado, es aquella que expresa la imposibilidad de su demostración. El proceso de construcción de una oración de este tipo escapa al alcance de este trabajo. Pero pueden consultarse diferentes fuentes. Más formalmente Hamilton (1981); un poco más informalmente, Newman y Nagel (1958).

Tal y como es indicado por Soare (2009) y otros autores, von Neumann se encontraba entre la audiencia que escuchó esta conferencia. Muy pocos matemáticos comprendieron esta prueba y las implicaciones de ésta para las matemáticas y, en particular, para el programa de Hilbert. Von Neumann, quien asistió a este congreso en representación del programa hilbertiano, de inmediato entendió perfectamente las implicaciones de esta demostración. Como señala Soare (2009), p. 8, "(…) y reconoció que el programa de Hilbert había terminado. En las siguientes semanas von Neumann se percató que aritmetizando la prueba del primer teorema de Gödel, uno podría probar un teorema aún mejor, que no existe un sistema formal T (con capacidad de expresar la aritmética ordinaria) tal que pueda probar su propia consistencia. Pocas semanas después él llevó su prueba a Gödel, el cual le agradeció y le informó con gentileza que él había ya sometido a publicación el Segundo Teorema de Incompletitud." Este segundo teorema es, pues, una generalización del primero y establece la incompletitud de la teoría de números.

Hay varios aspectos que deben recalcarse. Primero, lo que prueba Gödel es que si un sistema de axiomas de aritmética es consistente (no tiene contradicciones), entonces no puede demostrar todas las verdades aritméticas. Existe, pues, un conjunto numerable de expresiones que no pueden ser probadas dentro de ese sistema axiomático. Segundo, más aún: ningún sistema axiomático para la aritmética puede probar todas las verdades de la aritmética. Así pues, se tiene que tomar la siguiente decisión: mantener consistencia y sacrificar completitud, o mantener completitud y sacrificar consistencia.

Claramente, estos resultados referidos al primero y segundo de los problemas planteados por Hilbert en 1928, tiene fuertes implicaciones en el tercero. Vamos a recordar este problema de nuevo con propósitos expositivos.

C. "¿Son las matemáticas decidibles, en el sentido de que se puede crear un sistema de deducción paso a paso que aplicado a cualquier postulado permita determinar si es cierto o falso?"

En efecto, en términos generales, se puede concluir que la respuesta a este problema es negativa, ya que habrá "postulados", para usar la expresión que aparece en el enunciado del problema, que no pueden ser probados ni refutados. Lo más importante, ahora, es encontrar un procedimiento o mecanismo que nos permita definir con claridad el alcance de lo que debemos entender por "efectivamente calculable" o sistema de deducción paso a paso. En adelante, este problema será conocido como 'Entscheidungsproblem' (el problema de la decisión). Como vamos a ver, Turing mostrará un gran interés en el *Entscheidungsproblem* y su propuesta de las máquinas de Turing, es la respuesta al mismo. Sin embargo, antes de hacerlo, varios matemáticos habían ya comenzado a hacer este trabajo incluyendo al mismo Gödel. A partir de 1931, Gödel se convertirá en punto de referencia y árbitro respecto de las propuestas que harán Church, Kleene y Turing en torno a este problema.

El papel de Church y de Kleene

Después de estos primeros trabajos de Gödel vemos aparecer un grupo muy joven de destacados matemáticos y lógicos que comienzan a ocuparse del *Entscheidungsproblem*. Todos ellos con el objetivo de probar la imposibilidad de un algoritmo de decisión bajo la influencia directa de los resultados de Gödel. El primero de ellos es Alonzo Church (1902-1995). Obtuvo su doctorado en 1927 en Princeton. Es becado por dos años (1928-1929) lo que le permite visitar Harvard, Göttingen y Amsterdam. Durante este viaje conoce a Bernays y a Heyting. Es muy probable que fuera durante esta visita que se enterara con detalle sobre el programa de Hilbert. En el año de 1933-34 Gödel visita el Institute for Advanced Study de Princeton, donde dictó conferencias sobre los resultados de sus investigaciones recientes, incluyendo los teoremas de incompletitud, y en 1935 se traslada de manera permanente a Princeton. Es a partir de aquí que Church comienza a trabajar en el problema de la decisión y lo hace con base en su cálculo lambda (cálculo λ) de reciente construcción. Investiga este problema con sus discípulos Stephen Kleene y Barkley Rosser, explorando varias alternativas, todas ellas muy fructíferas para la teoría de la computabilidad. También von Neumann se encontraba en Princeton. En este sentido, Princeton se convierte en un centro muy importante para el desarrollo de esta incipiente disciplina, cuyos resultados comenzamos a ver los a partir de la década de los 40.

Como señala Soare (2009), p. 9, Church presenta a Gödel sus primeros resultados en marzo de 1934 en torno a "funciones que son efectivamente calculables, el término que en los años 30 se utilizaba para una función que es computable en el sentido informal" Kleene, que había obtenido su doctorado en 1934, cuando escucha de este resultado trató de refutarlo utilizando el método de diagnonalización de

Cantor, probando lo contrario, es decir, que toda función definible es efectivamente computable utilizando el cálculo lambda. Este resultado se conoce como el teorema de la recursión de Kleene. A partir de aquí Kleene se convence de que Church ha logrado establecer un procedimiento general para entender lo que es efectivamente calculable, y a partir de esta base responder negativamente al problema planteado por Hilbert.

Sin embargo, Gödel no será de la misma opinión. Este autor tendrá serias dudas respecto a que Church haya encontrado tal procedimiento general. Primero, dudaba que la teoría sobre la que Church había establecido sus resultados fuera completa. De inmediato él comenzó a trabajar el concepto de 'función recursiva', desarrollado por Hebrand con el fin de encontrar una teoría más general con la que corregir algunos de los problemas encontrados (teoría que se conoce con el nombre de Herbrand-Gödel) y que va a ser adoptada por Kleene casi de inmediato. La segunda duda se relaciona con el hecho de que lo que ofrece Church es más bien una heurística y no una demostración. Para citar a Soare, "era su (de Gödel) aprobación lo que Church más deseaba. Chruch había resuelto el Entscheidungsproblem solo si su caracterización de funciones efectivamente calculables era precisa". Caracterización ésta de la que Gödel no estaba convencido. Kleene va a utilizar estos nuevos desarrollos de Gödel y Herbrand para expresar el concepto churchiano de 'efectivamente calculable' o 'definible'. Aun así las dudas persistían. No será sino hasta que Gödel conozca el método utilizado por Turing que considerará que el problema ha sido resuelto de la manera más satisfactoria.

El concepto de lo 'efectivamente contable' de Turing

Alan Turing publica en 1936 su trabajo titulado "On Computable Numbers, with an Application to the Entscheidungsproblem". Tuvo problemas para que fuera publicado debido a su corta edad y a que era todavía no suficientemente conocido y, además, Church había sometido a consideración de la London Mathematical Society un artículo sobre el mismo tema. Tuvo que intervenir el matemático Max Newman para que esta Sociedad accediera a publicar el artículo de Turing. Max Newmann, un topologista, jugó un papel muy importante en la vida de Turing. Fue en 1935 que Turing asistió a las lecciones que impartió Newman sobre Hilbert, Gödel y el Entscheidungsproblem. Fue aquí la primera vez que Turing tuvo conocimiento de estos problemas fundamentales de las matemáticas y de los desarrollos alcanzados en su solución. De inmediato comenzó a trabajar. No tenía conocimiento de todos los trabajos que se estaban desarrollando en Princeton, excepto los dos artículos de Church publicados en 1936, tanto en el *Journal of Symbolic Logic* como en la American Society of Mathematics, a los que hace referencia en su publicación de

1936. La segunda oportunidad en la que Newman tuvo que ver directamente con Turing fue cuando le rechazaron el artículo a Turing en el que describía su solución al Entscheidungsproblem. Fue en Abril de 1936 que "Turing mostró su solución al perplejo Max Newman. (…) Newman los persuadió de que el trabajo de Turing era suficientemente diferente, y ellos publicaron el trabajo de Turing en el volumen 42 del 30 de Noviembre de 1936 y el 23 de Diciembre de 1936", Soare (2009), p. 12.

1936 es uno de los años más fructíferos para la teoría de la computabilidad. No solamente se publicaron los trabajos de Turing, sino también los de Church, Kleene y Post sobre diferentes alternativas para abordar el problema de la parada o de decisión. Pero lo que lo hace también interesante es que se pudo analizar las distintas soluciones y comparar sus méritos. Claramente, la solución de Turing no solo será considerada la que mejor expresaba el concepto de lo 'efectivamente computable', sino que también permite ver en perspectiva las soluciones de Church, Kleene y Post, de manera que puede sustentarse su equivalencia. En efecto, es uno de los principales resultados de la teoría de la computabilidad la equivalencia entre todos estos distintos formalismos: las máquinas de Post, las de Turing, el Cálculo lambda de Church y teoría de la recursión de Kleene. Pero el establecimiento de la equivalencia se hará paso a paso. En efecto,en 1936 Allan Turing aún no había concluido sus estudios doctorales. Escribirá su tesis bajo la supervisión de Church en 1938. Aquí Turing establecerá la equivalencia entre el formalismo de Church (el cálculo lambda) y las máquinas de Turing. Esta equivalencia tendrá enorme importancia en el campo de las matemáticas y en la teoría de la computabilidad como mencionaremos más adelante.

En el trabajo de Turing publicado en dos partes en 1936, el autor solo conoce los trabajos de Church en torno al concepto de lo 'efectivamente calculable', pero también señala que su aproximación será bastante diferente a la de Church. El autor introduce dos tipos de máquinas. Las que él denomina 'automatic machines' (a-machines), que son aquellas cuyo comportamiento está completamente determinado por las instrucciones que se le proporcionan (abbreviated tables of behavior), y por tanto, son máquinas de Turing determinísticas. Tal y como lo indica Soare (2009), p. 31, este tipo de máquinas corresponden a las denominadas "computaciones offline" dentro de las arquitecturas de computación actuales, en las que no es necesario utilizar recursos externos para mejorar o tomar una decisión. Las a-máquinas son las más conocidas. Sin embargo, Turing también introduce una segunda clase de máquinas, conocidas como c-machines (choice machines, o como las expresará Kleene años después, oracle machines). Estas máquinas son una extensión de las anteriores y agrega la siguiente característica: en determinadas condiciones, cuando hay ambigüedad o 'incertidumbre' respecto a la decisión que se debe tomar, se consulta a un "operador externo", a fin de clarificar la decisión que debe tomarse. Este tipo de máquinas corresponden al modelo 'online' en el que el dispositivo o la persona que está tomando la decisión, necesita consultar fuera del 'programa' a fin de tomar esta decisión. Este tipo de modelos refleja más claramente las características interactivas que encontramos en este momento, en el que nosotros, mientras escribimos, consultamos bases de datos en la red www, libros en formato digital o en formato expreso para verificar o clarificar un concepto. Es interesante que Turing haya considerado en este trabajo pionero este tipo de máquinas. Sin embargo, solo le dedica un pequeño párrafo en la sección de 'definiciones' sin mayores desarrollos. Tal y como lo indicamos anteriormente, será Kleene el que le dará una mayor estructura en 1946.

Comencemos por la presentación de las a-máquinas y algunos de los resultados generales que obtiene Turing, para pasar luego a hacer algunas referencias generales a las c-máquinas. La representación que utiliza Turing para este tipo de máquinas es muy sencillo e intuitivo: ésta consta de una cinta constituida por un conjunto de celdas que se extiende de manera indefinida en ambas direcciones; en cada una de las celdas se puede registrar un único símbolo; tiene además una cabeza de lectura/escritura cuyo comportamiento está determinado por las configuraciones o reglas, y finalmente, dos movimientos básicos: hacia la derecha R o hacia la izquierda L (obviando aquel caso en el que la máquina permanece en la misma celda después de llevar a cabo una operación). Las operaciones que sobre la cinta puede realizar la máquina son de tres tipos diferentes: borrar el símbolo que se encuentra en la celda (sustituyéndolo por un blanco), reemplazarlo por uno nuevo, o simplemente, reescribir el mismo símbolo. En cada caso, toma la decisión de avanzar o retroceder una posición en la cinta o permanecer en la misma celda.

Este concepto se puede formalizar sin necesidad de hacer referencia a este modelo físico, de la siguiente manera:

Una máquina de Turing es una séxtuple $< Q, \Sigma, \Gamma, q_0, \delta, F >$ donde:

- Σ un alfabeto de entrada, digamos, $\Sigma = \{a, b, c, d, \ldots.\}$
- Γ un alfabeto auxiliar, conteniendo tanto a Σ como a otros símbolos, en particular, B (el blanco), es decir, $\Gamma = \Sigma \cup \{B.,,,\}$
- Q un conjunto de estados, $Q = \{ q_o, q_1, q_2, \ldots \}$
- $Q_o \in Q$, denominado estado inicial
- $F \subseteq Q$, el conjunto de estados finales
- δ es una función de transición definida como $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

Una descripción de estado, es decir, el comportamiento de la máquina en un momento determinado puede ser descrito en términos de los siguientes cinco elementos:

$(q_i, X, Y, q_h, \{L,R\})$, es decir, el estado en el que se encuentra, el símbolo que está leyendo, el símbolo que reemplazara, el nuevo estado al que cambiará y el movimiento que realizará a la derecha o a la izquierda.

Veamos un ejemplo muy sencillo: la suma de dos números naturales. Estos estarían representados en la cinta mediante dos secuencias de 1s, cada una de los cuales representa el número que se desea sumar; la primera secuencia está separada de la segunda por un 0, y cada una de las secuencias está también delimitada por 0s, a la izquierda o a la derecha de las secuencias, respectivamente. Una secuencia de este tipo es la siguiente:

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

En este caso, corresponde a la suma de 8 + 5.

La máquina de Turing en sus componentes puede ser definida de la siguiente manera (aunque hay muchas otras alternativas):

- $\Sigma =\{1\}$
- $\Gamma = \Sigma$ U $\{0\}$
- $Q =\{ q_o,q_1,q_2,q_3\}$
- Q_o estado inicial
- Q_3 el estado final
- δ viene definida por las siguientes transiciones:

$(q_o, 1) = (q_o,1, R)$
$(q_o, 0) = (q_1, 1, R)$
$(q_1, 1) = (q_1, 1, R)$
$(q_1, 0) = (q_2, 0, L)$
$(q_2,1) = (q_3, 0, L)$

En este ejemplo, las dos primeras reglas señalan que la máquina ha analizado el primer sumando, cuando termina, es decir, cuando encuentra un 0, cambia de estado, indicando que se prepara para analizar el segundo sumando, inserta un 1 a fin de obtener la suma total. Las reglas tercera y cuarta corresponden al análisis del segundo sumando. Cuando encuentra un 0, cambia de estado, lo que indica que ha terminado de recorrer dicho número, se devuelve una posición y reemplaza el 1 por un 0, y se detiene. En este ejemplo, hemos supuesto que ambos sumandos son mayores que 0. No consideramos el caso especial en el que uno o ambos son igual a 0. Pero una máquina que considere todos los casos (ambos sumandos son 0, o alguno de ellos es 0) resulta fácil de construir.

Hay tres usos generales que se le dan a las máquinas de Turing (también a otras clases de mecanismos computacionales): a) como aceptores, es decir, como dispositivos que aceptan determinados tipos de lenguajes o conjuntos, en particular, los conjuntos recursivamente enumerables; b) como generadores, es decir, que generan un conjunto de funciones computables (las llamadas funciones parciales) de manera que, dada una función parcial, existe una máquina de Turing que simula las salidas de la función, y c) como algoritmos, es decir, como procedimientos para responder *sí o no* en determinadas condiciones. Por ejemplo, si una determina expresión es parte del lenguaje aceptado/generado por una máquina: Manna (1974).

Turing hace uso, en su artículo de 1936, de dos de estos usos principales. Este artículo está escrito en la forma que permite probar que determinadas secuencias computables, en este caso, aquellos números reales cuyas secuencias decimales pueden ser probadas utilizando medios finitos y caracterizar éste como un conjunto con determinadas propiedades, en particular el de ser recursivamente numerable. Claramente, se está haciendo el uso de las máquinas como generadores. En efecto, podemos definir las máquinas de Turing en términos de las clases de funciones que generan. Esto será fundamental para la llamada 'Tesis Turing-Church'. Pero cuando Turing prueba la existencia de una máquina universal de Turing, utiliza el concepto de aceptor con este propósito, es decir, muestra que para cualquier función computable hay una máquina de Turing que la acepta. Se trata de un proceso de simulación, diferenciando los principales casos. Dado que esta clasificación es posterior, Turing no se ocupa de hacer este tipo de diferencias, sin embargo, están presentes en su trabajo y son diferenciables.

En relación con la a-máquinas, tres resultados generales son de importancia en esta presentación:

1. Cualquier secuencia computable (función) puede ser calculada por una máquina de Turing. Esto es conocido como la máquina universal de Turing. Este resultado es muy importante pues prueba que las máquinas de Turing son los dispositivos más generales que podemos construir. Este es uno de los principales resultados de la teoría de la computabilidad: las máquinas de Turing establecen el límite de lo computable.
2. En un Apéndice agregado en Agosto de 1936, Turing hace un esbozo de la equivalencia entre lambda-definible (λ-definible) y computabilidad. Utiliza Turing la formalización realizada por Kleene de los números enteros para mostrar que toda expresión λ-definible puede ser aceptada por una máquina de Turing y al revés, toda expresión generada por una máquina de Turing, con características específicas, puede ser expresada en términos de λ-definible. Esta es la conocida tesis Turing-Church. Sin embargo, como mencionamos anteriormente, será en 1938 cuando Turing complete esta prueba. Pero esta equivalencia tendrá una importancia fundamental. En efecto, si toda función recursiva es Turing computable, entonces, es suficiente con probar que nos encontramos ante una función recursiva para garantizarnos de que existe una máquina de Turing que la reconoce. Podemos, pues, trabajar con un nivel de

independencia importante sin tener que estar haciendo corresponder nuestras funciones con las máquinas de Turing.

3. Su solución al Entscheidungsproblem. Turing muestra que el problema C. "¿Son las matemáticas decidibles, en el sentido de que se puede crear un sistema de deducción paso a paso que aplicado a cualquier postulado permita determinar si es cierto o falso?" tiene que ser contestado negativamente. Aquí Turing utiliza la formulación del problema que hicieran Hilbert y Ackermann en 1931. La prueba adquiere la forma siguiente: "Propongo, por tanto, mostrar que no puede existir un proceso general para determinar si dada un fórmula U del cálculo funcional K <de Hilbert> puede probarse, es decir, que no puede existir una máquina que teniendo una de estas fórmulas U, eventualmente diga si se puede probar" (Turing (1937), p. 259). Claramente, como se ha mencionado anteriormente, este resultado se corresponde de manera perfecta con los teoremas de incompletitud de Gödel: da fundamento a la sospecha que se mantenía en relación de que el Entscheidungsproblem tiene una respuesta negativa.

La demostración de que lo efectivamente computable puede ser decidido por una máquina de Turing y el que no exista un procedimiento general para decidir la verdad de cualquier verdad matemática, presenta un gran atractivo, pues es independiente de cualquier formalismo. Como ya mencionamos anteriormente, Gödel tenía dos objeciones principales a la propuesta de Church sobre lo efectivamente computable: una primera que se relaciona con la extensión, es decir, con la seguridad de que el cálculo lambda captura efectivamente todas las funciones computables, y segundo, con la intensión del concepto, es decir, la duda de que existan otras situaciones, independientes del cálculo utilizado que queden también por fuera de la demostración de Church. Gödel se convenció de que la propuesta de Turing proporciona una seguridad mucho más general de que el concepto de lo 'efectivamente computable' se había delimitado con la precisión necesaria para captar el uso ordinario del concepto de procedimiento finito. Así lo señala Gödel, "que esta es realmente la correcta definición de computabilidad mecánica, (la cual) fue establecida más allá cualquier duda por Turing". Soare (2009), p. 14.

El otro campo de una extraordinaria fecundad del trabajo de Turing son sus 'oracle machines' (máquinas oráculo), que como indicamos anteriormente, se parecen mucho más al tipo de procedimientos que nosotros hemos utilizado y que hemos aprendido a potenciar después del advenimiento de la red Internet: la consulta de diversas fuentes de información para tomar una determinada decisión. Representan una arquitectura online de computadora. Las a-máquinas son un caso particular de las o-máquinas y, por tanto, presentan una mayor riqueza conceptual. Soare ha sido uno de los que ha dedicado más tiempo a establecer las propiedades de estas máquinas, así que las breves referencias que hagamos están basadas en su trabajo de 2009 (un trabajo en el que prueba varias propiedades de este tipo de máquinas).

Una o-máquina de Turing es similar a la descrita anteriormente, excepto que agrega una o más alfabetos adicionales (los cuales pueden ser considerados como cintas con celdas similares al ejemplo dado anteriormente), las cuales son consultadas por la máquina en determinadas condiciones. Como señala Soare (2009), es suficiente con una de estas cintas adicionales, aunque no es obligante. En el caso que utilizamos aquí es necesario utilizar una sola cinta externa (de consulta del oráculo). Repitamos de nuevo los elementos de una a-máquina de Turing agregando los elementos nuevos:

Una máquina de Turing es una séptuple $< Q,\Sigma,\omega,\Gamma,q_0,\delta,F >$ donde:

- Σ un alfabeto de entrada, digamos, $\Sigma =\{a, b, c, d,\ldots\}$
- Γ un alfabeto auxiliar, conteniendo tanto a Σ como a otros símbolos, en particular, B (el blanco), es decir, $\Gamma = \Sigma$ U {B.,,,}
- ω es el conjunto o alfabeto del oráculo $\omega = \{ \varphi_1 \text{ U } \varphi_2 \text{ U},\ldots,\text{U } \varphi_i\}$
- Q un conjunto de estados, $Q = \{q_o, q_1, q_2,\ldots \}$
- $Q_o \omega \in Q$, denominado estado inicial
- $F \subseteq Q$, el conjunto de estados finales
- δ es una función de transición definida como $\delta : Q \times \omega \text{ x } \Gamma \rightarrow Q\times\Gamma\times\{L,R\}$

Como se observa, hay dos ampliaciones principales: el agregar el alfabeto del oráculo y en la transición se toma en cuenta, en algunos momentos, la información que le proporciona el oráculo a fin de tomar una mejor decisión.

Retómenos el ejemplo anterior, pero modificándolo de manera que la máquina tenga que decidir en un momento determinado si lo que debe realizar es una suma o una resta. Usaremos como alfabeto del oráculo (ω) los símbolos +, - que corresponden a suma y resta respectivamente. También ampliamos el alfabeto Σ con el signo "-" que corresponde a aquellos casos en los que el resultado de la resta es negativo, es decir, en el caso en el que el sustraendo sea mayor que el minuendo. Adicionalmente, agregamos un símbolo bandera d, miembro de Σ, con el fin de delimitar la segunda de las expresiones. Así pues, la cinta tendría la siguiente forma:

			0	0	1	1	1	1	1	1	1	1	d	1	1	1	1	1	d	0				

El nuevo conjunto de reglas es el siguiente (las separamos con propósito de visualización):

$(q_o, 1) = (q_o, 1, R)$
$(q_o, +,d) = (q_1, d, R)$
$(q_1, 1) = (q_2, 1, L)$
$(q_2, d) = (q_3, 1, R)$
$(q_2, 1) = (q_3, 0, R)$
$(q_3, d) = (q_4, 0, L)$

$(q_o, 1) = (q_o, 1, R)$
$(q_o, -, d) = (q_5, d, R)$
$(q_5, 1) = (q_6, 1, R)$
$(q_6, d) = (q_7, 0, L)$
$(q_7, 1) = (q_8, d, L)$
$(q_8, 1) = (q_8, 1, L)$
$(q_8, d) = (q_9, 1, L)$
$(q_8, 0) = (q_9, 0, R)$
$(q_9, 1) = (q_5, 0, R)$
$(q_9, d) = (q_{10}, d, R)$
$(q_{10}, 1) = (q_{11}, 1,L)$
$(q_{11}, d) = (q_{11},-, \ )$

Esta o-máquina de Turing toma la decisión de qué operación realizar mediante consulta al oráculo, es decir, si debe realizar una suma o una resta. Como se mencionó anteriormente, este tipo de máquinas son mucho más generales que la a-máquinas de Turing y, por tanto, computan conjuntos mucho mayores de funciones continuas que, de otro modo, no serían computables. Remitimos al trabajo de Soare (2009) para una visión más amplia de este importante campo de la teoría de la computación. Lo que sorprende es la genialidad de Turing al visualizar estos dispositivos tan generales aun cuando no fue él sino Kleene el que le dará un mayor desarrollo.

Para finalizar esta sección, mencionemos que la estrategia seguida por Turing es particular y muy fructífera. Primero, expresa el concepto de algoritmo o de lo ‘efectivamente computable’ a un dispositivo mecánico que realiza una instrucción cada vez. Una vez realizado esto, prueba que ninguna máquina de Turing prueba resolver el problema Entscheidungsproblem. Es decir, un *no* a la tercera pregunta de Hilbert de 1928.

Evolución del concepto de inteligencia

En 1950 Turing publica en la revista *Mind* el trabajo “Computing Machinery and Intelligence” en el que analiza la cuestión de si “puede una máquina pensar”.

En 1948 es la primera vez que encontramos un escrito de Turing en el que se hace mención explícita al concepto de inteligencia. Se trata de un reporte para el National Physical Laboratory del Reino Unido. Podría, entonces, pensarse que se trata de una evolución de su pensamiento en el sentido de que procura extender sus resultados anteriores (y maduros) a este nuevo campo de la computación conocido ahora como 'inteligencia artificial' y que es aquí cuando se introduce en campo de la filosofía. Sin embargo, no es así. En efecto, desde los primeros trabajos de Turing, específicamente, el de 1936 que hemos ya comentado, hay referencias claras al 'agente humano' como fuente de inspiración para sus desarrollos; lo mismo ocurre con la referencia a 'dispositivo mecánico'. Comencemos por el segundo. Su concepto de 'dispositivo mecánico' está basado en dos conceptos muy familiares. Como señala Hodges (2007) "el formalismo de la máquina de Turing fue modelada sobre el impresor remoto (teleprinter), ligeramente ensanchado en ámbito para permitir una cinta de papel que podía moverse en ambas direcciones y una 'cabeza' que podía leer, borrar e imprimir nuevos símbolos y no solo leer y perforar huecos". Claramente es una de las ideas geniales de Turing, utilizar dispositivos conocidos para visualizar un modelo de gran generalidad y de tan diversas aplicaciones en el campo de la ciencia y la computación.

Por otro lado, este concepto de máquina, un concepto muy intuitivo y próximo, como señala Turing, a la división del papel en cuadrados "similar a los libros de aritmética para niños". En este sentido, Turing está tratando de emular la manera en la que los humanos llevamos a cabo determinadas operaciones. De hecho, cuando discute las justificaciones para la afirmación con la inicia el artículo de que "los números 'computables' pueden ser descritos brevemente como los números reales cuyas expresiones como un decimal son *calculadas por medios finitos*", señala que "para este caso, solo se debe decir que la justificación radica en el hecho de que la memoria humana es necesariamente limitada" (el énfasis es agregado). En este sentido, 'medios finitos' significa "según las capacidades de memoria humana". Como señala Shagrir (2006), p. 395, Turing compara a "un hombre en el proceso de cálculo de un número real" con "una máquina que solo es capaz de un número finito de condiciones".

Esto significa que desde el inicio Turing está considerando determinada perspectiva humana en el diseño de las máquinas. Este aspecto no se presenta en ninguno de los otros autores de esta misma década. Sin embargo, las comparaciones entre estos dispositivos y los seres humanos que encontramos en este primer trabajo de Turing no terminan allí. En efecto, es paradigmático el siguiente párrafo de Turing en el que justifica las restricciones adoptadas en la construcción de su dispositivo:

"El comportamiento de un computador en cualquier momento está determinado por los símbolos que está observando, y su 'estado de mente' (state of mind) en ese momento. Podemos suponer que existe un límite B en el número de símbolos o cuadros que el computador puede observar en un momento. Si desea observar más, debe hacer observaciones sucesivas. Podremos también suponer que el número de estado de mente que necesita tomar en consideración es finito. Las razones para esto son del mismo tipo que aquellas que restringen el número de símbolos. Si admitimos un infinito número de estado de mente, algunos de ellos serían 'arbitrariamente cercanos' y serían confundidos. De nuevo, la restricción no afecta seriamente el cálculo, dado que el uso de más complicados estados de mente puede evitarse escribiendo más símbolos en la cinta" Turing (1937), p. 350.

Como puede observarse, hay claramente una referencia analógica entre el computador y el ser humano; analogía que se muestra en diferentes aspectos, entre ellos: número de estados computaciones, número de estados mentales; símbolos que pueden ser analizados en un momento determinado, número de elementos que pueden ser observados, y así sucesivamente.

Comentando sobre estos aspectos, Shagrir (2006), p. 397, señala que estas consideraciones están motivadas por las "limitaciones de nuestro aparato sensorial, y, en particular, por el hecho de que el campo visual contiene un limitado número de símbolos o cuadros que una mente humana puede reconocer". Este autor resume el argumento de Turing de la siguiente manera:

"Premisa 1 (Tesis H): un computador humano satisface ciertas restricciones.

Premisa 2 (Teorema de Turing): Las funciones computables por un computador que satisfacen estas restricciones son máquina de Turing-computables.

Conclusión (Tesis de Turing): Las funciones computables por un computador humano son máquinas de Turing-computables", Shagrir (2006), p. 396.

Shagrir (2006) dedica su trabajo a analizar la aparente contradicción de Gödel respecto del tipo de mecanismo computacional propuesto por Turing. En efecto, por un lado, ya hemos hecho referencia a la opinión de Gödel en el sentido de que Turing había logrado satisfactoriamente formalizar el concepto intuitivo de 'efectivamente computable'. Pero por otro lado, Gödel en un trabajo posterior, publicado en 1972 y titulado "A Philosophical error in Turing´s work", critica el trabajo de Turing por basarse en supuestos oscuros.

La explicación de esta aparente contradicción la encuentra Shagrir justamente en la mediación humana utilizada por Turing en la conceptualización de estos dispositivos mecánicos. La generalidad de su aplicación es, como ya mencionamos, uno de los rasgos más extraordinarios de Turing: llevar al extremo del éxito el uso de una analogía. Claramente, Gödel tenía objeciones importantes a esta equivalencia mente humana-máquina, ya que para él, la mente humana es capaz representar y evaluar proposiciones matemáticas, más allá de lo que es efectivamente computable, y representar utilizando recursos de orden superior. En este sentido, la analogía claramente tiene sus límites. Más aún, Gödel objetaría el recurso analógico mismo: un procedimiento mecánico de lo que es efectivamente computable, debe proponerse sin tener que usar como punto de partida ningún otro concepto referente (mente humana). Realmente, el desarrollo posterior de la teoría de la computabilidad ha despojado a las máquinas de Turing este tipo de referencias analógicas de manera que puede ser estudiado por sí mismo.

La acotación anterior tiene como objetivo señalar la estrecha relación entre mente humana y máquina de Turing en este primer trabajo sobre computabilidad; correspondencia ésta que encontramos ya más explícitamente formulada en el artículo de 1950 "Computing Machinery and Intelligence". Este artículo como se recordará plantea la pregunta de si las máquinas pueden pensar. Claramente, él se refiere a las computadoras digitales, y no a las computadoras disponibles en el momento en el que escribe (1950), sino pensando más bien en las máquinas que estarían disponibles medio siglo después, es decir, en el año 2000. Esto motivó fuertemente el desarrollo de una nueva disciplina computacional conocida como 'inteligencia artificial' a partir de 1956, inicialmente con tesis muy fuertes, las cuales fueron suavizadas conforme se avanzaba en la compleja tarea de modelación computacionalmente de las múltiples dimensiones de la 'inteligencia humana' – pero que ha sido extraordinariamente fructífera para arrojar nueva luz sobre los procesos de inteligencia y sus límites.

La manera de abordar el problema de ¿pueden las máquinas pensar? es planteando una situación o juego en el que un agente humano interroga a una persona y un computador, sin saber previamente cuál es el humano y cuál la máquina. Si no logra distinguir entre humano y computador, podríamos decir, que la respuesta a la pregunta sería afirmativa.

Volviendo a nuestro tema de la relación entre mente humana y computador, Turing, señala lo siguiente: "La idea detrás de los computadores digitales puede ser explicada diciendo que estas máquinas están planeadas para realizar cualquier operación que pueda ser hecha por un computador humano. El computador humano se supone que sigue reglas fijas; no tiene potestad de desviarse de ellas en ningún detalle. Podemos suponer que estas reglas le son proporcionadas por un libro, el cual es alterado cada vez que es puesto a ejecutar una nueva labor. Puede además hacer un uso ilimitado de papel en el cual realizar sus cálculos. También puede hacer sus multiplicaciones y adiciones en una 'máquina de escritorio', pero esto no es importante". Turing (1950), p. 4.

Como se observa en esta referencia, Turing piensa las computadoras digitales en términos de máquinas humanas. Afirmación ésta que refuerza la argumentación de Shagrir (2006) que ya hemos mencionado. Pero hay más, los dispositivos principales que caracterizan las computadoras digitales también son pensados en términos de máquinas humanas: su memoria (almacenamiento), unidad ejecutiva y control, como se puede fácilmente verificar en el texto que comentamos.

En las objeciones, Turing analiza y anticipa la objeción de Gödel. Dentro de la lista de las objeciones que Turing analiza, ésta corresponde a la objeción matemática. Esta tiene que ver las limitaciones encontradas en los desarrollos de la lógica, en particular, las de Gödel en las capacidades de las máquinas. El argumento aquí es que los formalismos tienen limitaciones en la capacidad de responder a determinadas cuestiones que claramente el ser humano (computador humano) no las tiene. Siendo esto así, claramente la equivalencia entre ambas, que Turing busca establecer, falla. Lo que Turing argumenta es que este asunto, es de naturaleza empírica, y que la superioridad de la mente humana, ha sido hasta el momento respecto de máquinas específicas, no de todas las máquinas. Pero esto no solo no es homogéneo en el ser humano, tampoco lo es en las máquinas. "En breve, entonces, pueden existir humanos con mayor capacidad que cualquier máquina específica, pero también pueden existir otras máquinas más capaces, y así sucesivamente". En este sentido, parece que Turing se deslinda diferenciando entre los resultados formales, referidos a las propiedades de determinados sistemas formales y aquellos referidos al comportamiento de máquinas humanas y de Turing. Aquí la decisión es empírica y, por tanto, abierta en todo momento a nuevos desarrollos. Es decir, dicha equivalencia no es de naturaleza formal, y por tanto, debe ser decidido de manera empírica.

En relación pues, con la evolución del concepto de inteligencia, debemos decir, que ya en su primer trabajo hay una referencia analógica muy fuerte a entender las nuevas máquinas en términos de las capacidades de la mente humana. Esta analogía le permite seguir hablando de máquinas en ambos dominios. En este sentido, no hay mucho cambio en su primer trabajo y aquel publicado en 1950. Pero en este último trabajo, Turing explora las consecuencias de esta analogía para una audiencia que siempre ha encontrado problemática la equivalencia. Hay una importante novedad, nos parece: el alejarse de la equivalencia formal entre máquinas humanas y de Turing, planteando la necesidad de que el problema se decida de manera empírica. Claramente, si el problema fuera formal, Gödel había quebrado la equivalencia.

Bibliografía

Aleksandrov, A. D. (1982) "Geometrías no euclidianas". En Aleksandrov, Kolmogorov y Laurentiev. *La matemática: su contenido, métodos y significado* (5ª ed.), Madrid: Alianza Editorial.

Heijenoort (1967). *From Frege to Gödel: A Source Book in Mathematical Logic, 1879--1931.* Cambridge: Harvard University Press.

Hodges, Andrew, "Alan Turing", *The Stanford Encyclopedia of Philosophy* (Summer 2011 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/sum2011/entries/turing/>.

Kennedy, Juliette, "Kurt Gödel", *The Stanford Encyclopedia of Philosophy* (Fall 2011 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/fall2011/entries/goedel/>.

Iemhoff, Rosalie, "Intuitionism in the Philosophy of Mathematics", *The Stanford Encyclopedia of Philosophy* (Fall 2012 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/fall2012/entries/intuitionism/>.

Manna, Z. (1974). *Mathematical Theory of Computation.* New York: McGraw Hill.

Nagel, E. & Newman, J.R. (1958): Gödel's Proof. New York, London: New York University Press.

Shagrir, O. (2006): "Gödel on Turing on computability", en: A. Olszewski, J. Wolenski, R. Janusz (eds.), *Church's thesis after 70 years*, pp. 393-419. Frankfurt-Heusenstamm: Ontos-Verlag.

Soare, R. (2009): "Turing oracle machines, online computing, and three displacements in computability theory", *Annals of Pure and Applied Logic*, vol. 169, 3, pp. 368-399.

Turing, A. (1937). On Computable Numbers with and application to Entscheidungsproblem". *Proceedings of the London Mathematical Society*, serie 2, vol. 42, pp. 230-265.

Turing, A. (1950). *Computing Machinery and Intelligence*. *Mind* 49: 433-460.

Zach, Richard, "Hilbert's Program", *The Stanford Encyclopedia of Philosophy* (Spring 2009 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/spr2009/entries/hilbert-program/>.