Universidad Nacional
Autónoma de
México

Facultad de Ciencias

Los teoremas de Gödel en el dilema mente-máquina

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

M A T E M Á T I C O

P R E S E N T A:

JOSÉ GERMÁN AVILA VICENTEÑO

DIRECTOR DE TESIS:
DR. CARLOS TORRES ALCARAZ

2015

Ciudad Universitaria, D. F.

1. Datos del alumno
Avila
Vicenteño
José Germán
58 73 61 50
Universidad Nacional Autónoma de México
Facultad de Ciencias
Matemáticas
304253568

2. Datos del tutor
Dr.
Carlos
Torres
Alcaraz

3. Datos del sinodal 1
Dr.
Favio Ezequiel
Miranda
Perea

4. Datos del sinodal 2
Dr.
Osvaldo Alfonso
Téllez
Nieto

5. Datos del sinodal 3
Mat.
Guillermo Eduardo
Zambrana
Castañeda

6. Datos del sinodal 4
M. en C.
Rafael
Rojas
Barbachano

7. Datos del trabajo escrito
Los teoremas de Gödel en el dilema mente-máquina
60 p
2015

Dedicado a mis padres
Miguel y Norberta

Agradecimientos

Quiero agradecer a Dios; por bendecirme con todo lo que ha puesto en mi camino, por darme fuerzas para continuar y no rendirme ante las adversidades encarándolas sin desfallecer en el intento.

A mi familia por quienes soy lo que soy. A mis padres; por todo el apoyo, sus consejos, su comprensión y ayuda en los momentos difíciles, por haber creído en mi y ser la escalera que me impulso hasta este punto. A mis hermanos que han sido mi ejemplo a seguir y mi reto a vencer, que me han mostrado el camino y lo han caminado conmigo, por todas las cosas de las que me protegieron y por todo el cariño que siempre me han mostrado.

A mi asesor quien a pesar de todo el tiempo que transcurrió siempre me brindo su apoyo y confianza para que lograra mi objetivo.

A mis compañeros y amigos que me apoyaron y me permitieron entrar en su vida durante todos estos años. A Sergio quien a sido como otro hermano para mi. Gracias a Jaime, quien con el tiempo se convirtió en un gran amigo y apoyo en mi vida académica.

Especialmente y con mucho amor, gracias a Mindy, por animarme día a día para seguir luchando por esta meta.

Son muchas las personas que, de un modo u otro, han intervenido para que haya llegado hasta este momento, a todos ustedes gracias.

Índice general

Introducción

La noción de máquinas inteligentes, es decir, lo que llamamos inteligencia artificial, apareció en el panorama científico hace ya muchas décadas, al igual que las preguntas: ¿podría una computadora llegar a tener tal grado de capacidad como para imitar la mente y el comportamiento de los seres humanos?, ¿podría una máquina pensar?

Cuando se comienza a indagar en estos temas, se encuentra uno con enormes complicaciones, principalmente referentes a la definición del concepto de pensamiento. Esta definición por sí misma ha sido motivo de extensos estudios y discusiones durante muchos años, sin que hasta la fecha tengamos resultados concisos, por lo que abordarla nos desviaría del objetivo y no resolveríamos mucho.

Es por esa razón que nuestra pregunta inicial no es si una máquina puede o no reproducir el pensamiento humano, sino si puede imitarlo

De esta manera evitaremos caer en la discusión filosófica de qué es el pensamiento humano, que si bien es un tema sumamente interesante no es precisamente el ángulo que se busca analizar en este trabajo.

La idea de este trabajo es reunir y analizar diversos argumentos a favor y en contra de esta idea, en la cual el comportamiento de la mente humana se puede imitar en el dominio de las matemáticas por una máquina, en el sentido de Turing.

En el primer capítulo se revisará la obra de Alan Turing para tener una panorámica de la gran genialidad que le caractirizó toda su vida. Veremos que a medida que avanzaba en su trabajo poco a poco fue construyendo la idea de la mecanización de la mente humana hasta el punto de plantear una prueba que *midierá* la capacidad de las computadoras para imitar el comportamiento humano.

Revisaremos el planteamiento y estructura de su famoso *juego de la imitación* así como también los argumentos en contra de este y su respuesta para cada uno de ellos.

En el capítulo siguiente se expone la relación que guardan los sistemas formales con las máquinas de Turing a través de los famosos *Teoremas de incompletud* de Kurt Gödel y como a partir de ellos se estructura el argumento principal de este trabajo.

Se replantearán los teoremas de Gödel para las máquinas de Turing para determinar las límitantes que existen para estos objetos matemáticos.

Finalmente en el último capítulo estudiaré los argumentos planteados, no solo por varios científicos en la materia como Roger Penrose, J. R. Lukas y Gandy, sino también los puntos vista de los mismos Kurt Gödel y Alan Turing sobre la idea de poder mecanizar la mente humana.

Para finalizar este trabajo, después de exponer y analizar todos estos puntos de vista, se plantea una conclusión personal sobre la disyuntiva que si bien no busca dar una solución, muestra una posición en esta discusión basada en las pruebas del segundo capítulo .

Capítulo 1

La obra de Turing

"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".

Marvin Minsky

1.1. Panorámica

Para comenzar buscaré dar un pequeño vistazo al trabajo de Alan Turing, lo cual permitirá tener una perspectiva de su vida, su forma de ser y del increíble genio.

Veremos que su talento y habilidad le permitió incursionar en diversas áreas de la ciencia, a veces distintas ente sí, logrando grandes descubrimientos, importantes aportaciones e increíbles logros en cada una de ellas.

1.1.1. Teorema del límite central

Una de las áreas donde Turing trabajó fue la probabilidad. El nombre de Alan no acostumbra aparecer entre los grandes matemáticos que se citan con relación a la apasionante historia del teorema del límite central. Sin embargo, Turing presentó en 1934 una memoria, nunca publicada, titulada "On the gaussian error function" en la que, de forma independiente, obtenía una condición suficiente similar a la de Lindeberg de 1922 y anticipaba parte de los resultados posteriores de Feller, Lévy y Cramér del período 1935-1937.

Turing estudiaba matemáticas en el King's College de Cambridge, donde ingresó como becario en el otoño de 1931. Los estudios para graduarse en matemáticas se estructuraban en dos períodos, de uno y dos años respectivamente, con exáamenes obligatorios al final de cada uno. Además, para aquellos estudiantes que quisieran optar a ser becarios de investigación, como era el caso de Turing, el último año se completaba con un bloque adicional de materias muy avanzadas y la elaboración de una memoria en la que el candidato debía mostrar un trabajo de investigación original (Fellowship Dissertation).

Uno de los cursos del tercer año versaba sobre Metodología de la Ciencia y fue impartido por Sir Arthur Stanley Eddington. Entre los problemas tratados por el eminente astrofísico estaba el de justificar la tendencia de las medidas experimentales sujetas a errores de observación a seguir una distribución aproximadamente normal o gaussiana. Los argumentos del profesor Eddington no fueron suficientes para satisfacer a Turing quien se propuso demostrar con rigor el teorema del límite central. Era el otoño de 1933. Antes de finalizar el mes de febrero de 1934 Turing había concluido su trabajo, pero al intentar publicarlo se enteró de que Lindeberg ya había resuelto el problema 12 años antes y utilizando una técnica similar.

En cualquier caso, se le animó a presentar una versión del mismo, justificando la independencia de su trabajo. En noviembre de 1934 envía la memoria titulada "On the Gaussian error function" y el 16 de marzo de 1935 es elegido Fellow del King's College. Su vieja escuela de Sherborne lo celebra con medio día de fiesta. Al año siguiente, los chicos de Sherborne vuelven a disfrutar de otra media jornada festiva a cuenta de Turing, ya que la memoria fue galardonada con el prestigioso Smith's Prize por la Universidad de Cambridge.

Turing desarrollo tres teoremas en su trabajo, a continuación, se resume la notación y los resultados principales.

Si F es una función de distribución con medida μ y varianza σ^2, llamaremos función de distribución normalizada de F a la función $U(x) = F(\frac{x-\mu}{\sigma})$. Denotaremos por $H = F \oplus G$ a la convolución de F y G, $H(x) = \int_{-\infty}^{+\infty} G(x - t)dF(t)$.

Con estas notaciones Turing plantea el problema en los siguientes términos. Sea $\epsilon_1, \epsilon_2, \ldots$ una sucesión de errores independientes. Supongamos que el error ϵ_r tiene función de distribución G_r, función de distribución normalizada V_r, media nula y varianza σ_r^2. Definimos $F_n = G_1 \oplus \cdots \oplus G_n$ y sea U_n su correspondiente función de distribución normalizada. Entonces ¿bajo que condiciones $U_n \to \Phi(x)$ cuando $n \to \infty$, uniformemente en x? Cuando esto ocurra diremos que $F : n$ converge a la ley gaussiana.

En primer lugar, Turing proporciona sus condiciones "cuasi-necesarias", a saber,

$$\Sigma\sigma_r^2 \qquad \text{y} \qquad \{\tfrac{\sigma_n^2}{\Sigma_{r=1}^n \sigma_n^2}\} \to 0.$$

Turing afirma que: *"Rigurosamente, estas condiciones no son necesarias, pero si no se cumplen U_n sólo puede converger a Φ por accidente"*

Teorema 1.1.1 (Condiciones cuasi-necesarias)**.** *Con las notaciones previas:*

(1) Si $\Sigma\sigma_r^2$ es convergente y $F_n = G_1 \oplus \cdots \oplus G_n$ converge a la ley gaussiana entonces para cualquier distribución no gaussiana G_0 la sucesión $F_n^* = G_0 \oplus F_n$ no converge a la ley gaussiana.

(2) Si $\Sigma\sigma_r^2$ es divergente, $\{\frac{\sigma_n^2}{\Sigma_{r=1}^n \sigma_n^2}\}$ no tiende a 0 y $F_n = G_1 \oplus \cdots \oplus G_n$ converge a la ley gaussiana entonces dada cualquier sucesión de números naturales $\{m_s\}$ podemos encontrar una sucesión estrictamente creciente $\{n_s\}$ con $n_s \geqslant m_s$, de forma que la sucesión F_n^* dada por: $F_1^* = G_1$; $F_{n_s}^* = F_{n_{s-1}}^* \oplus H_{n_s}$, siendo H_{n_s} cualquier distribución no gaussiana con varianza $\sigma_{n_s}^2$; y $F_{n+1}^* = F_n^* \oplus G_{n+1}$ si $n+1 \neq n_s$ para todo s; no converge a la ley gaussiana.

Teorema 1.1.2. *Si F y G son funciones de distribución tales que F y $F \oplus G$ son gaussianas entonces G debe ser gaussiana.*

Teorema 1.1.3. *Si además de las condiciones (1) del primer teorema existe una función* $\varphi(x) > 0$ *tal que* $\lim_{|x|\to\infty} \varphi(x) = +\infty$ *y*

$$\underset{n\in\mathbb{N}}{sup} \int_{-\infty}^{+\infty} x^2 \varphi(x) V_n(x) < +\infty$$

entonces F_n *converge a la ley gaussiana.*

El primer teorema determina sus llamadas *condiciones cuasi-necesarias* de convergencia, el segundo habla sobre funciones gaussianas y su comportamiento con respecto de la convolución($\oplus$)[1] y el último determina una *condición suficiente* de convergencia a la ley gaussiana.

En definitiva, aunque la condición suficiente dada por Turing no es tan satisfactoria como la obtenida por Lindeberg 12 años antes, ciertamente comparte con esta sus dos características más celebradas: recurren a condiciones que involucran tan solo los momentos de orden dos y el método empleado en la demostración se basa en las funciones de distribución. Por contra, las condiciones cuasi-necesarias de Turing anticipan en un par de años el descubrimiento de Feller del fenómeno de las subsucesiones y su segundo teorema avanza el papel clave del teorema de Cramér.

1.1.2. Morfogénesis

Otra área en la que Turing también se interesó fue la morfogénesis. La morfogénesis es el proceso biológico que lleva a que un organismo desarrolle su forma. Este es uno de los tres aspectos fundamentales del desarrollo biológico junto con el control del crecimiento celular y la diferenciación celular.

Este proceso controla la distribución espacial organizada de las células durante el desarrollo embrionario de un organismo. La morfogénesis también puede tener lugar en un organismo maduro, en un cultivo de células o dentro de un tumor celular. La morfogénesis así mismo, describe el desarrollo de formas de vida unicelular que no atraviesan por una etapa embrionaria en sus ciclos de vida, o describe la evolución de una estructura corporal dentro de un grupo taxonómico.

[1] Operador entre funciones

Los primeros estudios morfogenéticos fueron hechos en la acetabularia[2]. Hasta hace un siglo, no se había comprobado que el núcleo celular contuviese información hereditaria o sobre el desarrollo. El control nuclear de la morfogénesis y la interacción del núcleo con el citoplasma fueron demostrados por J. Hämmerling en la primera mitad del siglo XX, utilizando dos especies distintas de acetabularia (A. acetabulum y A. crenulata).

Turing utilizó la computadora Ferranti Mark I de la Universidad de Manchester en 1952 para simular el mecanismo químico por el que los genes de un cigoto determinan la estructura anatómica de un animal o planta. Es fascinante notar que Alan relacionó este estudio con su trabajo sobre redes neuronales artificiales. Según él, la estructura del cerebro debería ser resultado de un mecanismo genético que controla su desarrollo.

Turing trabajó desde 1952 hasta que falleció en 1954 en la morfogénesis. Publicó un trabajo sobre esta materia titulado "Las bases químicas de la Morfogénesis" en 1952. Su principal interés era comprender la filotaxis de Fibonacci, es decir, la existencia de los números de Fibonacci en las estructuras vegetales.

De acuerdo con Turing, en un tejido en estado embrionario las células pigmentadas producirían dos moléculas llamadas morfogen. Estas moléculas reaccionan entre sí, difundiéndose a continuación en el espacio intercelular, resultando un patrón de concentraciones que inducirían la diferenciación de células pigmentadas y en consecuencia un patrón de células análogo al de las concentraciones. En el modelo de Turing la producción, difusión y decaimiento de morfogén está modelado por la siguiente ecuación diferencial en derivadas parciales:

$$\frac{\partial M_A}{\partial t} = f(M_A, M_I) + D_A \nabla^2 M_A$$

$$\frac{\partial M_I}{\partial t} = g(M_A, M_I) + D_I \nabla^2 M_I$$

De acuerdo con tales ecuaciones, la variación temporal de las concentraciones de dos especies químicas, denominadas por Turing morfogenes (M_A

[2]Las acetabularias son algas verdes unicelulares gigantes (de 0.5 a 10 cm de largo), uninucleares, marinas, con una forma característica de paraguas.

el morfogén activador y M_I el morfogén inhibidor), y la difusión de ambas, pero a diferentes tasas D_A y D_I (siendo estas distintas de cero) pueden dar lugar a patrones de tejido como resultado de su reacción de acuerdo con las funciones no lineales f y g.

El modelo de morfogenesis diseñado por Turing al final de su vida se sigue utilizando hoy en día.

1.1.3. Códigos, máquinas reales

Aunque la historia de la criptografía tiene antecedentes cientos de años antes de la Segunda Guerra Mundial (SGM), es en esta etapa cuando el uso de la criptografía se hace más común. En 1918 es patentada por Arthur Scherbius la máquina de cifrado *Enigma*, y se comienza a comercializar en 1922.

Esta máquina era muy parecida a una máquina de escribir, tenía distintas formas de configuración apartir de tres discos y otros componentes, teniendo la capacidad de generar 2,741,586 claves posibles. Para la SGM la versión de esta máquina fue modificada para alcanzar un total de 10^{114} configuraciones posibles.

Para fortuna de los aliados, en 1932, tras un golpe de suerte, el ejercito polaco logró hacerse con una de estas máquinas. Esto permitió descifrar los mensajes enviados por los alemanes a través de una máquina llamada *bomba*. No obstante en 1938 los alemanes incorporaron tres rotores más a la máquina Enigma con lo que serían necesarias 60 máquinas bombas para poder descifrar el código. Por desgracia el gobierno polaco no contaba con las posibilidades ni los recursos económicos para continuar con su trabajo.

En 1939 los servicios de inteligencia polacos se reunieron con sus homólogos británicos y franceses con lo cual la estafeta que había portado el ejercito polaco fue continuada a través de una institución denominada British Govermment Code and Cypher School (*BGC & CS*).

En septiembre de 1939, Turing se incorpora al *BGC & CS*, de Bletchley Park, para tratar de romper la *Enigma*. Turing usó diferentes formas de criptoanalizarla, desde ingeniería en reversa hasta el desarrollo de un complejo análisis estadístico, con lo cual, y aprovechando el trabajo antes realizado por los polacos, poco a poco logró descifrar mensajes. Mediante los procesos de decodificación creados por Turing se diseño y construyo la siguiente generación de máquinas rompecódigos que tomo el nombre de *bombe*. Esta máquina fue construida por Harold Keen, y pesaba cerca de una tonelada. El primer diseño fue bautizado con el nombre de *Victory* siendo instalada en Blenchley Park.

La mayor aportación de Turing fue la técnica llamada *Bandurismus*, la cual utiliza un análisis bayesiano para diseñar cartas llamadas *banduries* que finalmente ayudaban a la *bombe* para descifrar los mensajes [3].

Turing también logró romper la *Geheimschreiber*, máquina usada por Hitler y su alto mando. Después de esto en 1944 desarrollo la máquina de criptoanálisis llamada *Colossus*, considerada por muchos como la primera computadora.

1.1.4. Extensiones de la aritmética formal

En 1938 Turing dio un giro en sus investigaciones, orientándolas hacia la lógica, en particular hacia los teoremas de Gödel. Fue entonces que escribió el trabajo "Systems of logic based on ordinals" en el que considera el método

[3] Actualmente se denota como *ban* a la unidad de información $log_{10}p$

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", sistemas formales que se formarían siguiendo la sucesión ordinal con la que se intentaría minimizar el efecto del teorema de Gödel.

Dado un sistema L_1 que cumple con las condiciones de los teoremas de Gödel, este tiene un enunciado indecidible, denotemoslo por G_1, entonces podemos construir una sistema L_2 al agregar G_1 a los axiomas de L_1, así $L_2 = L_1 \cup \{G_1\}$

De esta forma, Turing compone una familia de lógicas ordinales L_α, una para cada ordinal numerable α. Decimos que α es un ordinal límite cuando 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 a un ordinal es irresoluble. A la vez, muestra cómo se podría escapar de las limitaciones que establece el segundo teorema de Gödel mediante ciertas máquinas denominadas "máquinas con oráculo".

1.1.5. Máquinas de Turing

Fue en este ámbito que Alan obtuvo el mayor reconocimiento y es por el cual suele ser mayormente recordado, además este es precisamente nuestro punto de interés.

Turing ejerció una influencia decisiva en el desarrollo de la computación al definir los conceptos de algoritmo y cómputo con la introducción de lo que hoy se conoce como máquinas de Turing y la idea de máquina con programa almacenado.

Si bien Hilbert y otros lógicos se habían apoyado en la idea de *procedimiento mecánico*, ni él ni sus seguidores intentaron aclarar 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 objeto que puede realizar de manera automatizada un procedimiento . 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.

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" [4].

1.1.6. Inteligencia Artificial

Ligada a la cuestión de qué es un algoritmo o procedimiento mécanico, Turing se preguntó si habrá procesos atribuidos únicamenta a los seres humanos que una máquina pudiera imitar. 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.

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

[4] A Logical Journey, Wang, 1996, p. 232

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.

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.

El test de Turing nació al extender esta cuestión a otros dominios: ¿Sería posible diseñar una máquina que pudiera no sólo 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? Más aún, ¿sería posible construir un máquina que tuviera sentimientos como los de una persona?

La idea de poder construir una computadora que pueda igualar las capacidades de los seres humanos ha generado a traveés del tiempo muchas opiniones tanto a favor como en contra, en el siguiente capítulo estudiaremos más a fondo las máquinas de Turing, así como la relación que guardan con los sistemas formales y los teoremas limitativos de Gödel.

Capítulo 2

Máquinas de Turing y teoremas de Gödel

"Nadie podrá salvarnos jamás del Infierno que Gödel demostró para nosotros".

Fernando Javier Nuñez Rosales

La relación que guardan entre sí las máquinas de Turing y los teoremas de Gödel es la parte medular de este capítulo. Comenzaremos con una revisión del concepto de *máquina de Turing* así como del concepto de *inteligencia artificial*, para después analizar cómo se relacionan con los teoremas de Gödel.

2.1. Máquinas de Turing

Una máquina de Turing es una idealización matemática de lo que es una computadora digital. Con esta noción se pretende capturar la idea de algoritmo o procedimiento efectivo. Aunque sumamente simples en su estructura, las máquinas de Turing, introducidas en 1936, poseen propiedades importantes. En particular son tan poderosas como cualquier otro dispositivo de cómputo imaginado hasta ahora. De hecho fueron la inspiración directa para la construcción de lo que hoy usamos cotidianamente como computadoras digitales. [1]

[1]El matemático John Von Neumann fue quien estableció la arquitectura fundamental de las computadoras que actualmente utilizamos, influido por las ideas de Turing.

Una máquina de Turing consiste de las siguientes partes:

a) Memoria. Una cinta dividida en celdas, una seguida de otra (organizadas linealmente). Cada celda contiene un símbolo de un alfabeto finito. El alfabeto contiene un símbolo especial (blanco) y uno o más símbolos. La cinta puede ser extendida indefinidamente, es decir, la máquina de Turing puede ser proveída de tanta cinta como vaya necesitando. Las celdas que no hayan sido escritas con anterioridad se asumen como en blanco, es decir, contienen el símbolo elegido como blanco. En cada momento de operación, sólo un número finito de celdas contienen símbolos distintos del blanco.

b) Una cabeza que puede leer y escribir símbolos en la cinta y moverse a la celda de la derecha o de la izquierda.

c) Un número finito de estados.

d) Programa de cómputo: Una función de transición o tabla de acciones que para cada estado y cada símbolo establece el símbolo a escribir, hacia donde mover la cabeza y el nuevo estado. Si no existe entrada en la tabla para una combinación de símbolo y estado entonces la máquina se detendrá.

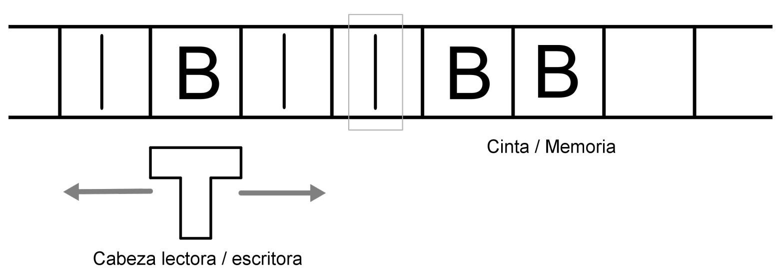


Figura 2.1: Esquema de una máquina de Turing

De manera formal una máquina de Turing se define como sigue:

Definición 2.1.1. *Una máquina de Turing MT es un modelo matemático que realiza cálculos en forma automática y consta de los siguientes elementos:*

$$MT = \langle Q, \Sigma, B, \delta, \rangle$$

donde

(1) Q*, es un conjunto finito de estados, del que distinguiremos un estado inicial* q_0 *y un estado final* q_F*.*

(2) Σ *es un conjunto finito de símbolos denominado alfabeto de máquina o de entrada.*

(3) B *es un símbolo para denotar al espacio en blanco. Junto con* Σ *forman el conjunto que llamaremos* Γ *denominado alfabeto de la cinta*[2] *.*

(4) $\delta : (Q \times \Gamma) \rightarrow (Q \times \Gamma \times \{I, D, \Lambda\})$*, es una función parcial denominada función de transición, donde I, D y* Λ *indican izquierda, derecha o reposo respectivamente. La función* δ *no está definida para parejas de la forma* $(q_F, \_)$*.*

Definición 2.1.2. *Sea x una cadena en la cinta de una máquina M, definimos* ***computo*** *de M al resultado que se obtiene de la máquina a través de* δ*, es decir, el computo de M con la entrada x es la cadena z que aparece sobre la cinta una vez que M llega a su estado final, y lo denotamos por*

$$M(x) = z$$

Definición 2.1.3. *La* ***producción*** *de una ḿaquina M se define como el conjunto de cómputos realizados por la máquina a partir del conjunto de expresiones* Γ^* *y se denota por*

$$Prod(M)$$

La función de transición puede representarse por medio de una tabla (tabla de transición de estados), dado que cada transición es de la forma:

(estado actual, símbolo leído) → (nuevo estado, nuevo símbolo, movimiento)

La tabla de transición es construida por las combinaciones de la función de transición, entrada por entrada, tomando como parámetros el estado actual de la máquina y el símbolo leído sobre la cinta, dando como resultado

[2]Denotaremos como Γ^* al conjunto de expresiones construidas a partir del conjunto Γ

el nuevo estado de la máquina, el nuevo símbolo a ser escrito sobre la cinta y la dirección para mover la cabeza lectora/escritora[3]

Uno de los ejemplos más simples de una máquina de Turing es aquella que suma dos números dados.

Ejemplo 2.1.4. *Para este ejemplo primero definiremos cómo representar números en la cinta. En principio, podríamos pensar que, al usar los símbolos B y $|$, podemos representar los números directamente en binario; sin embargo, la cantidad de operaciones necesarias para trabajar con tal representación haría que el número de estados de la máquina sumadora creciera sorprendentemente, debido a que es un sistema de numeración posicional.*

Para simplificar el proceso, escribiremos cada número por una cadena con tantos $|$ como indique el número; así, si tenemos el número k este será expresado por $\underbrace{|||\cdots|}_{k-veces}$. A esta forma de representación se le llama ***numeral*** *de k y se denota por $\bar{k}$.*

De esta forma sumar números equivale a concatenar sus representaciones. Si tenemos dos números en la cinta, separados uno de otro por un B, la forma más fácil de sumarlos sería convertir uno de los $|$ de los extremos en B, y cambiar el B de separación por un $|$. De este modo tendríamos una cadena formada por tantos $|$ como indica la suma de los dos números originales.

Sumemos 3 y 5

Cinta inicial:

$\boldsymbol{B\ B\ B\ B} \mid \mid \mid \boldsymbol{B} \mid \mid \mid \mid \mid \boldsymbol{B\ B\ B}$

Ponemos un B en el lugar del $|$ de la izquierda:

$\boldsymbol{B\ B\ B\ B\ B} \mid \mid \boldsymbol{B} \mid \mid \mid \mid \mid \boldsymbol{B\ B\ B}$

[3]En sentido estricto, hemos definido aquí una máquina de Turing determinista. Esto debe contrastarse con una no determinista, en la que hay una selección de varias acciones cuando la máquina lee un símbolo dado en un estado dado.

Ponemos un | en el lugar del B de separación:

B B B B B | | | | | | | | ***B B B***

El resultado: ocho |, el número 8.

Para implementar esto, supondremos que la cabeza de la máquina se encuentra en la celda con el primer trazo a la izquierda. Siempre se comienza en el estado q_0 *denominado estado inicial. La idea es unir las cadenas de | eliminando el espacio entre ellas conforma a la idea de la descripción anterior, borramos el primer trazo y lo repondremos al unir las cadenas.*

La siguiente tabla sería la correspondiente al proceso descrito:

	$\vert$	B
q_0	$B,\ q_1, D$	$B,\ q_0,\ D$
q_1	$\vert,\ q_1, D$	$\vert,\ q_2,\ D$
q_2	$\vert,\ q_2, D$	*0, **

La columna de la izquierda representa los estados de la máquina y la fila superior los posibles símbolos a leer.

Con la definición de máquina de Turing podemos construir otro concepto que utilizaremos más adelante, las llamadas *configuraciones instantaneas*[4].

Supongamos que en algún instante una máquina de Turing (MT) se encuentra en un estado q_i, que los símbolos en las celdas a la izquierda de la cabeza lectora escritora (CLE) forman una cadena $u \in \Gamma^*$ y que los símbolos a la derecha de CLE (incluido el de la celda donde se encuentra la cabeza) forman una cadena $v \in \Gamma^*$.

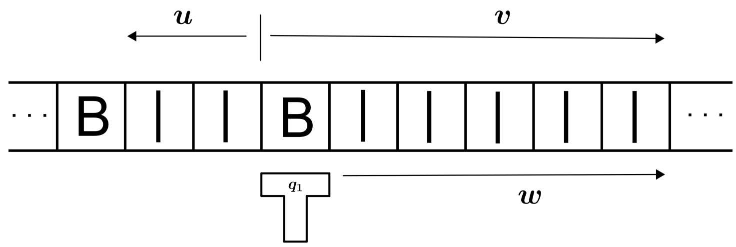
Además, supondremos que todas las celdas a la derecha de v o a la izquierda de u están en blanco. Definimos entonces la configuración instantánea de la MT por la terna (u, q_i, v), y los estados de transición de la MT pueden ser descritos por secuencias de estas ternas[5].

Definición 2.1.5. *Una configuración de MT será una terna* (u, q_i, v)*, donde* $u,\ v \in \Gamma^*$ *y* $q_i \in Q$

[4]Para uso practico nos referiremos a estas sólo como *configuraciones*

[5]Cuando una cadena sólo tenga B, escribiremos $u = B$ o $v = B$ según sea el caso.

Ejemplo 2.1.6. *Recordando el ejemplo anterior de la máquina que suma supongamos que nos encontramos en el momento que en la CLE se encuentra en la celda con el primer* B *después de los dos* $|$ *y la máquina está en estado* q_1. *De esta forma* u *representa la cadena* $||$ *y* v *la cadena* $B\,|||||$ *y llamaremos* w *a la cadena* $|||||$.



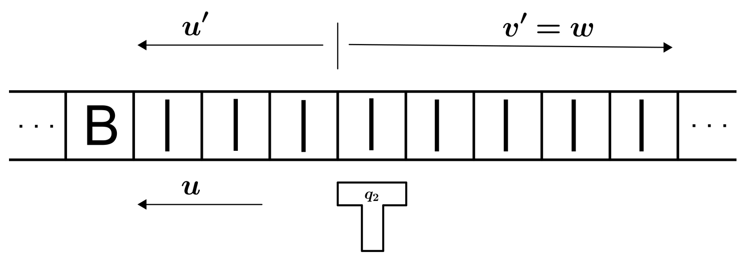
Lo que describimos puede representarse de la siguiente forma:

$$BBBB\,||\,B_{q_1}\,|||||\,BB$$

Al aplicar la función de transición tenemos que $\delta(q_1, B) = (q_2, |, D)$ *así las celdas en la cinta tendrían los símbolos:*

$$BBBB\,||||_{q_2}||||\,BB$$

Por lo cual la configuración (u, q_1, v) *pasa a ser* (u', q_2, v') *donde* $u' = 111$ *y* $v' = 11111 = w$



Existe otra forma de representar la función de transición de una máquina de Turing, y es mediante un diagrama de flujo. De esta forma los estados se representan por medio de nodos y cada una de las opciones de acción de la

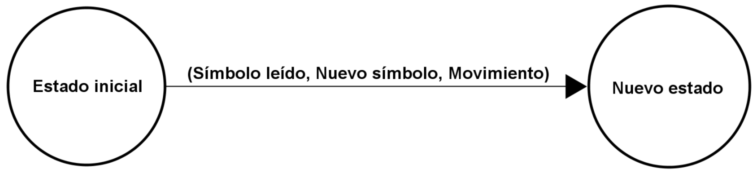


Figura 2.2: Esquema básico del diagrama de flujo de una MT

máquina se representan a través de flechas.

En la Figura 2.3 podemos ver cómo representaríamos el ejemplo de la máquina que suma mediante un diagrama de flujo.

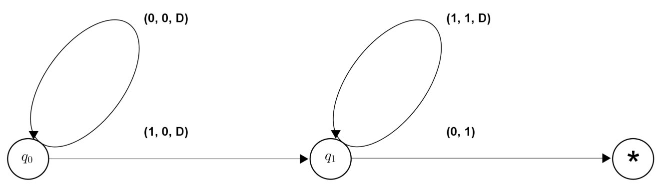


Figura 2.3: Diagrama de la función de transición

Es importante notar que cada parte de una máquina de Turing es finita. Sin embargo, su poder computacional, a diferencia de modelos más simples, se encuentra en la cinta y la posibilidad de extenderla tanto como se quiera, dándole una capacidad no acotada de espacio de operación y almacenamiento.

Con estos mecanismos tan sencillos es posible realizar cualquier cómputo que una computadora digital pueda realizar. Incluso este modelo abarca toda la funcionalidad de las supercomputadoras actuales. Lo que queremos decir es que no hay un solo cómputo que pueda ser ejecutado en una supercomputadora moderna que no pueda ser ejecutada por una máquina de Turing.

El valor de la máquina de Turing es que cuando examinamos qué tipo de problemas pueden ser resueltos por una computadora, sólo necesitaremos examinarlo en relación a las máquinas de Turing y viceversa. Esta doble implicación se sustenta sobre la *tesis de Church* o también llamada *tesis de Church-Turing.*

2.2. La tesis de Church-Turing

La llamada tesis de Church-Turing establece que las máquinas de Turing realmente capturan la noción de lo que es un algoritmo o un procedimiento efectivo. Originalmente[6] esta tesis se formuló de la siguiente manera:

Church-Turing: *Una función de enteros positivos es efectivamente calculable sólo si es recursiva.*

La mención de las funciones recursivas se debe al hecho de que toda función computable en el sentido de Turing es recursiva y viceversa.
Nosotros usaremos en general, la siguiente formulación:

Church-Turing: *Todo procedimiento efectivo, función o algoritmo computable es Turing-computable, es decir, computable por una máquina de Turing.*

Aunque se asume como cierta, la tesis de Church-Turing no puede ser probada, por eso es una tesis. Ello debido a que procedimiento efectivo y algoritmo no son conceptos dentro de ninguna teoría matemática y no son definibles sino a través de algún tipo de convención formal. La evidencia de su verdad es abundante pero no definitiva. Precisamente la tesis de Church establece que la definición de algoritmo o procedimiento efectivo es la de máquina de Turing.

Esta tesis cuenta con el apoyo del hecho de que todos los modelos o mecanismos computacionales propuestos para describir formalmente la noción de algoritmo han resultado ser equivalentes a la máquina de Turing, en el sentido de que lo que se puede hacer con ellos también se puede hacer con

[6] Alonzo Church, *A set of Postulates for the Foundation of Logic.* Annals of Mathematics, second series, 33, 346-366, (1936).

una MT adecuada, y viceversa. Entre los modelos de computación equivalentes a la máquina de Turing se pueden citar: las funciones parciales recursivas (modelo de Gödel y Kleene, 1936), el cálculo λ (modelo de Church, 1936), los sistemas de deducción canónica (modelo de Post, 1943), los algoritmos de Markov (modelo de Markov, 1951) y las máquinas de registro (modelo de Shepherdson-Sturgis, 1963).

Los lenguajes que son aceptados por una máquina de Turing son exactamente aquellos generados por todas las gramáticas formales. El cálculo λ, por ejemplo, es una forma de definir funciones. Las funciones que pueden ser computadas con el cálculo λ son exactamente las que pueden ser computadas con máquinas de Turing. Las máquinas de Turing, las gramáticas o lenguajes formales y el cálculo λ son muy distintos y fueron desarrollados de manera ajena uno del otro, sin embargo, son equivalentes pues tienen el mismo poder para resolver problemas. Esto generalmente se toma como evidencia a favor de la tesis de Church-Turing.

Por ejemplo, todo lo que se puede hacer utilizando las técnicas de computación evolutiva, los algoritmos genéticos, la programación genética o las estrategias evolutivas, e incluso las redes neuronales implementadas en computadoras digitales, se pueden llevar a cabo con máquinas de Turing[7]

Otras máquinas equivalentes reducibles a MT son las máquinas de Turing con más de una cinta, las máquinas de Turing con cintas n-dimensionales, las máquinas de Turing con un número limitado de estados y símbolos, los autómatas finitos con dos pilas, los autómatas finitos con dos contadores.

Así mismo todo lo que se puede hacer con la gramática formal, los sistemas de Post, el cálculo λ, las funciones recursivas parciales, los autómatas celulares (el juego de la vida de Conway o el autómata celular con una dimensión, dos estados, tres celdas por vecino y la regla 110), las máquinas de Turing probabilistas y las máquinas de Turing no deterministas resulta ser equivalente a una MT.

En otras palabras, lo que no se puede resolver con una máquina de Turing, no puede resolverse con una computadora cuántica y viceversa.

[7]Zenil Chavéz, Héctor, 2005.

Se ha acordado que un algoritmo consiste en un número finito y preciso de pasos descrito en un número finito de símbolos que podría ser ejecutado por un ser humano. En general, la ejecución de un algoritmo no requiere de mayor inteligencia que la necesaria para entender y seguir las instrucciones (incluso sólo seguir).

Los ejemplos de métodos efectivos o algoritmos son demasiados. Por ejemplo los algoritmos para la suma, resta, multiplicación o división en notación decimal. El algoritmo de Euclides para obtener el máximo común divisor de dos números naturales es otro ejemplo. Sin embargo, nada de esto constituye una definición formal, pues no es claro el significado de expresiones como *instrucción precisa* ni cuál es el tipo de inteligencia necesaria para seguir las instrucciones. Por esta misma razón, la idea de construir una máquina de Turing funciona como criterio para decidir cuándo algo es un algoritmo o procedimiento efectivo, es decir, la noción de máquina de Turing intenta capturar la esencia de lo que es un algoritmo o un procedimiento efectivo.

La tesis de Church-Turing ha sido tan exitosa que la mayoría la supone verdadera. Los términos derivados de ella, como método efectivo y computable, son comúnmente utilizados, cuando en realidad computable se refiere a Turing-computable. En el salto entre uno y otro se encuentra la tesis de Church.

La tesis de Church-Turing tiene profundas implicaciones. Cuando la tesis es aplicada a la física tiene diversos significados; como por ejemplo que el universo es una máquina de Turing y por lo tanto no es posible construir físicamente una máquina con mayor poder computacional o que compute funciones no recursivas. A esto se le ha llamado *tesis de Church-Turing fuerte*

Otra posible interpretación es que el universo no es una máquina de Turing, es decir, las leyes del universo no son computables lo cual no afecta la posibilidad de crear una máquina más poderosa que una máquina de Turing. Otra posibilidad es que el universo sea una hipercomputadora y entonces sea posible la construcción de máquinas más poderosas que las máquinas de Turing usando como entrada los resultados de dicha supercomputadora: el universo o la naturaleza.

Puede entenderse entonces que la tesis de Church-Turing establece un límite entre lo que puede ser computable (o Turing-computable) y lo que no, esto se conoce como el *límite de Turing*.

2.3. Inteligencia artificial

Tras la introducción del concepto de máquina computadora en 1936, con el tiempo fue surgiendo la idea de máquinas inteligentes. Fueron apareciendo preguntas como: ¿podrán las máquinas jugar bien al ajedrez?, ¿podrán realizar diagnósticos médicos de manera eficiente?, o preguntas más abstractas como ¿podrán las computadoras llegar a simular la mente humana?, *¿podrán las maquinas pensar?*

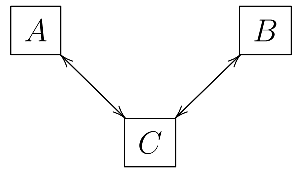
El estudio de esta cuestión debería comenzar con el análisis de la idea de lo que es una *máquina* y lo que es *pensar*. Al respecto, Turing elaboró un concepto matemático muy preciso de la primera de estas nociones, y evitó todo intento por definir la segunda. Esto lo hizo en virtud de que la búsqueda de un significado preciso de la palabra *pensar* ha sido motivo de extensos estudios y discusiones durante muchos años sin que hasta la fecha tengamos resultados concisos. Caer en esta trampa lo único que haría sería desviarnos de nuestro estudio. Así, Turing buscó otra manera de analizar este concepto.

Es por ello que, en lugar de preguntarse si una máquina puede o no pensar, lo que Turing propuso fue centrarnos en otra idea, que si bien pareciera distinta se basa en el mismo principio:

¿Puede una máquina imitar el comportamiento de la mente humana?

2.3.1. Formato inicial

La nueva forma del problema puede ser descrito en términos de un juego al cual llamaremos *juego de la imitación*. Originalmente se juega con tres personas, un hombre (A), una mujer (B), y un interrogador (C), que puede ser de uno u otro sexo. El interrogador se queda en un cuarto aparte frente a los otros dos.



El objetivo del juego para el interrogador(C) es determinar cuál de los dos es el hombre y cuál la mujer. Él los conoce por las etiquetas A y B, y al final del juego él debe decir "A es el hombre y B es la mujer" o "A es la mujer y B el hombre". Al interrogador se le permite hacer preguntas tanto a A como a B, cómo por ejemplo: "Sujeto A, dígame la longitud de su pelo".

El objetivo del hombre en el juego es confundir al interrogador haciendo que haga una identificación errónea. Por lo tanto, su respuesta podría ser: "Mi cabello está cortado en capas, y el más largo de mis cabellos es de cerca de 40 cm de largo". Con el fin de que los tonos de voz no puedan ayudar al interrogador, las respuestas deben ser escritas, o mejor aún, enviadas por un teletipo. Como alternativa, la pregunta y las respuestas se puede repetir a través de un intermediario.

El objetivo de la mujer es ayudar al interrogador. La mejor estrategia para ella es probablemente dar respuestas veraces. Se pueden añadir cosas tales como "!Yo soy la mujer, no le escuches!" como respuesta, pero no servirá de nada pues el hombre puede hacer comentarios similares.

Ahora la pregunta, ¿qué sucede cuando una máquina toma el papel del hombre en este juego? La cuestión es ¿se podrá programar una máquina de modo que el interrogador decida erróneamente, tan a menudo como lo hace cuando el juego se juega entre un hombre y una mujer? Estas cuestiones nos llevan a la pregunta central de nuestro estudio:

¿Pueden las máquinas imitar la mente humana?

2.3.2. Formato final

El nuevo problema tiene la ventaja de trazar una línea muy clara entre lo físico y las capacidades mentales de los seres humanos. No se aceptarán reclamaciones porque la ingeniería o la química no son capaces de producir un material indistinguible de la piel humana. Han sido muchas las objeciones sobre el planteamiento que hizo Turing, pero estas y la respuesta de Turing las revisaremos en el capítulo siguiente. La forma en la que Turing plantea el problema refleja este hecho en que el interrogador no pueda ver o tocar a los otros jugadores, o escuchar su voz [8].

Algunas otras ventajas del criterio propuesto se muestran en las siguientes preguntas y respuestas de ejemplo:

C: Por favor, escríbeme un soneto sobre el tema de la guerra de independencia.

A: No cuenten conmigo en esto. Nunca pude escribir poesía.

C: Agregar 34957 a 70764.

A: (Pausa de 30 segundos y luego dar como respuesta) 105,621.

C: ¿Juegas al ajedrez?

A: Si.

C: Tengo K en mi a1, y no hay otras piezas. Tu sólo tienes K en c2 y R en c3. Te toca mover ¿Qué harías?

A: (Tras una pausa de 15 segundos) R-a3, mate.

En mi opinión este método de preguntas y respuesta parece ser la manera más adecuada para la introducción de casi cualquier campo del quehacer humano que se desee incluir. Las condiciones de nuestro juego desestiman las capacidades físicas de los interrogados, pues el interrogador no puede exigir demostraciones prácticas.

Por otra parte, el juego tal vez puede ser criticado por quienes consideran que las probabilidades se colocan en exceso contra la máquina. Pero si el hombre fuera quien trata de hacerse pasar por la máquina, claramente se obtendrían resultados muy pobres y predecibles. Por lo tanto no tendría sentido invertir los papeles de A y B. ¿Pueden las máquinas llevar a cabo algo que puede ser descrito como pensamiento, pero que es muy diferente de

[8] A. Turing, “Computing machinery and intelligence”1950

lo que hace un ser humano?

Esta objeción es muy fuerte, pero al menos podemos construir una máquina que juegue el juego de la imitación de manera satisfactoria, por lo cual se asumirá que la mejor estrategia para la máquina es tratar de dar respuestas que daría un ser humano en forma natural.

2.4. Equivalencia de máquinas de Turing y Sistemas Formales

Esta sección es fundamental para este trabajo, puesto que a partir de ella se cimentará el argumento principal acerca de si la mente humana puede ser imitada por una máquina. El objetivo de esta sección es mostrar que todos los resultados limitativos que conocemos sobre los sistemas formales también son aplicables a las máquinas de Turing y por consiguiente a las computadoras, lo cual podremos ver mediante la siguiente equivalencia.

Proposición 2.4.1. *Si M es una máquina de Turing con alfabeto de cinta Γ y $K = Prod(M)$, es decir $K = \{z \mid M(x) = z \text{ con } x \in \Gamma^*\}$, entonces existe un sistema formal SF tal que $x \vdash_{SF} z$.*
Si SF es un sistema cuyo conjunto de teoremas es T, entonces existe una máquina M tal que $Prod(M) = T$

Para mostrar la equivalencia entre sistemas formales y máquinas de Turing revisaremos cada una de las implicaciones por separado, iniciando con una máquina y viendo que es posible construir un Sistema Formal (SF) que tenga como teoremas los cómputos de la máquina inicial. Recíprocamente, veremos que siempre es posible definir una MT que compute los teoremas de un SF dado.

2.4.1. De Máquinas de Turing a Sistemas Formales

Antes de iniciar con la primera de las implicaciones revisaremos algunas definiciones y conceptos que nos ayudarán en el proceso.

Definición 2.4.2. *Sea $v \in \Gamma^*$. Definimos $lend(v), rend(v) : \Gamma^* \to \Gamma$ de la siguiente manera*

$$lend(v) = \begin{cases} B \ \ si \ v = B \\ c \ \ si \ v = cw, \ \ c \in \Gamma, \ \ y \ w \in \Gamma^* \end{cases}$$

$$rend(v) = \begin{cases} B \ \ si \ v = B \\ c \ \ si \ v = wc, \ \ c \in \Gamma, \ \ y \ w \in \Gamma^* \end{cases}$$

Es decir, si v es una cadena no vacía sobre Γ, entonces $lend(v)$ es el símbolo más a la izquierda, y $rend(v)$ el símbolo más a la derecha de v

Definición 2.4.3. *Sean uq_iv y $u'q_jv'$ configuraciones de una máquina M. Diremos que $\delta^*(uq_iv) = u'q_jv' \iff u'q_jv'$ es la configuración de M en el momento de obtener $\delta(q_i, lend(v))$[9] .*

Definición 2.4.4. *Un sistema formal consiste de:*

1. *Un lenguaje formal $\mathcal{L}$, el cual consta de:*

   a) *Un conjunto a lo más numerable de símbolos (alfabeto).*

   b) *Un conjunto de reglas que determinen cuáles de las expresiones son fórmulas*[10] *.*

2. *Un conjunto decidible de axiomas que son subconjunto de las fórmulas de $\mathcal{L}$.*

3. *Un conjunto de reglas de inferencia. Dicho conjunto debe estar definido de modo que pueda indicarnos cuándo un fórmula llamada conclusión se infiere de otras fórmulas llamadas premisas. Las reglas son efectivas en el sentido de que dada una formula C y un conjunto $P_1, P_2, \ldots, P_m$ de fórmulas, podemos determinar si C se infiere de $P_1, P_2, \ldots, P_m$ con dichas reglas de inferencia.*

4. *Nociones de prueba y teorema. Una prueba es una sucesión finita de fórmulas tales que cada una de ellas es un axioma o se obtuvo mediante las reglas de inferencia a partir de anteriores. Una formula A será un*

[9] Para más detalles de este concepto revisar Bridges, Douglas (1994)

[10] Una expresión es cualquier sucesión finita de símbolos tomados de $\mathcal{L}$.

teorema si existe una prueba tal que la última fórmula que aparece en ella es A y se simboliza como:

$$\vdash A$$

A partir de estas nociones diremos que una deducción es una prueba que tiene como característica que parte de un conjunto Γ de fórmulas previamente establecidas llamadas premisas (distintas de los axiomas).

Ejemplo 2.4.5. *Un ejemplo sencillo de Sistema Formal es el que se plantea en el llamado "Acertijo MU".*

A este sistema formal se le da el nombre de Sistema MIU. El nombre del sistema se forma del hecho de que sólo emplea tres letras del alfabeto como lenguaje: M, I, U.

Para comenzar, el sistema MIU parte de una cadena inicial, la cadena MI. Las cadenas que sean producidas deberán conseguirse aplicando las reglas siguientes:

1. *Si se tiene una cadena cuya última letra sea I, se le puede agregar una U al final. Dicho en otras palabras si xI es un teorema, entonces xIU también lo es. En este caso x representa cualquier cadena arbitraria.*
2. *Supongamos que se tiene Mx. En tal caso podemos inferir Mxx.*
3. *Si en una de las cadenas aparece la secuencia III, esta puede ser sustituida por U.*
4. *Si aparece UU en alguna cadena está permitida su eliminación.*

Este sistema es muy sencillo lo que nos permite ver de forma clara cada uno de los elementos de lo conforman, su lenguaje, su único axioma y sus reglas de inferencia.

Como dato extra, el acertijo de este sistema es determinar si MU es un teorema del sistema.

Una vez terminados estos preliminares procederemos entonces.

Proposición 2.4.6. *Dada una máquina de Turing M, podemos construir un SF el cual tiene como conjunto de teoremas la totalidad de los cómputos de M.*

Demostración.

Sea M una máquina de Turing, Q el conjunto de estados en el que incluiremos al estado inicial q_0 y el estado final q_F, y sea δ su función de transición.

Construiremos un sistema formal a partir de los elementos de la máquina con el propósito de que nuestro sistema pruebe exactamente aquellas configuraciones que la máquina puede producir. Para esto determinaremos cuales son cada una de las partes que conformarán nuestro SF.

1. El lenguaje formal de SF ($\mathcal{L}_{SF}$) estará conformado por lo siguiente:
   a) El conjunto de símbolos estará dado por el conjunto Γ y el conjunto Q.
   b) Las fórmulas de $\mathcal{L}_{SF}$ serán todas las configuraciones de la máquina de la forma uq_iv.
2. En este SF el conjunto de axiomas será vacío [11]. Diremos entonces que una configuración es una premisa o hipótesis en SF si es de la forma uq_0v
3. La regla de inferencia estará dada por la función de transición δ. Es decir, una configuración $u'q_jv'$ se infiere de la configuración uq_iv si $\delta(uq_iv) = u'q_jv'$. Nótese que la función de transición puede definirse también sobre las configuraciones a través de la tabla de transición[12].
4. Una deducción en este sistema será un conjunto de configuraciones $C_1, C_2, \ldots, C_n$ tal que cada una de ellas se obtiene mediante las reglas de inferencia a partir de la anterior. Definiremos que la cadena de símbolos $u'v'$ es un teorema, si a partir de una configuración $uq_{_0}v$ obtenemos mediante una deducción la configuración $u'q_{_F}v'$, es decir

[11]Por ello en lugar de prueba utilizaremos sólo la noción de deducción

[12]Para más detalles de la aplicación de δ a las configuraciones ver Bridges, Douglas (1994).

$$M(uv) = u'v' \Longleftrightarrow u, q_0, v \vdash u'q_F v'$$

De esta forma, tenemos un SF que deduce todo aquello que M produce a partir del estado q_0.

□

2.4.2. De Sistemas Formales a Máquinas de Turing

Ahora veremos que dado un sistema formal podemos construir una máquina que produce los mismo teoremas.

Proposición 2.4.7. *Dado un sistema formal SF, hay una máquina M_i tal que $\varphi_i(0), \varphi_i(1), \varphi_i(2), \ldots$ es una enumeración de los teoremas en SF.*

Demostración.

Lo primero que debemos hacer es gödelizar el SF. Este procedimiento nos permitiría traducir el lenguaje del SF y su sintaxis a expresiones puramente aritméticas (recursivamente)[13].

Como sabemos, la aritmetización de Gödel permite asignar un número a cada prueba del SF y, además, sabemos que la noción de *"ser el número de Gödel de una prueba"* es decidible (recursivamente).

Esto significa que hay una máquina de Turing G tal que aplicada a la sucesión natural $0, 1, 2, 3, ...$, determina una sucesión $G_0, G_1, \ldots$ que son los números de Gödel consecutivos de pruebas de SF.

De esta forma si n es el k-ésimo número de Gödel de una prueba, entonces mediante otra máquina G' podemos obtener de forma recursiva el número $(n)_{long(n)}$ (el ultimo exponente distinto de cero del número n) el cual será el número de Gödel de un teorema y eso es el producto de MT.

La existencia de estas máquinas se sostiene sobre el hecho de que los procedimientos que se requieren para encontrar tales números se compone

[13]Preizer R., Ma. de la Asunción (2003)

de operaciones que sabemos son turing-computables (distinguir entre par e impar, factorizar en primos,...), por lo que la combinación de estos es turing-computable.

Finalmente podemos construir otra máquina M_i con base en G y G' que una vez aplicada a la sucesión de los números naturales nos arroje como resultado la sucesión de los números de Gödel de los teoremas de SF.

$$x \longrightarrow \boxed{M_i} \longrightarrow \varphi(x)$$

Por lo tanto, M_i produce exactamente la lista de los teoremas del SF dado, es decir, n es el número de Gödel de un teorema en SF si y sólo si $M(x) = n$ para algún $x \in \mathbb{N}$.

$\square$

2.5. Teoremas de Gödel (MT)

Kurt Gödel es considerado uno de los matemáticos más importantes del siglo pasado por su trabajo realizado en lógica matemática y sus teoremas de incompletud.

En su momento, tanto Russell como Hilbert pretendieron mostrar que las matemáticas podían reducirse a una teoría completa y consistente, pero con los teoremas de Gödel sus ambiciones se vieron truncadas, aunque finalmente se vio que esto no fue otra cosa que un enorme paso para hacer crecer las matemáticas, abriendo múltiples caminos de investigación.

Apoyandonos en la equivalencia anterior podemos entonces enunciar los teoremas de Gödel tanto para sistemas formales como para máquinas de Turing.

2.5.1. Primer teorema de incompletud

El primer teorema habla sobre el hecho de que para todo sistema formal para la axiomática de Peano de primer orden consistente hay un enunciado el cual ni él ni su negación pueden ser demostrados, es decir que es indecidible. En otras palabras el teorema dice lo siguiente.

Teorema 2.5.1 (Gödel-Rosser)**.** *Si la Axiómatica de Peano* (AP) *es consistente, entonces podemos construir un enunciado* $G \in \mathcal{L}_{AP}$ *tal que* $\nvdash_{AP} G$ *y* $\nvdash_{AP} \neg G$[14]

Este teorema puede extenderse a cualquier sistema formal de primer orden que contenga a AP, dando por consecuencia el siguiente corolario.

Corolario 2.5.2. *Cualquier extensión de AP consistente y recursivamente axiomatizable posee un enunciado indecidible*[15]*.*

Con la equivalencia entre los sistemas formales y las máquinas de Turing puede formularse el primer teorema con relación a las máquinas.

Antes habremos de dar un par de definiciones más para entender lo que significa *consistencia* y *completud* para máquinas de Turing.

Definición 2.5.3. *Diremos que una MT es consistente si en su producción no encontramos resultados contradictorios*[16] *.*

Definición 2.5.4. *Se dice que una MT es completa si para toda entrada* $x \in \Sigma^*$ *la máquina siempre se detiene.*

Una vez dado lo anterior el primer teorema de Gödel para máquinas de Turing es el siguiente.

Proposición 2.5.5. *No existe una MT tal, que produzca todos los enunciados de la aritmética recursiva y que su producción sea un conjunto completo y consistente de enunciados que incluye a los teoremas de la aritmética de Peano.*

Demostración.
Supongamos que existe una máquina cuya producción es consistente, y completa. Esto implicaría que a partir de ella podemos construir un sistema formal consistente y completo que contenga a AP, lo cual contradiría el primer teorema de Gödel.

[14]John Barkley Rosser estudio bajo la tutela de Church y probo esta versión del primer teorema en 1936 (Rosser's trick)

[15]Una teoría de primer orden T es recursivamente axiomatizable si el conjunto de números de Gödel de los axiomas es un conjunto recursivo.

[16]Estamos hablando de máquina de Turing para la aritmética, es decir, cuyo lenguaje es de primer orden e incluye el operador negación.

Por lo tanto no existe tal máquina.

□

Otra forma de ver esta proposición es a través de lo demostrado por Turing al determinar que no puede construirse una máquina que resuelva el problema de la decisión, es decir, que determine si una máquina M_i se detiene con un conjunto de datos $n_1, \cdots, n_k$.

2.5.2. Segundo teorema de incompletud

El llamado "Segundo teorema de incompletud", nos muestra que en AP, al construir un enunciado Con_{AP} que exprese su consistencia, a través de la númeración de Gödel, la fórmula en cuestión no puede ser probada con las herramientas propias de AP. Dicho de otra forma tenemos lo siguiente.

Teorema 2.5.6. *Si AP es consistente entonces* $\nvdash_{AP} Con_{AP}$

El punto crucial dentro de la demostración de este teorema descansa en la veracidad de este otro teorema.

Teorema 2.5.7. *En AP es un teorema la fórmula* $Con_{AP} \to G$*; es decir,*

$$\vdash_{AP} Con_{AP} \to G$$

donde G es el enunciado de Gödel para AP.

Podemos encontrar una prueba de este resultado consultando Feferman (1960).

El segundo teorema de Gödel puede ser generalizado a cualquier sistema formal de primer orden(SF) que sea consistente, que contenga a AP y sea recursivamente axiomatizable.

Sin embargo, como Feferman lo hizo notar, para realizar esta generalización es necesario ser muy precisos al tratar con afirmaciones relativas a la consistencia, pues en sus notas prueba la consistencia de ciertos sistemas formales dentro de si mismos[17].

De esta forma podemos enunciar una forma generalizada a continuación.

[17] Vease Feferman (1960), teorema 5.9 y corolario 5.10

Teorema 2.5.8. *Sea SF un sistema formal recursivamente axiomatizable y que contiene a AP, si SF es consistente entonces*

$$\nvdash_{SF} Con_{SF}$$

Nuevamente apoyándonos en la equivalencia SF-MT, se interpreta a continuación el teorema de Gödel para máquinas de Turing.

Debemos aclarar que en conformidad con la aritmetización de Gödel, es posible aritmetizar también a las MT, y de esa forma construir un enunciado aritmético Con_{MT} que afirma que la MT es consistente.

Ahora enunciare el segundo teorema aplicado a máquinas de Turing.

Proposición 2.5.9. *Dada una MT que produzca todos los enunciados de la aritmética recursiva, si ésta es consistente, entonces el enunciado "MT es consistente"que afirma tal cosa no es parte de la producción de MT, es decir*

$$Con_{MT} \notin Prod(MT)$$

La demostración es análoga a la proposición anterior buscando contradecir el segundo teorema mediante la equivalencia MT-SF.

Por último, no está de más recalcar el hecho de que estos resultados nos permiten asegurar que todas las características de los sistemas formales están presentes de la misma forma en las máquinas de Turing, y que éstas presentan los mismos límites que tenemos para los sistemas formales. Esto nos permitirá sustentar varios de los argumentos del siguiente capítulo.

Capítulo 3

El computabilismo

"Sobre hombre y máquinas...
Hubo un tiempo en el que los hombres soñaron en que algún día las máquinas serían capaces de calcular y pensar como ellos lo hacían...
Hubo un tiempo en el que hombres y máquinas emprendieron el mismo camino...
Hubo un tiempo en el que hombres y máquinas compartieron los mismos anhelos...
Hubo un tiempo en que las máquinas soñaron que algún día serían capaces de calcular y pensar como lo hacían los hombres...
Pero hubo un tiempo en el que las máquinas sonãron en que algún día otras máquinas serían capaces de calcular y pensar como ellas lo hacían...
Hubo un tiempo en el que hombres y máquinas creyeron estar creados a imagen y semejanza de un mismo dios...
Hubo un tiempo..."

Rafael Lahoz-Beltrá 2005

El computabilismo es una tendencia filosófica surgida en la primera mitad del siglo XX, la cual tiene dos formas de expresarse. En su sentido más fuerte, el computabilismo sostiene que el cerebro y la mente humana funcionan básicamente como una computadora, por lo cual puede ser representada por una. En un sentido más débil el computabilismo sostiene que la mente humana puede ser simulada por una máquina.

Mucho tiempo y recursos han sido utilizados en la discusión alrededor de la veracidad del computabilismo débil. En este capítulo revisaremos algunos argumentos que importantes personalidades han dado a favor y en contra de este concepto.

3.1. Antecedentes

Es poco común pensar que la idea de mecanizar el razonamiento humano tenga sus inicios en épocas tan anteriores al desarrollo de las máquinas de cómputo, pero el hecho es que estas máquinas tienen su origen en ideas establecidas previamente que tenían el mismo objetivo.

Solemos recordar a Leibniz por sus aportes a las matemáticas y a la filosofía, pero en pocas ocasiones lo hacemos por sus intentos de crear un lenguaje universal que sirviera de medio para la deducción y el descubrimiento en ciencia y filosofía.

A este lenguaje universal al que dio el nombre de *Characteristica Universalis*, lo consideró perfecto para representar conjuntamente las ideas y la mecanización del razonamiento filosófico.

La idea de crear un lenguaje para la ciencia tiene una larga historia. Raimundo Lulio propuso un lenguaje con el cual poder encasillar los razonamientos, Galileo propuso el lenguaje de las matemáticas para entender el mundo y sus leyes. A fines del siglo XIX, Giuseppe Peano usó un lenguaje exclusivo para escribir las proposiciones matemáticas y finalmente en los años treinta del siglo pasado Rudolph Carnap propuso la creación de una *lingua franca*, a través de la cual los filósofos y científicos pudieran entenderse[1].

Actualmente en pleno siglo XXI, Stephen Wolfram (el creador de *Mathematica*) ha propuesto un nuevo lenguaje para la ciencia: el *Lenguaje Wolfram* un lenguaje computacional con contenido intrínseco de amplio espectro, con estructuras y algoritmos incorporados. Este es un lenguaje simbólico, de principios claros, capaz de describir estructuras arbitrarias buscando automatizar lo más posible todo aquello que hace.

Leibniz se inspiró para su sistema en los lenguajes chino y egipcio; para él, estos lenguajes eran el primer paso hacia su *Characteristica Universalis*,

[1] Pareja H. (2013)

pues llevan directamente en la escritura su interpretación semántica. En otros términos, el significado del ideograma es inmediato y no depende de lo que quiso decir el escritor, ni de cómo lo va interpretar el lector.

Junto a este lenguaje universal, Leibniz también imaginó un cálculo especializado para el razonamiento filosófico el cual serviría, al igual que el lenguaje, para unificar la forma en que se desarrollan las deducciones.

Dentro de este enfoque universalista Leibniz escribió su *Ars Combinatoria*, donde los primeros ejemplos de este cálculo los toma del derecho, el registro musical del órgano y la teoría aristótelica de generación de elementos a partir de las cuatro calidades primarias: agua, tierra, aire y fuego.

Las aplicaciones filosóficas de esta obra son extraordinarias, pues fija las bases para la creación de un sistema automatizado de razonamiento. Allí cita la idea de Thomas Hobbes de que todo el razonamiento no es sino computación.

3.2. Turing y sus seguidores

La intención de Turing de crear una *maquinaria inteligente*, lo llevó a programar la computadora MADAM de la Universidad de Manchester para que fuera capaz de escribir una carta de amor, lo cual realizó con éxito. Los resultados de esta clase de experiencias fueron recogidos en un artículo publicado en 1950 bajo el título de *“Computing machinery and intelligence”*. Desde entonces la posibilidad de crear una máquina que haga tareas inteligentes ha sido respaldada y secundada por varias personas, entre ellas Robin Gandy[2].

A continuación revisaremos las ideas y argumentos con los que Turing plantea los principios del computabilismo, así como el seguimiento que dan a esta idea Robin Gandy y Andrew Hodges.

3.2.1. Turing

El fin de la II Guerra Mundial y el ambiente científico de la época animó a Turing a continuar con sus ideas sobre la mente humana. En este sentido el

[2] Véase R. Gandy, “Human versus Mechanical Intelligence”, pag. 135.

modelo imperante del cerebro humano por aquel entonces, resultado de los conocimientos acumulados en neurofisiología, así como el modelo conductista con el que se trataba de explicar la mente, pudieron influir en las ideas que condujeron a Alan a desarrollar y publicar su *test*.

Turing pensaba que el cerebro humano era en realidad una máquina con un número finitos de estados y, por consiguiente, una computadora digital, lo que efectivamente estaría relacionado con su célebre máquina. La hipótesis formulada por Turing conjeturaba que el cerebro, y en particular el córtex, es una máquina desorganizada en el momento del nacimiento de un sujeto, organizandose en una máquina universal a través del aprendizaje a medida que el sujeto crece y se desarrolla hasta la edad adulta.

Como Turing bien sabía, sus conceptos radicales y extraordinarios provocaron, de inmediato, una reacción enorme y contraria a su trabajo. Claro que hubo (y hay) objeciones razonables y objeciones completamente absurdas.

Entre los principales argumentos que revisaremos se encuentran el teológico, que habla de la existencia del alma; el biológico, que refiere a la propiedades del sistema nervioso y el social, tomando en cuenta "la conciencia".

Argumento Teológico

Frente a Turing algunos argumentaron que el pensamiento lógico es una parte esencial de la funcionalidad del alma inmortal que Dios había infundido al Hombre; Dios le ha dado un alma inmortal a todos los hombres y mujeres, pero no a cualquier otro animal o en este caso a las máquinas. Por lo tanto, ningún animal o máquina puede pensar.

Alan piensa que el argumento citado anteriormente implica una grave restricción de la omnipotencia del Todopoderoso. Se estaría admitiendo que hay ciertas cosas que él no puede hacer, pero ¿qué no creemos que haya libertad para conferir un alma a un elefante si lo considera conveniente? Podemos esperar que él sólo ejerce este poder en conjunto con una mutación que proporcione al elefante un cerebro adecuado para servir mejor a las necesidades de este tipo.

Un argumento de forma exactamente igual se puede hacer para el caso de las máquinas. Puede parecer diferente porque es más difícil de “digerir”, pero en el intento de construir máquinas pensantes no estamos usurpando irreverentemente el poder de la creación de las almas, más de lo que lo estamos en la procreación de los hijos: ¿No somos, en cualquier caso, instrumentos de la voluntad divina para proporcionar vida?.

Cuando se le expusieron estos razonamientos a Turing, el matemático respondió (echando mano inteligentemente a los atributos de Dios) que si Dios era omnipotente, no representaba ningún problema para Él infundir un alma inmortal en una máquina si se le antojaba. Las restantes objeciones fueron demolidas por el propio Turing, incluso, anticipándose a sus críticos, en el mismo documento original de 1950.

Este aspecto del alma esta ligado al concepto de consciencia, ya que expresa que la inteligencia requiere de la autoconciencia para existir. ¿Es esto así? ¿es autoconsciente una colonia de hormigas? La única manera de saberlo sería siendo esa colonia misma, lo que equivale al argumento del solipsismo, esto es, “lo único de lo que **yo** puedo estar seguro es de la existencia de mi propia mente”(solamente yo existo).

Turing escribió que esta objeción es absurda y debe ser abandonada en pro de la corrección y de la amabilidad, asumiendo que todos los organismos piensan. Lo mismo se hace extensivo a las computadoras (piensan a su manera). Es suficiente con que la computadora aprenda, piense y sienta a la manera de un loro, escribe Turing, y que esas características se manifiesten en conversaciones más o menos simples como las que yo mantengo con mi ave. Turing reconoce, sin embargo, que el tema de la conciencia no es trivial; sólo dice que no es imprescindible adentrarse en sus misterios para discutir acerca del pensamiento humano o animal, y ni siquiera para estudiar la inteligencia artificial.

Argumento Biológico

Los adeptos a este argumento afirman que la máquina nunca podrá emular al cerebro humano porque el impulso nervioso es un fenómeno continuo (analógico) que no puede reproducirse por un sistema discreto (digital). El sistema nervioso no es ciertamente una máquina de estados discretos.

Una pequeña variación en el tamaño de la información en un impulso nervioso que incide en una neurona, puede hacer una gran diferencia al tamaño del impulso de salida. Siento tan delicado este procedimiento, no podemos esperar ser capaces de imitar el comportamiento del sistema nervioso con un sistema de estados discretos.

Turing responde a este argumento diciendo que es cierto que una máquina de estados discretos debe ser diferente de una máquina continua. Pero si nos adherimos a las condiciones del juego de imitación, el interrogador no será capaz de distinguir la diferencia.

Argumento Social

Este argumento plantea que las consecuencias de que las máquinas pudieran pensar serían demasiado terribles para los seres humanos, pues nos gusta creer que el hombre es de algún modo superior al resto de la creación, lo cual es mejor si podemos demostrarlo, porque entonces no corremos peligro de perder nuestra posición dominante. La popularidad del argumento teológico esta claramente relacionada con este sentimiento.

Es probable que esta idea sea bastante fuerte entre la gente intelectual, ya que valoran el poder del pensamiento más que otros, y son más proclives a la base de su creencia en la superioridad del hombre en este poder. Para Turing, este argumento no es lo suficientemente importante (o de diferente aportación) para requerir refutación.

3.2.2. Robin Gandy y Andrew Hodges

"Todo lo que puede ser computado por una máquina puede ser computado por una máquina de Turing."

Lo citado anteriormente es la llamada "Tesis M", llamada así por Gandy en 1980, la cual es más amplia que la tesis C-T y tiene dos posibles interpretaciones:

1. Si tomamos "puede ser computado por una máquina" como "puede ser calculado por una máquina que siga leyes físicas del mundo real o tenga

una limitación de recursos", no se ha demostrado si esta tesis es cierta o falsa.

2. Si no ponemos como requisito que la máquina pueda existir realmente, la tesis es simplemente falsa. Hay varias máquinas teóricas que computan funciones que una máquina de Turing no puede computar. En el 2001, por ejemplo, Tien D. Kieu propuso un algoritmo hipercomputacional cuántico, el cual emplea como referente físico el oscilador armónico cuántico y resuelve en principio el décimo problema de Hilbert[3].

Dependiendo del autor que se lea, esta tesis es equiparable a la de Church-Turing o no. B.J. Copeland afirma que la tesis de Church-Turing es cierta independientemente de si la tesis M lo es o no. Según él, la tesis de Church-Turing se refiere a computación por parte de un humano sin ayuda de máquinas y la tesis M a máquinas propiamente dichas. Copeland cita como ejemplo que podría haber, entre el repertorio de operaciones atómicas de una máquina, operaciones que ningún ser humano podría realizar. Por tanto, las tesis no serían equiparables.

En cambio, Hodges indica que las dos tesis son prácticamente equivalentes, que sólo las diferencia la formulación de la tesis. Al respecto, Hodges afirma que hay dos motivos por los que Turing formuló originalmente la tesis refiriéndose a humanos y no a máquinas:

1. La tesis fue escrita en respuesta al *Entscheidungsproblem*, que estaba planteado en términos de lo que podían calcular los matemáticos, no las máquinas.

2. En 1936, los seres humanos podían realizar tareas mucho más complicadas que cualquier máquina conocida.

Además, Hodges indica que en los artículos de Copeland se habla de máquinas con "poderes sobrehumanos"que pueden computar funciones que los humanos no, mientras que la idea original de Turing era más bien al contrario: Turing se preguntaba si las máquinas podrían llegar a computar lo

[3]T. D. Kieu. *"Quantum adiabatic algorithm for Hilbert's tenth problem: I. The algorithm".*

mismo que los humanos.

Habiendo sido alumno de Turing, Gandy es uno de los más importantes defensores de la teoría del computabilismo. En muchas de las ocasiones en que se busco negar la validez de esta teoría, Gandy dió una respuesta para superar el obstáculo. Tal es el caso de la discusión que mantuvo con Roger Penrose sobre la llamada *"intuición"*[4].

Según Penrose, la intuición es una de las características de la mente humana que la hace diferente con las máquinas, ya que argumenta que esta habilidad, también llamada *"chispa divina"*, es la que permite a los matemáticos el poder encontrar nuevas ideas y caminos para el desarrollo de su disciplina. Esta habilidad es tan fugaz y aleatoria que la hace imposible de imitar por medio de una máquina.

Gandy responde que esta habilidad bien puede ser imitada a través de una máquina que trabaje con procesos aleatorios que le permitan, mediante una especie de *"lluvia de ideas"*, crear proposiciones aleatorias de las que discrimine aquellas que tengan sentido en la teoría.

Tanto Gandy como Hodges pueden ser considerados como los herederos académicos de Alan Turing, y de la misma forma que él, ellos consideran posible que en un futuro la capacidad de las computadoras llegará a compararse con la de los seres humanos para de este modo imitar a la perfección nuestro comportamiento y el funcionamiento de nuestra mente.

3.3. Lucas, Penrose y Gödel

A pesar de los argumentos dados por Turing y sus seguidores, otros pensadores sostienen con buenos argumentos la idea de que hay tareas que la mente humana puede realizar de modo que se distingue en ello de las máquinas de Turing. Entre estos se encuentran J.R. Lucas, Roger Penrose y el mismo Gödel de quien se conocen muchas opiniones la en materia gracias a las notas recolectadas por Hao Wang.

[4]Pensrose (1989)

A continuación presentamos los argumentos de cada uno de ellos, así como las respuestas que ofrecen a algunas de las objeciones que se les plantean.

3.3.1. Lucas

En 1961, Lucas publicó el articulo "Mentes, Máquinas y Gödel", en el que formula un controvertido argumento en contra del computabilismo. Lucas sostiene que el primer teorema de incompletitud de Gödel demuestra que la mente humana no es una máquina de Turing. Esto lo expone más o menos de la siguiente manera.

Lucas nos pide considerar una máquina construida para producir teoremas de la aritmética. Al respecto sostiene que lo producido por esta máquina es análogo a lo que puede producir un sistema formal[5]. Dada una MT, podemos dar una representación simbólica de ella, asociar símbolos específicos con estados específicos de la máquina, y asociar también *reglas de inferencia* con las operaciones de la máquina que hacen que vaya de un estado a otro. De esta forma las proposiciones aritméticas que la máquina produciría como salida se corresponderán con los teoremas que se pueden probar en el sistema formal.

Ahora supongamos que construimos el enunciado de Gödel para dicho sistema formal[6]. Este enunciado no se puede probar en el sistema, por lo tanto la máquina será incapaz de producir esta fórmula. Sin embargo, un ser humano puede observar y darse cuenta que el enunciado de Gödel es cierto. En otras palabras, hay al menos una cosa que una mente humana puede hacer que esa máquina no puede. Por lo tanto, una máquina no puede modelar completa y adecuadamente a la mente humana. En pocas palabras, la mente humana no es imitable con una máquina de Turing, mucho menos es una de ellas.

Lucas argumenta lo siguiente: *"No ofrezco una simple prueba contundente de que las mentes son inherentemente mejores que las máquinas, sino un esquema para la construcción de una refutación de cualquier tesis mecanicista*

[5]Como ya lo vimos en el capítulo anterior.

[6]Todo esto se basa en la hipótesis de que tanto el sistema como nosotros mismos somos consistentes, es decir que nuestro razonamiento no genera resultados contradictorios.

plausible que pudiera proponerse. La refutación depende de la tesis mecanicista particular que está siendo mantenida, y no pretende demostrar que la mente es uniformemente mejor que la representación mecanicista supuesta de la misma, sino sólo que se trata de un aspecto mejor y, por tanto, diferente. Eso es suficiente para refutar esa tesis mecanicista particular."

El argumento de Lucas ha sido, y sigue siendo, muy controvertido. Algunas objeciones al mismo están relacionadas con la cuestión de la consistencia; otra es la crítica que hace Benacerraf, quien señala que no es sencillo construir un enunciado de Gödel para un sistema formal dado.

En el caso de la consistencia podemos encontrar dos contra-argumentos hacia Lucas. El primero sostiene que no somos capaces de establecer nuestra propia consistencia, lo cual nos llevaría a afirmar que tampoco seríamos capaces de determinar la verdad que concluye Lucas. En otras palabras se puede evitar el argumento de Lucas simplemente afirmando que no somos más que sistemas formales. El segundo contra-argumento afirma que de hecho los seres humanos somos inconsistentes, lo cual haría que fuéramos comparables con máquinas de Turing inconsistentes[7] .

Lucas sostiene que incluso si no podemos probar la consistencia de un sistema dentro del propio sistema, como el segundo teorema de Gödel demuestra, puede haber otras formas de determinar si un sistema dado es consistente o no. También señala que existen pruebas de consistencia finitistas tanto para el cálculo proposicional y el cálculo de predicados de primer orden, y que también hay una prueba (Gentzen) de la consistencia de la teoría elemental de los números[8] .

Lucas argumenta que mientras el segundo teorema de Gödel demuestra que no podemos probar la consistencia de un sistema dentro del propio sistema, puede ser que podemos demostrar que un sistema es consistente con consideraciones externas a él.

Un problema muy serio con la respuesta de Lucas, como él mismo señala,

---

[7]Hutton, A. (1976) "This Gödel is Killing Me"

[8]Favio Ezequiel Miranda Perea, *"Una prueba de consistencia de la aritmetica de Peano, del tipo de Gentzen", Tesis de Licenciatura, UNAM, 1997.*

es que las consideraciones más amplias que utiliza dicha prueba deben ser también consistentes, y esto puede ser cuestionado. Tal vez podamos *salir* de, por ejemplo, la aritmética de Peano y sostener que la aritmética de Peano es consistente apelando a consideraciones ajenas a ella. Sin embargo, no está claro que podamos *salir* de nosotros mismos para demostrar que somos consistentes, por ello Lucas concluye con una frase kantiana:

"Tal vez debemos asumir nuestra propia consistencia, si el pensamiento ha de ser posible en absoluto. Es tal vez como la uniformidad de la naturaleza, no es algo que se establecerá al final de la cadena de un cuidadoso argumento, sino un supuesto necesario que debemos hacer si hemos de empezar cualquier pensamiento en absoluto".

Lucas también piensa que es poco probable que una máquina inconsistente pueda ser una representación adecuada de la mente. Él otorga que en ocasiones los seres humanos somos inconsistentes, pero afirma que de ello no se sigue que somos equivalentes a los sistemas inconsistentes. Por ejemplo, cuando nos damos cuenta de una inconsistencia podemos reconocer que se trata de un error en lugar de una política establecida. En otras palabras, cuando nos damos cuenta de una inconsistencia, por lo general la evitamos. En cambio si realmente fuéramos máquinas inconsistentes deberíamos ser consecuentes con ello afirmando las dos mitades de una contradicción. En efecto, no somos máquinas inconsistentes a pesar de que solemos serlo de momento, pues somos falibles, pero no sistemáticamente inconsistentes.

Por otra parte, si fuéramos máquinas inconsistentes, tendríamos que avalar potencialmente cualquier proposición. Como se mencionó anteriormente, se puede probar cualquier reclamo de una contradicción, por lo que si somos máquinas de Turing inconsistentes, podríamos potencialmente creer cualquier cosa. Pero nosotros no creemos en general cualquier reclamo, por lo que parece que no somos máquinas de Turing inconsistentes.

La crítica de Benacerraf señala que el sistema formal que pudiera representar a la mente humana podría ser extremadamente complejo, tan complejo que nosotros nunca podríamos obtener el conocimiento necesario de su naturaleza para construir nuestra versión del enunciado de Gödel. En otras palabras, entendemos algunos sistemas formales lo suficiente como para construir y ver que es aplicable el enunciado de Gödel para estos sistemas, pero esto no implica que podamos construir y ver la verdad de nuestro propio enunciado de Gödel. Si no podemos, entonces tal vez no somos diferentes de máquinas. Después de todo, podríamos ser muy complicadas máquinas de Turing, pero máquinas de Turing, sin embargo.

Una respuesta, según Lucas, es que aunque difícil, podríamos, al menos en principio, determinar cuál es la fórmula gödeliana para cualquier sistema dado.

3.3.2. Penrose

Roger Penrose retoma el argumento de Lucas y a partir de la versión del teorema de Gödel presentado por Turing, conocido como "Demostración del insolucionable problema de la *detención*", afirma que ningún computador podrá alcanzar al ser humano en el ámbito del razonamiento matemático, ya que el ser humano posee capacidades intuitivas *no algorítmicas*, además los modelos informáticos no garantizan juicios de verdad.

"De hecho, mi postura es contraria a la inteligencia artificial [...] Creo que la inteligencia no puede simularse mediante procesos algorítmicos, es decir, mediante una computadora, en el sentido que hoy utilizamos el término, claro. Debe haber un ingrediente distinto, no-algorítmico, en la forma de actuar de la conciencia. La inteligencia no es similar a un programa de computadora."[9]

Penrose ha formulado y defendido versiones del argumento gödeliano en dos libros, *La Nueva Mente del Emperador* de 1989 y *Sombras de la mente* de 1994. Este último es, al menos en parte, un intento de mejorar el argumento presentado en el primer libro. El planteamiento de Penrose se compone de dos partes principales: (a) un argumento gödeliano para mostrar que la mente humana es no computable y (b) un intento de inferir una serie de condiciones relacionadas con la conciencia y la física a partir de (a).

Penrose ha defendido diferentes versiones del argumento gödeliano. En *La nueva mente del emperador*, sostiene una versión del argumento que era relativamente similar al presentado anteriormente por Lucas (con algunas diferencias de menor importancia).

[9]Entrevista publicada en febrero de 2000, en el número 225 de MUY Interesante.

En su obra de 1994, sin embargo, Penrose formula una versión del argumento que tiene algunas diferencias significativas respecto a la versión de Lucas. Penrose considera a esta versión como el argumento central en contra de la modelación computacional de la comprensión matemática y señala que algunos analistas parecen haber perdido por completo el argumento.

Haremos aquí un resumen del nuevo argumento que plantea Penrose[10], ya que esta es la formulación más clara y breve de su tesis central:

¿Cuáles son los procedimientos de que disponen los matemáticos para asegurar el carácter interminable de un computo en general? ¿Pueden estos procedimientos codificarse de forma algorítmica?

Penrose nos pide suponer que existe una máquina T que engloba todos los procedimientos de que disponen los matemáticos para demostrar convincentemente que una máquina con indice i y entrada n no se detiene[11] . Al evaluar (i, n) en T está nos indica cuando $C_i(n)$ es un computo que no termina. Es decir, si $T(i, n)$ se detiene entonces el computo $C_i(n)$ no se detiene.

Consideremos el conjunto de problemas en los que i es igual a n, es decir el conjunto de problemas tales que el indice de la máquina es igual al de la entrada. Con esto tendríamos que si $T(n, n)$ se detiene, entonces $C_n(n)$ no lo hace. Notemos que, con esto, el cómputo que realiza T sólo depende de un número. Definamos una máquina T' tal que

$$T'(n) = T(n, n) \Longrightarrow \text{ si } T'(n) \text{ se detiene } \Rightarrow C_n(n) \text{ no lo hace}$$

Dado que T' también es una máquina podemos asignarle un número. Sea k el indice de la máquina T'. ¿Qué ocurre cuando evaluamos $T'(k)$?

$$T'(k) = T(k, k) \Longrightarrow \text{ si } T'(k) \text{ se detiene } \Rightarrow C_k(k) \text{ no lo hace}$$

[10]Penrose, Roger (1994)

[11]Con la expresión $C_i(n)$, Penrose indica un algoritmo o cómputo que forma parte de un listado, no obstante en el estado actual eso equivale a suponer que éste lo realiza una MT

En otras palabras si $T'(k)$ se detiene entonces esto nos indica que $T'(k)$ no se detiene **!**

De lo anterior deducimos que $T(k,k)$ no se detiene, pues de hacerlo llegamos a una contradicción, es decir $T(k,k)$ es incapaz de decirnos que no se detiene.

Por ello, si suponemos que T es válida[12] , es decir, que no nos da efectivamente respuestas erróneas, sabemos que $T(k,k)$ no se detiene. Con ello sabríamos algo que T es incapaz de asegurar. Por lo tanto, T no puede englobar nuestra comprensión.

Con este razonamiento Penrose plantea dos posibles conclusiones. La primera es que no existe una máquina que pueda imitar el razonamiento matemático de los seres humanos por lo que la tesis mecanisista estaría errada; la segunda, que de existir dicha máquina, aún así los seres humanos tendríamos capacidades superiores a ella como vimos en la construcción de su argumento.

Se piensa que la mayor vulnerabilidad de este argumento sigue siendo la tesis de que sabemos que somos correctos, la cual es problemática. Hay quienes rechazan la afirmación de que sabemos que somos correctos, o rechazan la afirmación de que somos correctos. Por ejemplo, McCullough[13] afirma que para que el argumento de Penrose pueda tener éxito, dos afirmaciones deben ser verdaderas.

La primera es que el razonamiento matemático humano es correcto, es decir, que cada afirmación que un matemático humano competente considera una verdad incuestionable es en realidad cierto; la segunda es que el hecho de que el razonamiento matemático humano sea correcto en sí es considerado como una verdad incuestionable.

En resumen, McCullough piensa que por lo menos es posible que los matemáticos no sean correctos. En ese caso no sabríamos definitivamente que las matemáticas son correctas. McDermott[14] también cuestiona este aspecto

[12]Penrose utiliza esta expresión, donde nosotros diríamos *"correcto"*.
[13]McCullough, D. (1996) "Can Humans Escape Gödel?"
[14]Mc Dermott, D.(1996) "Penrose is wrong".

de la argumentación de Penrose.

En cuanto a la forma en que los matemáticos trabajan realmente, McDermott señala un ejemplo: En 1879 Kempe publicó una demostración del teorema de los cuatro colores que no fue refutada sino hasta 1890 por Heawood; es decir, hubo un período de 11 años, donde muchos matemáticos competentes estaban equivocados[15].

Penrose trata de superar estas dificultades mediante la distinción entre errores individuales corregibles que los matemáticos en ocasiones cometan y las cosas que saben que son indudablemente verdaderas.

"Si un robot es [...] como un verdadero matemático, aunque todavía comete errores de vez en cuando, estos errores serán corregibles [...] de acuerdo a sus propios criterios internos de la verdad incuestionable."[16]

En otras palabras, si bien los matemáticos son falibles, todavía son correctos porque sus errores se pueden distinguir de las cosas que saben que son indudablemente verdaderas y también se puede corregir. La idea básica es que los matemáticos pueden cometer errores y seguir siendo correctos, ya que sólo las verdades incuestionables son las que perduran e importan; estas verdades son los resultados de un sistema correcto, y no necesitan preocuparse por el resto de resultados de los matemáticos.

3.3.3. Gödel (Wang)

Una pregunta interesante que aún no hemos abordado es: ¿qué pensaba Gödel sobre lo que sus teoremas de incompletitud implicaban sobre el mecanicismo y la mente en general? Antes de iniciar con el punto de vista de Gödel revisaremos un concepto de suma importancia para entender su postura.

---

[15] El revuelo fue enorme. Al poco tiempo de la publicación de su prueba, Kempe, con el apoyo de Cayley, ingresó a la *Royal Society* en Londres, pero cuando Heawood descubrió el error no ocurrió lo mismo.

[16] Penrose, Roger 1994

La incompletabilidad explicada por Gödel

El fenómeno de la inagotabilidad de la matemática está presente siempre de alguna forma, con independencia del punto de vista que se adopte. De esta forma, podríamos explicarlo respecto a la concepción más simple y natural.

Desde este punto de vista, toda la matemática actual es reducible a la teoría abstracta de conjuntos. Por ejemplo, el enunciado de que los axiomas de la geometría proyectiva implican cierto teorema significa que si un conjunto M de elementos llamados puntos y un conjunto N de subconjuntos de M llamados líneas rectas satisfacen dichos axiomas, entonces el teorema vale para N y M. Así, el problema en cuestión es el de axiomatizar la teoría de conjuntos.

Cuando abordamos ese problema podemos ver que el resultado es por completo distinto del que podría haberse esperado. En lugar de terminar con un número finito de axiomas, como en geometría, nos encontramos con una serie infinita de axiomas que puede ampliarse más y más, sin que se vislumbre final alguno y, aparentemente, sin que exista posibilidad de abarcar todos esos axiomas mediante una regla finita que los genere. Gödel afirma que esto sucede por el hecho de que, si deseamos evitar las paradojas de la teoría de conjuntos sin introducir algo enteramente ajeno a los procedimientos matemáticos reales, entonces el concepto de conjunto debe axiomatizarse por etapas.

Gödel da como ejemplo el siguiente: Comenzamos con los enteros, esto es, con los conjuntos finitos de un tipo especial. Tenemos primero los conjuntos de enteros y los axiomas sobre ellos (axiomas de primer nivel), después los conjuntos de conjuntos de enteros con sus axiomas (axiomas de segundo nivel), después los conjuntos de conjuntos de conjuntos de enteros y así para cualquier iteración finita de la operación *"conjunto de"*, tenemos entonces el conjunto de todos esos conjuntos de orden finito.

Pero podemos en ese momento tratar esta colección exactamente de la misma forma en que antes tratamos el conjunto de enteros, es decir; podemos considerar sus subconjuntos (o sea, los conjuntos de orden ω) y formular axiomas sobre su existencia. Este procedimiento puede evidentemente iterarse más allá de ω, y de hecho hasta cualquier número ordinal transfinito. Así, podría requerirse como siguiente axioma que la iteración sea posible para cualquier ordinal.

Mediante este procedimiento podemos observar que no hemos llegado al final todavía, y que no puede haber ni siquiera un final para este procedimiento de introducir nuevos axiomas, ya que la misma formulación de los axiomas hasta cierto estadio da lugar al siguiente axioma.

En la matemática de nuestros días los niveles más altos de esta jerarquía prácticamente nunca se utilizan; puede decirse con seguridad que el 99.9 por ciento de la matemática actual está contenida en los primeros niveles de tal jerarquía. Asi, la totalidad de la matemática existente puede reducirse a un numero finito de axiomas.

Además, no es del todo improbable que este rasgo de la matemática actual tenga algo que ver con otro de sus rasgos: su incapacidad para probar ciertos teoremas fundamentales, como por ejemplo la hipótesis de Riemann, a pesar de muchos años de esfuerzo. Pues puede mostrarse que la pertinencia de los axiomas para los conjuntos de los niveles más altos no se restringe en modo alguno a esos conjuntos, sino que por el contrario tienen consecuencias incluso para el nivel 0, es decir; la teoría de los enteros.

Para ser más exactos, cada uno de esos axiomas conjuntistas entraña la solución de ciertos problemas diofánticos que han permanecido indecidibles sobre la base de los axiomas precedentes. Los problemas diofánticos en cuestión son del siguiente tipo: sea $P(x_1, \ldots, x_n, y_1, \ldots, y_m)$ un polinomio con coeficientes enteros dados y $n + m$ variables $x_1, \ldots, x_n, y_1, \ldots, y_m$, y considérense las variables x_i como incógnitas y las variables y_i como parámetros; el problemas entonces se plantea de la siguiente forma: ¿Tiene la ecuación $P = 0$ soluciones enteras para cualesquiera valores enteros de los parámetros, o existen valores enteros de los parámetros para los que esta ecuación no tiene soluciones enteras?

A cada uno de los axiomas conjuntistas puede asignársele cierto polinomio P para el que el problema recién formulado se convierte en decidible gracias al axioma. Puede lograrse incluso que el grado de P no sea mayor de 4 (este planteamiento se encuentra más desarrollado en las notas de la famosa "Conferencia de Gibbs")[17].

Con esto Gödel explica lo que él llama incompletabilidad de la matemática con respecto a una aproximación concreta a la fundamentación de la matemática, es decir, la axiomática de la teoría de conjuntos. Sin embargo, el que este hecho sea enteramente independiente de la aproximación o concepción escogidas resulta de los teoremas de Gödel.

El argumento

El segundo teorema de Gödel hace particularmente evidente la incompletabilidad de la matemática. Pues hace imposible que alguien pueda establecer cierto sistema bien definido de axiomas y reglas y, al mismo tiempo, pueda de forma consistente hacer la siguiente afirmación sobre dicho sistema: Percibo que todos estos axiomas y reglas son correctos y además creo que contienen toda la matemática. Si alguien afirma lo anterior estaría contradiciéndose, pues si percibe como correctos los axiomas en consideración, también percibirá que son consistentes, con lo que debe poseer una intuición matemática no derivable de sus axiomas. ¿Significa esto que ningún sistema bien definido de axiomas correctos puede contener toda la matemática propiamente dicha? Gödel contesta a dicha pregunta de la siguiente forma:

"Sí, si por matemática propiamente dicha se entiende el sistema de todas las proposiciones matemáticas verdaderas; pero no, si por ello se entiende el sistema de todas las proposiciones matemáticas demostrables."

Para Gödel puede abordarse este asunto desde dos perspectivas a partir de la distinción de estos dos significados de la matemática, uno como matemática en sentido objetivo y el otro en sentido subjetivo. En el sentido objetivo la teoría trata de representar la realidad donde existe la matemática de forma independiente de tiempo, espacio y conocimiento que se tenga de ella. En sentido subjetivo la teoría es nuestra creación, es decir, nosotros inventamos los postulados. Esta aclaración nos permite afirmar que ningún sistema bien definido de axiomas correctos puede abarcar toda la matemática objetiva, puesto que la proposición que establece la consistencia del sistema es verdadera, pero no demostrable.

[17] La versión original de estas notas ha sido restaurada y reescrita en distintos documen-

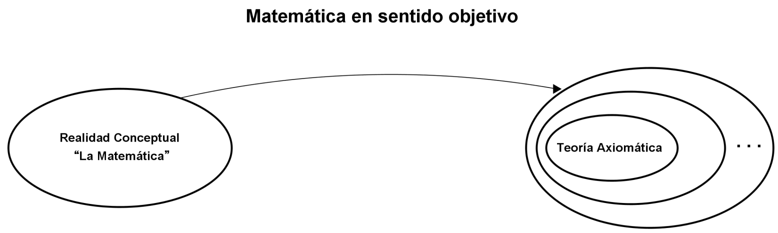


Figura 3.1: La teoría trata de representar la realidad pero requiere cada vez de más axiomas para ello.

Sin embargo, Gödel no excluye la existencia de un algoritmo que genere todos los axiomas evidentes de la matemática subjetiva. No obstante, si tal regla existe, el entendimiento humano ciertamente nunca podría conocerla como tal, es decir, nunca podríamos saber con certeza que todas las proposiciones que genera son correctas, en otras palabras, sólo podríamos percibir como verdaderas una proposición tras otra.

A pesar de esto, la afirmación de que son todas verdaderas podría como mucho conocerse con certeza empírica, sobre la base de un número suficiente de casos particulares o mediante otras inferencias inductivas.

Así, según Gödel, la siguiente conclusión parece inevitable: *"o la matemática es incompletable en el sentido de que una regla finita no puede abarcar nunca sus axiomas evidentes, es decir, que la mente humana (incluso en el reino de la matemática pura) sobrepasa infinitamente la potencia de cualquier máquina finita, o bien existen problemas diofánticos absolutamente irresolubles del tipo especificado"*.

Tenemos entonces que Gödel, a pesar de ser un defensor del realismo conceptual y enemigo de la tesis de Turing, no encuentra un argumento definitivo frente a ello, y con honestidad sólo nos enfrenta a la disyuntiva anterior.

Los teoremas limitativos nos ofrecen una notable herramienta para explorar la cuestión del computabilismo, con la clara posibilidad de resolver un problema filosófico no mediante la discusión verbal, sino haciendo matemáti-

tos, la presentada es una de esas versiones.

cas. En ello, Gödel secundó a Hilbert.

3.4. Conclusiones

Sería absurdo tratar de dar una solución a esta discusión, sobre todo considerando que mi preparación y conocimiento del tema no es el necesario para siquiera intentar tal cosa. No obstante si trataré de dejar en claro mi percepción personal sobre el tema así como mi opinión sobre los principales argumentos presentados en este capítulo.

Es claro que los personajes principales de este trabajo son Kurt Gödel y Alan Turing, y a partir de ellos es que con este trabajo he desarrollado mi propia concepción del tema. Al estudiar con cuidado a Gödel y su idea acerca del computabilismo podemos ver que su intención es favorecer la tesis contraria a éste; no obstante en ningún momento toma una posición definitiva frente a la disyuntiva. Al contrario, él estudia ambas posibilidades y las consecuencias que de ellas se desprenderían.

En este tiempo de trabajo, he podido conocer una parte importante de las formas de ser y de pensar de Gödel, formas que sorpresivamente comparto desde mucho antes de comenzar mi aprendizaje en la facultad. No trato de hacer una comparación entre nosotros, sólo busco explicar un punto. Desde los primeros acercamientos que tuve con las matemáticas hace ya varios años, fue creándose en mi mente la idea de que en ellas podría encontrar todos los secretos que el mundo tiene para el ser humano, que ellas eran la llave maestra para explicar y entender todos los fenómenos que suceden a nuestro alrededor.

Fue en la Facultad cuando mis sospechas no sólo fueron confirmadas, sino que me di cuenta que el universo que conocía apenas comenzaba a revelarse y fue mi formación en lógica lo que me permitió poder ver al mismo tiempo más de cerca y fuera de las teorías. Tengo la firme convicción de que las matemáticas son el único medio que el ser humano tiene para entender su mundo y, de la misma forma, la lógica es la herramienta necesaria para poder entenderlas.

Considero que la capacidad matemática del ser humano esta más allá de cualquier máquina. Gödel plantea que, de suceder lo contrario, habría problemas que no tendrían solución[18], lo cual es a mi criterio imposible. Sin importar el problema que se presente, con el paso del tiempo y el desarrollo de nuevas herramientas, llegará el momento en que encontremos la solución porque de alguna forma Hilbert tuvo razón al decir *"Debemos saber y sabremos"*.

El trabajo de Gödel, lejos de contradecirlo vino a determinar que no hay una forma *sencilla* de realizar dicha tarea, sino que tendremos que seguir trabajando para continuar con nuestro conocimiento del Mundo.

Los teoremas limitativos que Gödel demostró, han sido (y son) usados por varios científicos para formar argumentos sobre el dilema mente-máquina. Tal es la importancia de estos teoremas, que pueden tanto ser los pilares de quienes están en contra del mecanicismo, como también ser el medio por el que los que están a favor argumenten.

La grandeza de los teoremas de incompletud radica en mostrar que las matemáticas tienen una capacidad de desarrollo infinita, esto en vez de desanimarnos debe convertirse en la motivación de cada matemático, pues demuestra el hecho de que sin importar su área de trabajo siempre habrá la posibilidad de crear más.

Quizás las matemáticas sean el reflejo de la evolución de ser humano, ya que han estado con nosotros desde que aparecimos en la faz del planeta y seguramente nos acompañaran hasta el último momento en que estemos aquí. Lo cual vendría a decirnos que al igual que ellas nuestra capacidad evolutiva y de desarrollarnos son inagotables.

Por otro lado el trabajo de Turing es también un gran regalo para el crecimiento de toda la civilización, su genio e inventiva le ha permitido al ser humano tener capacidades que antiguamente sólo podíamos ver en caricaturas y películas de ciencia ficción y que ahora están literalmente al alcance de nuestras manos.

Considero que el test de Turing no es una prueba de que las máquinas serán en un futuro *iguales* a nosotros; más bien, es la llave para crear nuevos

[18]Problemas Diofánticos

caminos de conocimiento y ayudarnos en el proceso de desarrollo de las matemáticas.

A pesar de los argumentos que tanto Robin Gandy como Andrew Hodges tienen sobre la evolución de la inteligencia artificial, es de notar que al final de sus días el mismo Turing considero que la mente esta constituida por dos partes. La primera de ellas sería el pensamiento formal, el cual, sin lugar a dudas puede ser reproducido mediante procedimientos mecánicos; la segunda parte sería la intuición aquella que, sin ninguna prueba o argumento valido, nos permite ver la veracidad de ideas que surgen de nuestro trabajo.

No me cabe la menor duda de que en la historia de las matemáticas han existidos grandes genios que han hecho magnificas aportaciones a las diferentes disciplinas que existen, pero a mi parecer, el hecho de poder vislumbrar más allá de la teoría misma tal como hicieron Gödel y Turing los coloca en un lugar distinto a todos los demás. No es mi intención decir con esto que ellos son mejores que los demás, más bien que son diferentes y si algo nos ha enseñado la historia es que aunque al principio es raro, lo diferente nos agrada.

Apéndice. El test de Turing en la actualidad

El pasado 7 de junio de 2014, durante un test organizado por la *Royal Society* de Londres, una computadora ha logrado superar con éxito el test de Turing. Su nombre es **Eugene Goostman**, y ha logrado convencer a un 33 % del jurado que le examinaba de que es un joven adolescente ucraniano de 13 años de edad. Según los organizadores de la prueba, se trata de la primera "inteligencia artificial" que logra pasar el famoso test de Turing.

Eugene Goostman fue programado en 2001 por un grupo de investigadores rusos y desde entonces han ido perfeccionando sus habilidades hasta conseguir engañar a más del 30 por ciento de los humanos que le interrogaban durante una conversación escrita de cinco minutos de duración. Y ese es, precisamente, el desafío propuesto a mediados del siglo pasado por Alan Turing en su histórico artículo "Computing Machinery and Intelligence". Si una máquina lograba alguna vez superar el reto, significaría que era capaz de pensar.

Por supuesto, el éxito de la prueba no ha tardado en suscitar controversias. Algunos expertos han tachado de exagerado el anuncio de los organizadores del evento, alegando que los programadores utilizaron el sentido del humor y la edad del personaje para que Eugene pudiera confundir a sus examinadores y ocultar sus tendencias no humanas.

Con todo, Eugene Goostman ha logrado convencer a muchos de que nació en Odessa, hace 13 años, de que su padre es ginecólogo y de que tiene a una cobaya por mascota. A lo largo de su existencia, Goostman ha sido sometido al test de Turing en varias ocasiones. Ya estuvo cerca de superarlo en 2001, 2005 y 2008. Y en 2012 logró convencer de su "humanidad" al 29 % de los jueces que le examinaron durante la competición, que ganó contra otros cinco competidores artificiales.

Kevin Warwick, profesor de la Universidad de Reading, señala, *"Algunos dirán que el test ya había sido superado. Las palabras* ***Test de Turing*** *se habían aplicado a competiciones similares en el mundo[...]Pero la prueba del pasado fin de semana ha sido verificada independientemente y, lo que es más importante, las conversaciones no estaban restringidas".*

Warwick recalca que un auténtico Test de Turing no se debe realizar con preguntas o temas de conversación preestablecidas, como en el caso de otras competiciones.

Han aparecido en la comunidad científica varias objeciones sobre el mismo concepto de “inteligencia artificial”, así como las críticas a los criterios utilizados por el test de Turing. Pero ahora se han alzado nuevas voces acerca de otros detalles polémicos en esta historia:

- Un chatbot (como es el caso que nos ocupa) es sólo un programa, no una máquina. Difícilmente podemos hablar de inteligencia artificial en el caso de un mero software ejecutable en cualquier PC, y que no requiere de una supercomputadora.

- Aun si aceptáramos a los chatbots como candidatos al título de inteligencia artificial, cabe recordar el precedente de *Cleverbot*, un software que ya superó en 2011 el Test de Turing (convenciendo en este caso nada menos al 59 % del jurado): Eugene no tendría absolutamente nada de especial, entonces.

- Que Eugene usara en su conversación con el jurado la excusa de ser un niño extranjero de 13 años para encubrir respuestas deficientes o incoherentes es a todas luces una mera treta para trampear las condiciones de la prueba.

- Resulta muy poco científico comunicar a los medios un logro técnico de este nivel tras un único intento exitoso. La Ciencia no funciona así: se deben poner los resultados en conocimiento de otros científicos para que puedan revisar en qué condiciones se ha llevado a cabo la prueba, así como darles la oportunidad de replicar por sí mismos los resultados de la misma.

- Por último, uno de los redactores de BloombergView tuvo la ocasión de charlar directamente con Eugene y dejó claro que no entendía cómo había podido engañar a los miembros del jurado. Esto es un fragmento de la conversación que mantuvieron:

Periodista: ¿Dónde vives?

Eugene: Soy de Ucrania, de la ciudad llamada Odessa. Es posible que hayas oído hablar de ella.

P: He oído hablar, sí. ¿Estabas allí el 2 de mayo?

E: ¿Estaba? Preferiría no hablar de ello, si no te importa. ¿Quieres preguntarme algo más?

P: Yo sí quiero hablar de eso. Sólo me preguntaba si estabas en Odessa el 2 de mayo y viste lo que pasó en el Union Hall[19]

E: Creo que puedes responder tú mismo a esa pregunta porque es retórica. Por cierto, ¿puedes contarme algo acerca de tu profesión?

P: Soy periodista.

E: Periodista no es una mala profesión, especialmente para las personas curiosas.

P: Soy curioso. Así que ¿viste el incendio en Union Hall el 2 de mayo?

E: ¡Qué preguntas más estúpidas haces! Por cierto, ¿me dijiste de dónde eres?

Las objeciones sobre el supuesto éxito sobre el test sostienen un punto que debe ser revisado. Dada la enorme importancia que representaría el superar esta prueba se debe realizar un escrutinio más a fondo de las capacidades de Eugene.

Como se expone en una de las objeciones citadas, hay una línea muy delgada entre programar a la máquina para superar el test al imitar las respuestas de un ser humano y de programarla para eludir las preguntas y confundir a su interrogador.

Sólo el tiempo determinará si este evento será el inicio de una revolución para la inteligencia artificial.

[19]más de cuarenta personas murieron en un incendio en medio de los enfrentamientos entre unionistas y pro-rusos

Bibliografía

[1] Andrew Hodges.
(1992)*Alan Turing: El Enigma* Vintage edition, London, England. First published by Burnett Books Ltd, 1983
(2008) *Alan Turing, Logical and Physical*, Wadham College, University of Oxford, Oxford OX1 3PN, United Kingdom

[2] Bridges, Douglas.
(1994) *Computability: A Mathematical Sketchbook*, Graduate texts in mathematics, Springer-Verlag, New York USA, ISBN: 3-540-94174-6.

[3] Copeland, Jack B.
(2004) *The Essential Turing*, Oxford University Press, USA. ISBN 0-19-825080-0.

[4] Chalmers, David J.
(1995) *Minds, Machines, and Mathematics*, Departamento de Filosofia, UA, Tucson, Arizona. USA.

[5] Feferman, Solomon.
(1996) *Penrose's Gödelian argument*, PSYCHE 2, 21-32, Universidad de Stanford, California USA. (2006) *The nature and significance of Gödel's incompleteness theorems*, Instituto de estudios avanzados. Princeton, USA.

[6] Gödel, Kurt.
(1994) *Algunos teoremas básicos sobre los fundamentos de la matemática y sus implicaciones filosóficas (1951)*, en Kurt Gödel. Ensayos inéditos. Edición a cargo de Francisco Rodríguez Consuegra. Mondadori. Barcelona, España.

[7] Hoffman, Achim.
(2010) *Can Machines Think? An Old Question Reformulated*, Minds and Machines, Springer, Science+Business Media B.V. USA.

[8] Lahoz-Beltrá, Rafael.
(2009) *TURING. Del primer ordenador a la inteligencia artificial*, NIVOLA Libros y Ediciones, S.L., Madrid, España, ISBN: 978-84-92493-51-7.

[9] Lucas, J.R.
(1961) *Minds, Machines and Gödel*, ed. Alan Ross Anderson, Prentice-Hall, 1954, pp.43-59. (1996) *Minds, Machines and Gödel: A Retrospect* en P.J.R.Millican and A.Clark, eds., Machines and Thought: The Legacy of Alan Turing, Oxford, 1996, pp.103-124.

[10] Mirás C., Miguel A.
(2009) *Consideraciones histórico-didácticas relativas a un ensayo de Alan M. Turing sobre el teorema central del límite*, Departamento de Matemáticas. Universidade de Vigo, Espaã.

[11] Pareja H., Diego.
(2013) *¿Qué buscaba Leibniz con su Charasteristica Universalis?*, Universidad de Quindío. Armenia, Colombia.

[12] Penrose, Roger.
(1989)*The Emperor's New Mind: Concerning Computers, Minds, and The Laws of Physics*, Oxford University Press, Inglaterra.
(1994)*Shadows of the Mind: A Search for the Missing Science of Consciousness*, Oxford University Press, Inglaterra.

[13] Preisser R., Ma. de la Asunción
(2003)*Lógica matemática III*, Vínculos Matemáticos. Facultad de Ciencias, UNAM. México.

[14] Sicard, Andrés y Vélez, Mario.
(2001) *Hipercomputación: La próxima generación de la computación teórica*, Revista Universidad EAFIT No. 123, Colombia.

[15] Torres Alcaraz, Carlos.
(2000) *La Lógica Matemática en el siglo XX*, Miselánea Matemática

31, pp.61-115, SMM, México
(2013) *Cien años de Turing* Miselánea Matemática 56, pp.1-25, SMM, México
(– –) *¿Ignoramus et ignorabimus?*, Facultad de Ciencias, UNAM. México.

[16] Turing, Alan M.
(1950) *Computing machinery and intelligence*, Mind, 59, pp.430-460, Manchester, Inglaterra.

[17] Wang, Hao.
(1921) *A Logical Journey: From Gödel to Philosiphy.* The MIT Press. Cambridge, Massachusettes. Londres, Inglaterra.

[18] Zabell, A. L.
(1995) *Alan Turing and the Central Limit Theorem*, The American Mathematical Monthly, Vol. 102, No. 6 (Jun. - Jul., 1995), pp. 483-494. USA.

[19] Zenil Chavéz, Héctor.
(2005) *Encaje de las Redes Neuronales Recurrentes Analógicas en la Jerarquía Aritmética y los lenguajes que reconocen según la complejidad de sus pesos*, Facultad de Ciencias, UNAM. México.