UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE LETRAS Y CIENCIAS HUMANAS

UNIDAD DE POST GRADO

Computabilidad y máquina de Turing

TESIS
para optar el grado académico de Magister en Filosofía con Mención en Epistemología

AUTOR
Miguel Angel Salinas Molina

Lima –Perú
2011

Porque en la mucha sabiduría hay mucha molestia; y quien añade ciencia añade dolor.

Eclesiastés 1:18

Jehová con sabiduría fundó la tierra; afirmó los cielos con inteligencia. Con su ciencia, los abismos fueron divididos y destilan rocío los cielos.

Proverbios 3:19-20

El corazón del inteligente adquiere sabiduría y el oído del sabio busca ciencia.

Proverbios 18:15

A mi madre y

hermanos

ÍNDICE

INTRUDUCCIÓN 6

Capítulos

I. DEFINICIONES PRELIMINARES. 11

1.1. Axiomática 12

1.2. Las inferencias y los silogismos. 21

1.3. La inducción matemática. 34

1.4. Recursividad. 38

1.5. Incompletitud. 50

1.6. Modelo 54

II. COMPUTABILIDAD Y RECURSIVIDAD. 64

2.1. El número. 66

2.2. Los lenguajes formales. 76

2.3. Enumerabilidad. 86

2.4. Diagonalización. 91

2.5. Funciones computables. 94

2.6. Cálculo Lambda. 99

2.7. Tesis de Church. 103

2.8. El Método Logístico y Turing. 111

III.	AUTÓMATAS Y TEORÍA DE TURING.	114
3.1.	La Máquina de Turing.	116
3.2.	Máquina de Turing y el Oráculo	126
3.3.	Máquina de Turing que aprende.	130
3.4.	Teoría de autómatas y lenguajes.	136
3.5.	Tesis de Turing.	146
3.6.	Computabilidad y vida artificial.	153
IV.	COMPUTABILIDAD Y MÁQUINA DE TURING.	163
4.1.	Tesis de Church-Turing.	165
4.2.	Etimología de cálculo y computable.	170
4.3.	Filosofía de la computación.	173
4.4.	Computabilidad.	186
4.5.	Complejidad computacional.	191
4.6.	Paso a Paso.	196
4.7.	Máquina de lápiz y papel.	199
4.8.	Recursividad e inducción matemática.	201
4.9.	Interacción y comunicación.	209
4.10	Modelo y Máquina de Turing.	214
4.11.	La Computadora.	216
Conclusiones		224
Bibliografía.		226

INTRODUCCIÓN

La presente investigación evalúa el concepto de computabilidad en la teoría de Alan Turing, justificándose por la existencia de opiniones divergentes entre diversos académicos, expresadas alrededor del significado de la tesis de Church-Turing, que trata a la función recursiva como equivalente al procedimiento efectivo.

En la actualidad, operamos computadoras conectadas a la Internet, resultando habitual relacionar como computabilidad lo que se puede hacer en una computadora. Muy pocas veces asociamos a la computadora con la ejecución de un cálculo aritmético, tal vez porque disponemos de las máquinas calculadoras. En un sentido amplio, utilizamos la palabra computable como sinónimo de la obtención de un resultado utilizando una computadora.

Cuando mencionamos cálculos, no sólo significa operaciones numéricas, también corresponde a operar símbolos, como ocurre cuando entendemos los elementos que nos rodean, los percibimos como fenómenos, incluso nos percatamos del ánimo de las personas y las interpretamos, aunque no siempre acertemos sobre los estados de ánimo.

El cálculo asociado a la psicología, ayuda en cuanto explica nuestras conductas.

0Según ciertos autores refieren a intenciones como representaciones de la mente; así lo expresa Tim Crade en su libro, de 1995, *The Mechanical Mind, A philosophical Introduction to Minds, Machines and Mental Representatión.* Sin embargo, no tratamos sobre temas de la mente. Elucidamos el concepto de computación en relación con lo que se hace en las computadoras.

La computabilidad tiene un sentido más estricto en los textos utilizados en la ciencia de la computación. Hay dos términos que son significativos, el primero sobre la obtención de un valor mediante una función, y el segundo sobre la ejecución de un procedimiento. Ambos se consideran equivalentes y coextensivos mencionados en la tesis de Church-Turing, enunciada por el matemático Stephen Kleene, en los años 1950.

Consideramos que la tesis de Church-Turing no contiene la teoría de computabilidad que profesó Alan Turing, dado que el ilustre matemático abarcó diversos temas, que por circunstancias del celo en la protección del conocimiento científico por parte de los diversos gobiernos británicos, no permitieron que se divulguen oportunamente. De otro lado, la tesis sostenida por Kleene se contextualiza entre los años cuarenta y cincuenta del siglo pasado, cuando aún no existía la computadora, y que resultaron necesarias en los primeros pasos de la ciencia de la computación. El objetivo era la potencia de cálculo, así nos cuenta William Aspray en su libro, de 1993, *John von Neumann and The origins of modern computing.*

El concepto de computabilidad como procedimiento se encuentra en las matemáticas desde tiempos inmemoriales, por ejemplo, en la determinación del máximo común divisor de dos números enteros, que se resuelve mediante el algoritmo de Euclides.

Por otro lado, utilizamos expresiones formales para representar diversas situaciones como ocurre en las partituras musicales, que data desde el siglo XV, en la que correspondemos sonido, ritmo, armonía, escalas, tiempo y número.

Destacamos el hecho de que la tesis de Church-Turing no ha sido demostrada matemáticamente. Su formulación es una definición intuitiva, dado que contiene el concepto de algoritmo que es de la misma naturaleza. La constatación de la veracidad de la tesis se sustenta en diversas demostraciones que resultan equivalentes. Por esta razón, evaluamos el significado de computabilidad y nos proponemos un esclarecimiento del mismo.

La aplicación de la tesis de Church-Turing y los estudios en el campo de la lingüística contribuyeron en la definición de los lenguajes de programación, basados en los trabajos de Noam Chomsky sobre gramáticas generativas, tratados en sus libros *Syntactic Structures,* en 1957, y *Aspects of the Theory of Syntax,* en 1965. Los lenguajes de programación corroboran en cierto sentido la tesis mencionada, como se muestra en los libros de John Hopcroft y Jeffrey Ullman, *Formal Languages and their relation to Automata,* en 1969, y de Alfred Aho y Jeffrey Ullman en su libro *The theory of Parsing Translation and Compiling,* en 1972.

Alan Turing, en 1936, demuestra mediante una definición abstracta de máquina, que en la aritmética, existen problemas indecidibles. En forma simultánea, Alonzo Church demuestra que existen problemas sin solución, formalizando el cálculo lambda. En los años 1950, Kleene sostiene que la tesis de Church y las máquinas de Turing son equivalentes en relación al concepto de función computable.

Se dice que la tesis de Church-Turing expresa la definición de los algoritmos, pero nos percatamos que no suele ser mencionada en diversas áreas de investigación en la ciencia de la computación, tales como los algoritmos basados en colaboración, como indica C.A.R. Hoare y Jones C.B. en su libro *Essays in Computing Science,* en 1989, o los que se ejecutan en forma paralela como indica Per Brinch Hansen en su libro *The Architecture of Concurrent Programs,* en 1977, o los que se detienen o activan por condición, como indica el libro *Structured Programming,* de 1972, que fuera compilado por Hoare, Dijkstra y Dahl, entre otras como el artículo de Peter Wegner sobre *Introduction to System Programming*, y también el artículo de O.J. Dahl, E.W. Dijkstra y C.A.R. Hoare sobre *Structured Programming.*

El concepto de algoritmo se enriqueció en cuanto a su uso en las estructuras de datos, como indica E.F. Codd en su *libro The Relational Model for Database Management,* en 1990, también en la codificación, optimización y seguridad, como indica Dorothy Elizabeth en su libro *Cryptography and Data Security,* publicado en 1982.

La presente investigación analiza la tesis de Church-Turing, y comparamos los conceptos de función efectivamente computable y el de Turing computable. La primera refiere a una función matemática que obtiene un valor mediante funciones recursivas, y la segunda refiere al concepto de procedimiento en una máquina.

De otro lado, la concepción de la computadora data de tiempos anteriores, pero la fijamos desde la segunda guerra mundial. Así sabemos que los alemanes construyeron la máquina codificadora llamada *Enigma* que se utilizó para la transmisión de mensajes mientras que los ingleses fabricaron un equipo similar llamado *Colossus.* En Alemania, en 1941, se construyó el primer computador llamado *Z3*, su constructor Konrad Zuse. En Estados Unidos, Howard Hathaway Aiken diseñó el computador *Mark I*, operando en Harvard en 1944. Los ingleses diseñan el computador *ACE* del *National Physical Laboratory*, en 1945. Luego la *Electronic Numerical Integrator and Calculator, ENIAC*, de la Moore School of Electrical Engineering de la Universidad de Pennsylvania, siendo sus inventores John Presper Eckert y John William Mauchly, operativa hasta 1955.

Años después las computadoras seguirían siendo mejoradas, adquiriendo mayor capacidad y dispositivos más potentes, se incorporaría la multimedia (sonido, imágenes y lógica), y luego la relevancia de la comunicación expresada en las redes de computadoras.

Para defender la hipótesis de que el concepto de computabilidad es más amplio al de solo recursividad, presentamos cuatro capítulos. En el primero, sobre definiciones necesarias. El segundo capítulo versa sobre la tesis de Church y el significado de lo que se entiende por función efectiva. El tercer capítulo, discurre sobre la teoría de Turing. El cuarto capítulo resalta las diferencias entre lo propuesto por Church y Turing.

Finalmente agradezco a las personas que han ayudado a la presente: Mi madre y a mi hermana; amigos de Maestría: Nery Romero, Leonardo Rubio y Oscar Cuya. A los que contribuyeron en reuniones de discernimiento: David Villena, Ricardo Castillo, Raúl Huaita, Lizardo Luna, Edwin Ortiz, Ana Lepore y Luis Villanueva. Así también a quienes contribuyeron en la revisión del texto y las traducciones: Fabiola Malpartida, Norma Mattos y Tahyri Torres, a mi asesor: Oscar García Zárate, que desde el inicio me animó y aconsejó, también a mis profesores, entre ellos a Magdalena Vexler.

Las matemáticas no pueden apoyarse en la creencia. No importa que tan firme sea esta, sino que están obligados a llevar a cabo una elucidación hasta las últimas consecuencias[1].

David Hilbert

CAPÍTULO I

DEFINICIONES PRELIMINARES.

Iniciamos presentando conceptos necesarios a fin de revisar el significado de computabilidad. Afirmamos que una definición de sólo funciones recursivas en la computabilidad reduce el significado de lo computable. Para sustentar lo mencionado, incorporamos en el marco teórico, conceptos que son expuestos filosóficamente. Este tipo de análisis suele llamarse metamatemáticas, ya que utilizamos conceptos de conceptos matemáticos, seguimos en cierto sentido una orientación historicista, porque nos referimos a los momentos en que fueron formulados.

Revisamos el significado que tiene la denominada tesis de Church-Turing, formulada por Stephen Kleene[2] en los años 50 del siglo pasado. Su investigación es considerada como parte de la filosofía de la computación y de las matemáticas en lo

---

[1] David Hilbert en *La Fundamentación de la Teoría Elemental de Números,* (1931). En *La Antología Fundamento de la Matemática* de Álvarez Carlos, Martínez Rafael, Ramírez Santiago y Torres Carlos.

[2] Cole Kleene, nace el 5 de enero de 1909 en Hartford, Connecticut y muere en Madison Wisconsin el 25 de enero de 1994. Lógico y matemático. Se especializó en la teoría de las funciones recursivas. Recibió su doctorado en matemáticas en Princeton en 1934. Escribió diferentes artículos y libros, destacando *Introducción a la matemática (1952) y lógica matemática (1967).* Fue director de los departamentos de matemáticas y de análisis numérico de la Universidad de Wisconsin. Se especializó en las funciones recursivas y la teoría de los autómatas

particular. La tesis mencionada nos dice: El cálculo efectivo (expresada mediante funciones recursivas) es equivalente a un procedimiento efectivo (instrucciones simples). Ambos conceptos (cálculo efectivo y procedimiento efectivo) son considerados equivalentes en relación a la función computable.

En la presente investigación comparamos los conceptos mencionados y los diferenciamos, siguiendo sus sentidos concomitantes, resultando necesario incluir en el marco teórico temas sobre: Axiomática, inferencia y silogismo, inducción matemática, recursividad, incompletitud y modelo. Estos conceptos se encuentran relacionados al de algoritmo, que fuera mencionado en el planteamiento de Church[3] y expresado por Kleene en la Tesis de Church-Turing, en cuanto a las funciones recursivas, como poderosa formalización que contiene un esquema matemático muy rico e interesante.

1.1. Axiomática.

Cuando hablamos de axiomática nos referimos a una forma de definir una teoría conformada por un grupo de proposiciones[4] que se consideran verdaderas, a las que llamamos axiomas[5] y desde éstas derivamos todas las otras proposiciones que son parte de la teoría que se conocen como teoremas.

---

[3] Alonzo Church, nace el 14 de junio de 1903 en Washington y muere el 11 de agosto de 1995 en Hudson, Ohio. Matemático y lógico norteamericano, contribuyó en crear la base de la computación teórica. Se diplomó en la Universidad de Princeton en 1924 y obtuvo su doctorado en 1927, donde ejerció como profesor entre 1929 y 1967. Su obra más conocida es el cálculo lambda y su trabajo de 1936 que muestra la existencia de problemas indecidibles. Este trabajo precedió el famoso trabajo de su alumno Alan Turing sobre el problema de parada que también demostró la existencia de problemas irresolubles por dispositivos mecánicos

[4] La proposición se define, siguiendo a Aristóteles, como un discurso enunciativo perfecto que expresa un juicio y significa lo verdadero o lo falso. La proposición es enunciativa. (Ferrater Mora, tomo III, 2004:2930).

[5] Axioma. El terminó axioma, en sus orígenes, alude a una verdad que es evidente por sí misma. Este es el significado que Aristóteles le adjudica en los Analíticos posteriores. (Tomado del glosario del libro de García Zárate, 2007).

Los axiomas cumplen reglas[6] que permiten definir en forma deductiva, a esto lo llamamos sistema axiomático, y su principal característica es que basado en la verdad de los axiomas y del mecanismo deductivo nos garantiza la obtención de los teoremas. La Real Academia Española define axiomática como: "Conjunto de definiciones, axiomas y postulados en que se basa una teoría científica".

Suele decirse que el sistema axiomático es un conjunto de verdades de una determinada "realidad" definida en los axiomas, que son proposiciones que no pueden ser derivadas de ningún otro. Frege[7] sostuvo en contradicción con Hilbert[8], que los axiomas refieren a verdades intuitivas y tienen correspondencia a una "realidad" que las hace evidentes. Hilbert, por su parte sostuvo que los axiomas son definiciones más abstractas, que necesariamente no corresponden a una "realidad", estas diferencias de interpretación, específicamente con respecto a la axiomatización de la geometría, es mencionada por Mosterín[9] en el capítulo 7 de su libro *Conceptos y Teorías en la Ciencia* (Mosterín, 2003).

[6] En un sentido muy general se ha usado reglas para referirse a los preceptos de que se compone un método. Las reglas incluyen a menudo instrucciones. (Ferrater Mora, tomo II, 2004: 3038).

[7] Friedrich Ludwig Gottlob Frege, Nace en Wismar (actual Alemania) el 8 de noviembre de 1848 y muere en Bad Kleinen el 26 de julio de 1925. Matemático, lógico y filósofo alemán fundador de la moderna lógica matemática y la filosofía analítica. Comenzó sus estudios en la Universidad de Jena en 1869 trasladándose a Gotinga para completar sus estudios de física, química, filosofía y matemáticas licenciándose en esta última 1873. Al regresar a Jena se dedicó a la docencia de matemáticas.

[8] David Hilbert, nace en Königsberg, Prusia Oriental el 23 de enero de 1862, muere en Göttingen, Alemania el 14 de febrero de 1943, matemático alemán. Desarrollo una amplia actividad científica, como la teoría de invariantes, la axiomatización de la geometría y la noción de espacio de Hilbert, uno de los fundamentos del análisis funcional. Hilbert y sus estudiantes proporcionaron partes significativas de la infraestructura matemática necesaria para la mecánica cuántica y la relatividad general. Fue uno de los fundadores de la teoría de la demostración, la lógica matemática y la distinción entre matemática y metamatemática. Defendió vivamente la teoría de conjuntos y los números transfinitos de Cantor. Un ejemplo famoso de su liderazgo mundial en la matemática es su presentación en 1900 de un conjunto de problemas que establecieron el curso de gran parte de la investigación matemática del siglo XX.

[9] Jesús Mosterín nació en Bilbao en 1941. Filósofo español, estudió en España, Alemania y Estados Unidos. Obtuvo la cátedra de Lógica y Filosofía de la Ciencia en la Universidad de Barcelona. Desde 1996 es Profesor del Instituto de Filosofía del CSIC. Sus reflexiones se sitúan en la frontera entre ciencia y filosofía en la que la racionalidad es su hilo conductor. Es miembro del *Center for Philosophy of Science* de Pittsburgh, miembro de la Academia Europea de Londres, del *Institut International de Philosophie* de París y de la *International Academy of Philosophy of Science*. Contribuyó en la difusión de la filosofía analítica, de la lógica matemática, de la filosofía de la ciencia en España y América Latina. Ha desempeñado funciones editoriales en varios países. Se ha involucrado en la protección de la naturaleza y la defensa de los animales.

La axiomatización como método para definir una teoría, data de la época de Euclides[10] que formuló las bases para la axiomatización de la geometría a partir de cinco axiomas. Tomamos en cuenta esta experiencia, porque nos ilustra dos temas interesantes. La primera se refiere a la integración de los diversos conceptos en un cuerpo teórico. La segunda, y tal vez la más importante, en cuanto y cuando permite la creación y formulación de nuevas teorías, dado que desde un sistema axiomático es posible construir nuevos sistemas. Este último criterio se aparta del concepto del axioma como verdad intuitiva o verdad material. Puede darse un sistema axiomático y al mismo tiempo otro con la negación (definición contraria) de uno o más axiomas del primero.

La axiomática agrupa diversos conceptos en una unidad teórica, destacamos que en la axiomatización de una teoría, no sucede la sincronía con el tiempo, en el sentido que no ocurre primero la formulación de los axiomas para luego deducir los teoremas. Se requiere conocer todo el cuerpo de la teoría para definir los axiomas como verdades no demostrables, se suele formular los teoremas antes de que se definan los axiomas. Éste fue el caso en la geometría de Euclides, sabemos que el teorema de Thales (639 al 547 a.C.) describe la proporción aritmética entre las longitudes de segmentos de recta que se forman de dos líneas diagonales que cruzan a tres rectas paralelas, este enunciado fue formulado antes de la axiomatización de la geometría euclidiana, dado que Thales vivió antes de Euclides. También con el teorema de Pitágoras (582 al 507 a.C.) que describe la igualdad de la suma de los cuadrados de la longitud de los catetos con el cuadrado de la longitud de la hipotenusa de un triángulo rectángulo. La axiomatización de la geometría euclidiana

[10] Euclides de Alejandría, 365 antes de J.C., no se conoce su lugar de nacimiento, se supone que estudió en Atenas con los platónicos y fundó la escuela de matemáticas en Alejandría, sistematizó en quince libros las matemáticas de su época.

ocurrió posteriormente a la existencia de los teoremas, de forma tal que se agrupó en un cuerpo teórico los diversos conceptos. Euclides concluye con un orden y un esquema deductivo a la geometría.

Es posible formular otros sistemas axiomáticos desde uno establecido, así siguiendo con nuestro ejemplo de la geometría: Ocurrió que durante cientos de años algunos matemáticos dudaron de la independencia del quinto axioma de Euclides, se afirmaba que no era en realidad un axioma sino más bien un teorema, este dice: *"por un punto exterior a una línea recta sólo puede trazarse una línea paralela"*. No fue considerado evidente y durante ese tiempo se trató de demostrar que no era un axioma. Fue recién en el siglo XIX cuando se encontró una solución, reemplazar el axioma por una proposición contraria, ésta es una estrategia en matemáticas para hacer una demostración, consiste en negar lo que se quiere demostrar, y a partir de ésta, si se concluye una contradicción, entonces la negación es un error, por lo tanto se concluye lo que se quería demostrar[11].

El reemplazo del quinto axioma de Euclides por las proposiciones con sentido contrario, tales como: "no se puede trazar ninguna línea paralela" y "se puede trazar más de una línea paralela", que son formas distintas de negar el quinto axioma. Permitió en cada caso, la formulación de un nuevo sistema axiomático. La solución a la cuestión planteada obligó admitir que si no es adecuado el axioma de las paralelas, entonces su negación podía justificar nuevos sistemas axiomáticos y de esa forma se definieron dos nuevas geometrías.

---

[11] La demostración por el método de reducción al absurdo consiste en negar la conclusión de una proposición de la forma $p \rightarrow q$. Negando q hay que demostrar la negación de p. Ésta se basa en la tautología de la lógica clásica llamada contraposición en la que $p \rightarrow q$ es equivalente a: $\neg q \rightarrow \neg p$ (no q entonces no p).

Con la creación de las otras geometrías (llamadas no euclidianas) se cuestiona la auto evidencia de los axiomas, en el sentido que las teorías matemáticas no son verdaderos en correspondencia a una realidad sensible. Por lo tanto, en la negación de ciertos axiomas o en el cambio, se presenta la definición de nuevos sistemas axiomáticos, tal como en la geometría de Euclides, que permitieron la definición de las geometrías hiperbólica enunciada por Lobachevskiy[12] y la elíptica enunciada por Reimann[13].

La geometría de Euclides definida en cinco postulados, en la que el quinto es mencionado de forma diferente en los textos, así tenemos:

i. Es posible trazar una línea recta desde cualquier punto a cualquier otro.
ii. Es posible prolongar continuamente en línea recta una recta dada.
iii. Es posible trazar un círculo con cualquier centro y distancia radio.
iv. Todos los ángulos rectos son iguales entre sí.
v. Si una recta incide sobre otras dos formando del mismo lado ángulos internos menores que dos rectos, al prolongarlas indefinidamente se encontraran por el lado en que los ángulos sean menores que dos rectos. (Guillermo Martínez[14] y Gustavo Piñeiro[15], 2009: 237).

El quinto postulado, en su expresión original, presenta como condición necesaria la existencia de dos rectas, que si no son paralelas estas deben cruzarse en algún punto,

[12] Nikolai Ivanovich Lobachevski, nace en Nizhni Novgorod, Rusia, 1792 y muere en Kazán, 1856. Entró en la Universidad de Kazán a los 14 años. En 1820 fue nombrado decano de la facultad de Física y Matemáticas; en 1827, rector. En 1829 publica su trabajo que sería denominada geometría hiperbólica. La obra fue poco apreciada en su tiempo y tuvo que esperar a los trabajos de B. Riemann y F. Klein.

[13] Georg Friedrich Bernhard Riemann (1826-1866), nace en Breselenz, Alemania y muere en Selasca, Italia. Se doctoró en Gotinga, hizo diversos aportes a la matemática: la teoría de las variables complejas (superficies de Riemann), especialmente en la geometría no euclidiana conocida con su nombre.

[14] Guillermo Martínez, nace en Bahía Blanca Argentina, 1962. Doctor en matemáticas y escritor, entre sus obras figura *Crímenes imperceptibles*, obra que fue llevada al cine con el nombre *Crímenes de Oxford*.

[15] Gustavo Piñeiro, nace en Buenos Aires, 1966. Licenciado en matemáticas, profesor y divulgador de las matemáticas, coautor con Guillermo Martínez del libro *Gödel para Todos*.

razón que se atribuye de que requiere demostración, así nos dice Brunschvicg[16]: "el postulado de las paralelas de Euclides es un teorema que indica las condiciones en las cuales existe un punto de intersección entre dos rectas" (Brunschvicg, 1945: 117-118)

Los sistemas axiomáticos se comportan como sistemas lógicos, dado que siguen el concepto de verdad de los axiomas en correspondencia a lo que en lógica se denomina tautología, de forma que un sistema axiomático es un método eficiente que ordena la teoría en proposiciones básicas y establece criterios de la derivación. Obteniéndose una teoría como resultado de un proceso de abstracción y ordenamiento de los temas, resultando un instrumento teórico, tal como lo enuncia Blanche[17]: "Si se reflexiona sobre ellas, las ventajas del método axiomático resultan manifiestas. En primer lugar constituyen un instrumento precioso de abstracción y análisis" (Blanche, 1965: 76).

La axiomática refiere a teorías y sus axiomas son proposiciones que corresponden a leyes, Tarski[18] las denomina aserciones: "Toda teoría científica es un sistema de proposiciones que se aceptan como verdaderas y que pueden llamarse leyes o aserciones establecidas o, para abreviar, simplemente aserciones." (Tarski, 1951:21)

---

[16] Léon Brunschvicg, nace en París, 1869 y fallece Aix-les-Bains, 1944. Filósofo francés. Autor de numerosas obras de historia del pensamiento científico y filosófico, *El idealismo contemporáneo*, 1905, *Les étapes de la philosophie mathématique*, 1912; *La experiencia humana y la causalidad física*, 1922). Se doctoro en filosofía en 1897. A partir de 1909 enseñó en la Sorbona.

[17] Robert Blanche (1898-1975) Filósofo francés, seguidor de Descartes. Profesor asociado de filosofía de la universidad Toulouse. Escribió varios libros de temas matemáticos. Sus aportes en el campo de la lógica son diversos, crítica a la escolástica en el silogismo, proponiendo el hexágono con seis posiciones de la lógica.

[18] Alfred Tarski. Nació el 14 de enero de 1902 en la ciudad de Varsovia, Polonia, y murió el 26 de octubre de 1983 en Berkeley, California, Estados Unidos. De origen judío, adoptó su apellido al convertirse en 1923 al catolicismo. Formó parte de la importante escuela polaca de lógica y filosofía hasta 1939, en que se estableció en Estados Unidos; la emigración le salvó de la suerte de la mayor parte de su familia, que pereció bajo la ocupación nazi de Polonia. Desde Estados Unidos, enseñaría la lógica. Sus aportes en teoría de conjuntos, lógica polivalente, niveles de lenguaje y metalenguaje y conceptos semánticos. Fue el autor de *Introducción a la lógica y a la metodología de las ciencias deductivas* en el año 1941 y *La concepción semántica de la verdad y los fundamentos de la semántica* en 1944.

La axiomática es una forma de definición teórica que libera en cierto sentido las restricciones de la habitual interpretación de las expresiones. Requiere de un proceso de abstracción, que exige la coherencia de los axiomas, que no se puedan deducir teoremas contradictorios. Se tiene en la lógica una herramienta que proporciona un mecanismo para las deducciones, pero sucede que muchos matemáticos razonan teorías coherentes sin conocer los principios de la lógica, también ocurre que se puede conocer todos los principios de la lógica y no obtener adecuadamente una teoría, al respecto Tarski dice:

> La Lógica se considera con motivo como la base de todas las otras ciencias, por el hecho de que en todo razonamiento hay que enfrentarse con conceptos del dominio de la Lógica, y porque toda conclusión correcta coincide con las leyes de esta disciplina. Pero de esto no se infiere que el conocimiento exacto de la Lógica sea una condición necesaria para el pensar correcto; incluso los matemáticos profesionales que por lo general no cometen errores al razonar, no conocen de ordinario la Lógica hasta el grado de saber todas las leyes de ésta, en las que se apoyan inconscientemente (Tarski, 1951:121).

Para la definición de un sistema axiomático se requiere de un vocabulario y reglas que combinan los términos del vocabulario. Estas reglas precisan de proposiciones, de las cuales se derivan otras. Es una de las características de la axiomática: Ser un sistema deductivo que contiene proposiciones, las cuales son derivados desde axiomas o proposiciones ya deducidos, así tenemos:

> Si queremos acometer la edificación de una determinada disciplina, caracterizaremos ante todo un pequeño grupo de expresiones de ella que nos parezcan comprensibles de por sí; llamaremos a las expresiones de este grupo conceptos fundamentales o conceptos no definidos y las aplicaremos sin aclarar su significación. Al mismo tiempo, aceptaremos como principio el no utilizar ninguna de las restantes expresiones de la disciplina considerada, las llamadas conceptos deducidos (o expresiones deducidas), en tanto no haya sido determinada su significación con la ayuda de los conceptos fundamentales y de aquellos conceptos deducidos cuya significación ya este aclarada previamente (Tarski, 1951:130).

Como fue mencionado, la axiomatización de una teoría requiere, en primer lugar: La definición de los axiomas, que en el caso, son verdades indemostrables; éstas contienen la teoría y permiten deducir todas las demás proposiciones. Para definir un sistema axiomático se debe tener el universo de las proposiciones, para establecer los teoremas y sus relaciones en cuanto a las demostraciones, evitando la circularidad[19] y garantizando que el sistema sea completo, que todos los teoremas sean deducidos en la teoría.

Para definir un sistema axiomático se requiere contestar la pregunta: ¿Cuáles son las proposiciones que deberían ser axiomas? Para esto, hay que considerar que es un proceso de ordenamiento de una teoría, de manera que una vez que se tienen las proposiciones candidatas a ser axiomas, éstas deben ser analizadas y simplificadas o ampliadas, de manera que garanticen que ningún axioma pueda ser deducido de otra, en palabras de Tarski tendríamos:

> Elegiremos algunos de éstos, los que nos parezcan más evidentes, como proposiciones fundamentales o axiomas y los reputaremos ciertos sin fundamentarlos de ningún modo. En cambio, nos obligaremos a fundamentar todas las demás, llamadas proposiciones deducidas o teoremas (Tarski, 1951:130).

Al proponerse una teoría se suele presentar las ideas base como postulados[20], a éstas se le atribuía la propiedad de ser evidentes y/o verdaderas. La axiomática en el campo de la lógica, refiere a la estructura lógica como un mecanismo de transmisión de la

[19] Circulo vicioso, situación que se da cuando tratamos de definir, explicar o probar una primera cosa en función de otra segunda cosa que a su vez presupone que ya está definida, explicada o probada la primera, por lo que volvemos al punto de partida (Mosterín y Torretti, 2002: 90).
[20] Palabra con la que Euclides designa ciertos enunciados. Axioma y postulado son sinónimos con que se designa indistintamente a las premisas no demostradas de una teoría axiomática (Mosterín y Torretti, 2002: 451).

verdad[21]. Lo expresado se encuentra en el sentido, de que en las matemáticas no corresponde la evidencia material de un axioma, es una conceptualización.

En el transcurso del tiempo ha variado el concepto de axioma, así para Aristóteles los axiomas contienen la verdad que es captada de manera intuitiva, tal como se aprendía la geometría en su época; para Kant[22] la verdad de los axiomas en la geometría corresponden al concepto a priori del espacio, por lo tanto la intuición era un componente para la evidencia de la geometría, luego Hilbert[23], en 1899, en su obra *Fundamentos de la Geometría*, presenta una axiomatización sin recurrir a los conceptos como verdad intuitiva, creando una nueva forma de axiomática, como construcciones de teóricas abstractas.

El concepto de axioma es sinónimo al de postulado, en las matemáticas se elimina la interpretación de verdad en cuanto a que es "real", y establece su importancia en la construcción formal. Al respecto sobre la verdad en los axiomas, consideramos conveniente presentar dos expresiones mostradas por Robert Blanche, las que elucidan la relación entre las matemáticas y la "realidad", la primera de Bertrand Russell[24] en la que dice: "Las matemáticas son una ciencia en la que no se sabe nunca de que se habla, ni es

[21] La lógica no tiene que ver con la realidad al modo como una cosa se relaciona con otra, pues en tal caso habría que adherirse a una determinada teoría metafísica que explicara las supuestas coincidencias. ... Por eso la lógica no describe la textura inteligible de lo ontológico, en el Tomo III (Ferrater Mora, 2004: 2194).

[22] Inmanuel Kant (1724-1804) nace en Königsberg Prusia, considerado uno de los pensadores más influyentes, conocido por sus obras entre muchas: Crítica de la razón pura, Critica de la razón práctica.

[23] David Hilbert matemático alemán, propone en 1900 en el congreso internacional de matemáticos en París y en 1928 en el congreso internacional de Bolonia, un procedimiento algorítmico general para resolver cuestiones matemáticas.

[24] Bertrand Russell (Trelleck, 1872 - Plas Penrhyn, 1970) Filósofo y matemático británico. Estudio en el Trinity College de Cambridge, en calidad de "fellow", en la primera Guerra Mundial tuvo una actitud pacifista que le valió cuatro meses de cárcel, durante los cuales redactó su *Introducción a la filosofía matemática* (*Introduction to Mathematical Philosophy*, 1919). Anteriormente, en 1900, había publicado su libro acerca de Leibniz, y en 1910 *Principia Mathematica* (en colaboración con el filósofo A. N. Whitehead), texto que proponía una interpretación "logística" de las matemáticas. Dicha tesis de la reducción absoluta de tal ciencia a la lógica que había sido también sostenida en *Principles of Mathematics*, en 1903. La "teoría de los tipos", la de los números como "clases de clases" y la "paradoja de Russell" fueron los resultados más significativos de esta amplia labor de investigación.

verdadero lo que se dice", y la segunda de Henri Poincaré[25], que se dice: "Las matemáticas son el arte de dar el mismo nombre a cosas distintas" (Blanche, 1965:41).

La axiomática es una herramienta usada para la formalización de las teorías. Hilbert después de axiomatizar la geometría propone a inicios del siglo XX la axiomatización de las matemáticas. Este planteamiento luego fue conocido como problema de decisión – en alemán *Entscheidungsproblem* – que resulta equivalente a encontrar un procedimiento que permitía deducir cualquier teorema desde los axiomas.

Cabe mencionar que el resolver el problema planteado como el *Entscheidungsproblem*, consiste el encontrar un algoritmo[26] que permita decidir si una fórmula de las matemáticas es un teorema. Según lo expresado, la Axiomática permitiría deducir lo planteado.

1.2. Las inferencias y los silogismos.

Se define que "una inferencia es una operación lógica que consiste en derivar a partir de la verdad de ciertas proposiciones como premisas" (García Zárate, 2007:60).

En la demostración de los teoremas matemáticos se utiliza las inferencias[27] como

---

[25] Jules Henri Poincaré, nace en Nancy, Francia, 29 de abril de 1854 y fallece en París, 17 de julio de 1912, matemático, científico y filósofo de la ciencia. último «*universalista*» capaz de entender y contribuir en todos los ámbitos de la matemática. En el campo de la mecánica elaboró diversos trabajos sobre las teorías de la luz y las ondas electromagnéticas, y desarrolló junto a Albert Einstein y H. Lorentz, la Teoría de la Relatividad Restringida conocida como Relatividad especial. La conjetura de Poincaré es uno de los problemas recientemente resueltos de la topología algebraica, y fue el primero en considerar la posibilidad de caos en un sistema determinista, en su trabajo sobre órbitas planetarias.

[26] Algoritmo es un conjunto finito de reglas cuya aplicación mecánica conduce de modo necesario a la obtención de un resultado (Tomado del glosario del libro de García Zárate, 2007).

[27] En la lógica actual el problema de la inferencia es a menudo un problema metalógico; se trata de sentar ciertas reglas (las llamadas reglas de inferencia) que permiten derivar una conclusión de unas premisas, en el Tomo II (Ferrater Mora, 2004: 1822).

una forma rigurosa y expresiva de dar un paso hacia otro. Cuando se elabora una demostración, ésta se explica en sí misma, y los resultados son consecuencia de lo ejecutado. Las demostraciones fueron definidas utilizando el silogismo, enunciado por Aristóteles[28] en su libro *Analíticos*, ésta es una forma de razonamiento que consta de tres componentes; dos proposiciones que son denominadas premisas y una tercera proposición que viene a ser la conclusión obtenida como inferencia de las premisas.

Suele formularse de manera muy general dos métodos para obtener conclusiones: la deducción y la inducción; algunos sostienen que la deducción es "el paso de lo general a lo particular" y la inducción es "el paso de lo particular a lo general". Siguiendo con lo expresado, se suele sostener, que el campo de aplicación de la deducción es en las ciencias formales, mientras que el de la inducción es en las ciencias empíricas. Otros como Moulines[29] y Diez[30], sostienen que la deducción es una inferencia de proposiciones de naturaleza segura, mientras que la inducción es una inferencia de proposiciones que son probables (Diez y Moulines, 1997). Consideramos oportuno enunciar lo expresado por Charles Peirce,[31] sobre la existencia de tres formas diferentes de inferir, según los tipos de proposiciones, la primera es la idea general a la que denomina Regla; la siguiente es la idea

---

[28] Aristóteles, nace en Estagira, Macedonia, 384 a. C. y fallece en Calcis Eubea, Grecia, 322 a. C. fue uno de los filósofos más influyentes de la antigüedad, autor enciclopédico más portentoso en la historia de la humanidad. Formaliza la lógica, astronomía, precursor de la anatomía, la biología y un creador de la taxonomía. Está considerado (junto a Platón) como el determinante de gran parte del corpus de creencias del Pensamiento Occidental del hombre corriente (aquello que hoy denominamos "sentido común" del hombre occidental). Es reconocido por desarrollar la primera formalización lógica; la formulación del principio de no contradicción.

[29] C. Ulises Moulines, nace en Caracas, Venezuela en 1946. Estudió física y filosofía en la universidad de Barcelona. Se doctoró en la universidad de Múnich en 1975. Ha escrito diversos libros y artículos en filosofía de la ciencia.

[30] José A. Diez, filósofo español. En 1961, estudió filosofía obteniendo su doctorado en la universidad de Barcelona en 1992. Profesor de la universidad Revira i Virgili.

[31] Charles Sanders Peirce (1839-1914) nace en Cambridge, filósofo norteamericano de la corriente del pragmatismo, con trabajos en lógica, su producción es básicamente de documentos publicados en revistas. Actualmente existen diferentes centros de investigación que están redescubriendo sus escritos, reordenando sus documentos, ya no solo por temas, también cronológicamente, encontrándose una filosofía muy rica que abarca diversos aspectos, desde lógica y matemáticas hasta lingüística.

particular a la que denomina Caso; y la conclusión es la idea que relaciona la Regla con el Caso denominándola Resultado.

Peirce relaciona las definiciones de Regla, Caso y Resultado en los silogismos, estableciendo tres tipos de inferencia, según la forma en que participan las premisas y la conclusión, así manifiesta sobre la deducción, la inducción y la hipótesis (a la que nombraría también abducción).

La deducción es el inferir el Resultado (como premisas la Regla y el Caso), la inducción es el inferir la Regla (como premisas el Caso y el Resultado), y la hipótesis es el inferir el Caso (como premisas la Regla y el Resultado).

El concepto de deducción, inducción e hipótesis son ilustrados mediante un ejemplo que trata sobre unos frijoles (judías), en la que construye tres formas de ordenar las proposiciones, así tenemos:

Deducción.	Regla.- Todas las judías de esta bolsa son blancas.
	Caso.- Estas judías son de esta bolsa.
	Resultado.- Estas judías son blancas.
Inducción	Caso.- Estas judías son de esta bolsa.
	Resultado.- Estas judías son blancas.
	Regla.- Todas las judías de esta bolsa son blancas.
Hipótesis	Regla.- Todas las judías de esta bolsa son blancas.
	Resultado.- Estas judías son blancas.
	Caso.- Estas judías son de esta bolsa. (Peirce, 1878:69)

En la reflexión de Peirce no existe el dualismo: Inducción y Deducción, incluye a la Hipótesis, afirmando que en un proceso de inferencia se combinan las tres formas, primero sucede la hipótesis (abducción), luego la inducción (como prueba de algunos casos), para finalmente formular la teoría (deducción), sosteniendo que estas se mezclan de formas tales que determinan diversos tipos de construcciones de inferencias.

Como método, la deducción permite inferir proposiciones en la que sus significados están contenidos en las premisas. Este método fue utilizado por Aristóteles en donde a partir de ciertas premisas (que incluye la regla) se derivan las conclusiones, por ejemplo: Todos los hombres son mortales (la Regla), Sócrates es hombre (el Caso), luego entonces, Sócrates es mortal (el Resultado).

La deducción es la inferencia en la que estamos seguros del resultado, no se obtiene mayor conocimiento dado que está contenido en las premisas, el Caso aparece como condición necesaria en relación a la Regla, si expresamos en notación lógica, es de la forma de la inferencia Modus Pones[32]:

Si, $p \rightarrow q$ (Regla) y p (Caso)

Entonces q (Resultado)

El método deductivo está presente en las teorías que han sido expresadas como sistemas axiomáticos, en donde los teoremas son deducidos desde los axiomas. La deducción consiste en encontrar principios "desconocidos" desde los conocidos. Esto ocurre en la medida que no resulta evidente una determinada proposición, pero está contenida en los axiomas y/o proposiciones anteriores que ya fueron demostradas.

La deducción en la argumentación es una idea (como expresión) seguida de otra,

[32] Regla del Modus Ponens (MP): A partir de una formula condicional y de la afirmación de su antecedente, se obtiene el consecuente (Tomado del libro de García Zarate, 2007:108).

es una secuencia argumentativa, en la que una proposición lleva a otra según corresponde al uso de las reglas de inferencia[33].

Para realizar una demostración matemática, mediante la deducción, muchas veces hay que utilizar la intuición, como un mecanismo que ayuda a fijar una estrategia demostrativa. Una vez que la demostración queda evidente, los pasos o secuencias utilizadas muestran la demostración, eliminando los criterios intuitivos. Las demostraciones basadas en axiomas exigen un rigor que enfatiza en el uso de las reglas sin considerar la intuición, así es como se entiende en el cálculo para la deducción natural introducido por Gentzen[34], en 1934.

La inducción considera la existencia del principio de la regularidad de la naturaleza, como interconexión de los fenómenos en la naturaleza, que permiten pasar de la descripción, referida por los hechos, a la explicación (reconocimiento de causalidad) y predicción a través de leyes y teorías (que podemos sintetizar como lo observado). Se dice que las inferencias obtenidas a través de la inducción contienen la naturaleza de probable en el sentido que está presente la incertidumbre y ésta disminuye a medida que se incrementa el número de casos que se examinan.

La inducción es un método que obtiene reglas generales a partir de la observación

[33] Una regla de inferencia es un permiso convencional para pasar de una fórmula de un cierto tipo (la premisa) a otra nueva (la conclusión), es decir, para escribir esta última, si ya disponemos de la primera. De hecho, no es necesario que la regla de inferencia parta de una premisa; también puede partir de dos o de tres o incluso ninguna (Mosterín y Torretti, 2002:494).

[34] Gerhard Gentzen (1909-1945), nace en Greifswald, Alemania. Es conocido por su reformulación de la lógica de predicados conocidas como reglas de Gentzen. “El método en cuestión consiste en una serie de reglas de inferencia, llamadas reglas de Gentzen, para ejecutar inferencias tanto en la lógica sentencial como en la lógica cuantificacional” (Ferrater Mora, 2004: Tomo I 792).

de casos (según Peirce son el Caso y el Resultado) y que está implicada la probabilidad[35]; en la medida que aumente los casos de observación entonces mejorará la probabilidad (disminuye el error) de la conclusión.

Una representación lógica de la inducción, contendría como premisas los casos particulares, de forma tal que deducimos la Regla, ésta podría ser:

Si p (Caso) y q (Resultado)

Entonces $p \rightarrow q$ (Regla)

En lo expresado, las premisas dadas como p y q (Caso y Resultado) no muestran la forma para concluir en una expresión condicional, tal como se presenta en la conclusión (Regla). Para que tenga sentido lo expresado el condicional en forma de regla debe estar contenido en las observaciones particulares, dado que las observaciones particulares deberían estar contenidas en la regularidad que está expresada en la Regla, por lo tanto la notación debería mostrar la regularidad de cada caso particular observado, así la notación debe ser:

Si $p_1 \rightarrow q_1$ y $p_2 \rightarrow q_2$ y… $p_n \rightarrow q_n$

Entonces $p \rightarrow q$

En el campo de la lógica, la inducción presenta una interesante línea de

[35] Inferencia estadística. Razonamiento que aplica conceptos y emplea recursos del cálculo de probabilidades para sacar consecuencias acerca de una población o de una de sus partes. Se emplea sobre todo para inferir la probabilidad de los eventos de cierta clase de la frecuencia relativa con que se los ve ocurrir (Mosterín y Torretti, 2002:291).

investigación, así encontramos en la opinión en Newton da Costa[36].

> Las concepciones tradicionales de la razón evidenciaron impotencia para dar cuenta del nuevo estado de cosas, lo que está originando, como no podría dejar de ser, otra manera de encarar la inducción. Por esta circunstancia debemos ocuparnos de esas lógicas (Da Costa, 2000:22).

La investigación sobre la deducción y la inducción no está agotada, es más, si consideramos los avances obtenidos en el campo de la lógica heterodoxa[37] o lógica no clásica[38] y de las matemáticas, comprendemos lo acertado de la opinión de Newton Da Costa, en el sentido de que es un error el considerar que la deducción significa ir de lo general a lo particular y en sentido contrario si se trata de la inducción. Crítica la interpretación de que en la inducción las premisas no contienen la conclusión, como si fuera el Caso y el Resultado sin establecer la relación que está contenida en la Regla.

> … nada es más errado que afirmar que la inducción, al contrario de la deducción, va de lo particular a lo general, o tal vez, de lo menos general a lo más general...... También extravagante es la definición de inducción según la cual la inducción termina siendo una forma de raciocinio en que la conclusión no se halla contenida en las premisas, en oposición de la deducción (Da Costa, 2000:36).

Los juicios inductivos son de naturaleza probable (no contienen certeza) dado que se sustenta en la observación de algunos casos. Destacamos que la indución es de diferente naturaleza cuando se trata de situaciones en la que se conoce la estadística de probabilidad de las que no se tiene ningún conocimiento de su probabilidad. Las expresiones

[36] Newton C.A. Costa, (nacido en 1929, y graduado como ingeniero por la Universidad Federal del Estado brasileño de Paraná en 1952), matemático, ingeniero civil, filósofo y lógico, conocido por ser pionero en la investigación de la lógica para consistente.

[37] Según Miro Quesada, puede dividirse ante todo entre lógica clásica y lógica heterodoxa. La lógica clásica usa un lenguaje formal, el cual es asertórico y se atiende a los tres llamados grandes principios: de identidad, no contradicción y tercio excluso… La lógica heterodoxa es definible negativamente como una lógica, o algún tipo de lógica, que carece de algunas de las tres indicadas características. En la palabra lógica del tomo III (Ferrater Mora, 2004: 2185-2186)

[38] Según García Zárate, la lógica no clásica es la que se caracteriza bien por ser no asertórica como en el caso de la lógica normativa; primer por incorporar un lenguaje que no es de primer orden, como lo es la lógica modal…, bien por dejar de lado alguno de los tres principios lógicos fundamentales (Zárate, 2007:231)

probabilísticas de una estructura deductiva son inducciones dado que sus deducciones no son seguras, depende de la probabilidad del caso.

En cuanto a la inducción en relación a la regularidad de la naturaleza, consideramos que hay diferencia sustancial sobre las observaciones en la que la regularidad (la Regla) se encuentra presente en todos los casos, en contra de las observaciones que contienen parcialmente el cumplimiento de la regularidad, obteniéndose una conclusión de naturaleza probable, de la siguiente forma: "En 'n' casos se presenta lo observado del total de 'm' casos".

Peirce manifiesta que las fórmulas empíricas que se presentan en las ciencias naturales corresponden a un proceso de inducción, como sigue:

> Es posible, verbigracia, que siendo v el volumen relativo y t la temperatura, unas cuantas observaciones examinadas indiquen una relación de la forma: $v = 1 + a\,t + b^2 + c\,t^3$
> Tras examinar nuevas observaciones a otras temperaturas tomadas al azar, se confirma esta idea; y sacamos la conclusión inductiva (Peirce, 1878:80).

La inferencia del tipo inductiva no contiene la seguridad, dado que resulta ser la formalización de la ocurrencia de una Regla basada en la confirmación de su presencia en observaciones particulares. Se pensó que la inducción es un método heurístico para obtener conocimiento. Aunque desde diferentes puntos de vista, coinciden en reformular el problema de la inducción como justificación de teorías desde la evidencia empírica, tal como lo manifestaron Carnap[39] y Popper[40].

---

[39] Rudolf Carnap (1891-1970) nace en Rundsdort Alemania y muere en Estados Unidos, físico matemático, filósofo y lógico, miembro importante del circulo de Viena.
[40] Karl Popper (1902-1994) nace en Himmelhof en el distrito Ober Viena, físico, filosofó crítico del Círculo de Viena.

Popper pretende acabar con todo planteamiento que sea semejante a la inducción y lo que propone es que el científico debe adherirse a la hipótesis[41], porque las teorías no son demostrables, más bien son falsables[42]. En este caso, la crítica que se hace a este planteamiento es que una teoría soporta hasta cierto grado los argumentos que la falsean.

Si retrocedemos en el tiempo, podemos decir en términos muy generales que la filosofía contiene dos corrientes en cuanto al concepto de inferir, una iniciada por Descartes[43] denominada racionalismo y otra iniciada por Bacon[44] denominada empirismo. A la primera se le atribuye el método deductivo mientras a la segunda el método inductivo. Se suele descartar el método abductivo por considerarlo dentro de la etapa de construcción de la teoría, así tenemos lo propuesto por Hans Reichenbach[45] en el campo de estudio de la epistemología, que solo corresponde considerar a las teorías y que debe ignorarse las formas como fue formulada, tanto en los aspectos personales y los históricos sociales, considerando que le pertenecen a la psicología, historia o sociología, pero no a la ciencia.

El argumento de Reichenbach es insuficiente porque las teorías se van formulando y organizando según el sentido en que avanzan los resultados, así también se encuentran dentro de un contexto en correspondencia a múltiples factores. Reichenbach postula que

---

[41] Hipótesis entendida como proposición que requieren ser demostrada.

[42] K. R. Popper ha declarado que el llamado problema de la inducción, especialmente tal como ha sido formulado desde Hume, es insoluble: no se pueden justificar las inferencias inductivas sin caer en un círculo vicioso. En la palabra Falsabilidad del Tomo II (Ferrater Mora, 2004: 1213)

[43] René Descartes (1596-1659) nace en La Haye en Touraine, y muere en Estocolmo, fue filósofo, matemático y científico francés.

[44] Francisco Bacón (1561-1626) nace en Londres Entre sus obras *La Gran Restauración* (1620), en la que proponía una reformulación de las ciencias, propugnando la sustitución del método deductivo del *Órganon* aristotélico por un método inductivo que permita el mejor desarrollo de la ciencia.

[45] Hans Reichenbach (1891-1953). Nació en Hamburgo, estudió ingeniería en la *Technische Hochschule* de Stuttgart, matemáticas, física y filosofía en las universidades de Berlín, Gotinga y Múnich. En 1915 obtuvo el doctorado en filosofía en la Universidad de Erlangen. De 1920 a 1926 fue profesor en su propia escuela en Stuttgart, pasó a ser profesor de filosofía de la física en la Universidad de Berlín, permaneció hasta 1933 (despedido por los nazis). En este lapso de siete años se identificó con el Círculo de Viena. De 1933 a 1938 fue profesor en la Universidad de Estambul, y de 1938 a 1953, en la Universidad de California en EEUU.

una teoría es una reconstrucción racional y no interesa los eventos psicológicos y/o sociales que rodea a la investigación[46], resulta que lo que está en juicio es la teoría (no como fue hecha), pero consideramos que en la investigación y análisis del cómo fue hecha se encuentran elementos de naturaleza abductiva, que marca en cierto sentido a la investigación, resultando interesante el considerar el cómo fue elaborada la teoría, mostrándonos importantes conceptos de naturaleza psicológica y/o sociológica, que contribuyen a una mejor comprensión de los contenidos e intenciones en la formulación de la teoría. Si se rechaza esta afirmación se estaría en cierto sentido de acuerdo a la posición planteada por Popper en el denominado *Criterio de la demarcación de la ciencia*[47]. No es parte de esta investigación ahondar en este tema, pero si tomarlo en cuenta, dado que el silogismo abductivo o hipótesis tal como lo define Peirce está presente como una forma de inferencia, incluso es parte de ésta.

Bacón crítica el silogismo de la deducción Aristotélica, porque si se intenta una sólida investigación científica, se debe tomar en cuenta que existen una serie de prejuicios contenidos en las premisas que suelen colarse en nuestro conocimiento, por lo tanto, los errores serán derivados al resultado. Bacón en su libro, *Novun Organum*, publicado en 1620, señala cuatro tipos de prejuicios, que los llama ídolos (ídolos de la tribu[48], ídolos de

---

[46] Tomado de Reichenbach, del texto Tópicos en Epistemología. (Piscoya, 2009), en el capítulo de: *Las Tres Tareas de la Epistemología*.

[47] El criterio de la demarcación define los límites que configura el concepto "ciencia". Las fronteras entre el conocimiento científico y lo no científico, incluso entre la llamada pseudociencia. El planteamiento de este es conocido como *problema generalizado de la demarcación*. Lo que se intenta es encontrar criterios para poder decidir, entre dos teorías dadas, cuál de ellas es más "científica".

[48] Los *idola tribus*. Éstos se expresan en la tendencia intelectual a considerar que las cosas existen en un grado de orden y de igualdad mayor del que en realidad se encuentran. Los 'ídolos de la tribu' surgen también de la propia vida emocional humana, con la consiguiente falta de objetividad en el momento de valorar los argumentos a favor o en contra de un principio. Estos ídolos conducen, finalmente, a la falsedad porque se apoyan en los datos engañosos que proporcionan los sentidos.

la caverna[49], ídolos del foro[50] e ídolos del teatro[51]). Proclama el método inductivo como una generalización a partir de la observación de los casos particulares.

La abducción[52] es el término utilizado por Peirce como el primer paso de la inferencia durante la formación de una hipótesis, como si fuera una interrogación, como proposición sobre la que se tiene cierto grado de confianza. Se dice que es un término similar al de hipótesis (formulación de una idea a ser probada). Peirce afirma que las abducciones no son escasas en el razonamiento cotidiano, son inferencias probables, y por lo tanto, más débiles, porque asume los límites de la demostración inductiva y son menores al razonamiento deductivo en cuanto a la seguridad de lo que se afirma.

Peirce menciona que la abducción y la inducción tienen la característica de no ser totalmente ciertas. La diferencia entre ambas es que la inducción contiene la certeza de que los objetos observados poseen características comunes que forman parte de la conclusión (la regularidad de lo observado), en la abducción no participa la regularidad, así la conclusión de la abducción es diferente a lo observado, así nos dice el pensador:

> La gran diferencia entre la inducción y la hipótesis estriba en que la primera infiere la existencia de fenómenos iguales a los que hemos observado en casos similares, mientras que la hipótesis supone algo de tipo distinto a lo que hemos observado directamente, y con frecuencia algo que nos sería imposible de observar directamente (Peirce, 1878, 85-86)

---

[49] Los *idola specus*. Ídolos de la caverna, proceden de las características específicas de cada individuo: de sus gustos, su educación, sus ocupaciones, etc. Emergen, por tanto, de la subjetividad y velan la auténtica naturaleza de la verdad.

[50] Los *idola fori*. Ídolos del mercado, se originan por el contacto entre los hombres y derivan casi siempre del lenguaje. Causan un auténtico reino de la confusión, pues llegan a utilizar conceptos ilusorios para cosas inexistentes.

[51] Los *idola theatri*. Ídolos del teatro, son los que se derivan de las falsas teorías, que han engañado a los hombres de tal manera como los actores engañan a su público en el teatro.

[52] La teoría de la abducción desempeña un papel capital en la filosofía de Peirce. Peirce usó varios términos, además de abducción, reproducción, presunción, hipótesis, inferencia hipotética, pero el primero parece tener preferencia, posiblemente en Peirce y, desde luego, en sus comentaristas. En la palabra abducción del Tomo I (Ferrater Mora, 2004:13).

Notamos que la inducción puede estar en el campo de la abducción, en la medida que nuestras inferencias se alejan de la relación que establece la regularidad a la que estamos tratando de generalizar: “En consecuencia, cuando ensanchamos una inducción mucho más allá de los límites de nuestra observación, la inferencia participa de la naturaleza de la hipótesis” (Peirce, 1878, 86)

Una representación lógica de la abducción tendría como premisas la Regla y el Resultado, deduciendo el Caso, sería así:

Si $p \rightarrow q$ (Regla) y q (Resultado)

Entonces p (Caso)

Lo expresado es la forma de representación de una falacia[53] de la afirmación del consecuente, basado en la veracidad de la Regla y el Resultado debería concluirse el Caso. La notación adecuada debe indicarse como posibilidad, así estaríamos introduciendo la expresión de la conclusión en forma que se trata en la lógica modal[54]:

Si $p \rightarrow q$ (Regla) y q (Resultado)

Entonces $\Diamond P$ (Caso, es posible p).

[53] Falacia es un término en lógica que refiere a un defecto técnico en la fórmula aplicada de forma tal que hace inválida o confiable el resultado. Zárate nos dice: las falacias nos engañan, haciéndonos admitir como válidos razonamientos que no lo son (García Zárate, 2009: 199)

[54] Se dice lógica modal al sistema formal que se ocupa de las expresiones modales tales como "es necesario que" o "es posible que". Una página Web que instruye sobre este tipo de lógica es la del profesor Renato Lewin, profesor de matemáticas de la Universidad Católica de Chile /www.labmat.puc.cl/cursos/archivos/2003/1/MAT140S/1056116600/III.pdf Suele considerarse como sistemas de lógica modal, sistemas conocidos tales como la lógica deóntica, la lógica temporal, la lógica epistémica.

Encontramos adecuado citar a Peirce sobre la abducción lógica, mostrado en el documento de Fernando Soler[55] y de Ángel Nepomuceno[56]: “El hecho sorprendente, C, es observado. Pero si A fuera verdad, C sería aceptado como algo evidente. Por lo tanto, hay razón para sospechar que A es verdad” (Peirce 1903, 5, 189, en Soler Fernando y Nepomuceno Ángel, 2008). En la que un hecho C, sería aceptado como evidente, en relación a otra idea A que es plausible, por lo tanto concluimos que A es como posibilidad.

La inducción y la deducción se encuentran relacionadas con la abducción, porque para definir las premisas se utiliza en cierta medida la abducción. Peirce considera que la abducción se presenta en primera instancia y luego la inducción como consecuencia de las pruebas de casos particulares, seguidamente formalizar la Teoría y aplicar la deducción, como consecuencia de la utilización de la Teoría.

En los silogismos[57] encontramos la explicación de construir argumentos para una demostración lógica y/o matemática, las incluimos en el marco teórico, dado que se encuentran implícitamente en las demostraciones de temas a ser expuestos más adelante.

En los mecanismos de tipo lógico que están presentes en los programas de computadora contienen reglas de la forma $p \rightarrow q$. Dado el Caso (p), es posible obtener el Resultado (q) y también el deducir a la inversa, pero con un procedimiento distinto.

---

[55] Fernando Soler Toscano, Doctor en filosofía, ingeniero técnico en Informática, profesor e investigador en el departamento de Filosofía, Lógica y Filosofía de la ciencia en la Universidad de Sevilla España, con diversas publicaciones sobre temas relacionados a la inteligencia artificial y la lógica, su tesis doctoral versa sobre lenguajes formales y su uso en el campo de la inteligencia artificial.

[56] Ángel Nepomuceno Toscano, profesor e investigador de Lógica y Filosofía en la Universidad de Sevilla, con diversas publicaciones sobre lógica y lenguaje, estudioso de Frege.

[57] García Zarate define el silogismo: “Como un tipo especial de razonamiento deductivo y mediato, que consta solamente de tres proposiciones: dos premisas y su correspondiente conclusión. Aristóteles es el creador de la doctrina del silogismo. El filósofo distinguió tres tipos de silogismos categóricos, modales e hipotéticos.”

Lo que no es posible obtener (a menos que este prefijado explícitamente en el procedimiento) es la Regla: (p →q). La Regla está contenida en el programa (en las instrucciones), la modificación de las reglas, significa teorizar sobre programas que "aprenden" (tema que fuera tratado por Turing). La lógica en los programas de computadoras utilizan en forma condicional las reglas. Estas formas de lógica son denominadas lógicas condicionales[58] y en los formalismos no monótonos encontramos un interesante campo de investigación. Consideramos que la abducción es una forma de deducir, válido para obtener conclusiones, con la característica que contienen mayor incertidumbre que las de tipo inductivo.

1.3. La inducción matemática.

La inducción matemática es un procedimiento para hacer demostraciones que se sustentan en las propiedades de los números naturales. Prueba la validez de una fórmula mediante la ejecución de tres pasos; el primero: verifica que la fórmula cumple para el número más pequeño (suele ser la mayoría de los casos el número 1); segundo: se asume que la fórmula cumple para un número 'n', es decir, la fórmula es correcta para la variable; finalmente, el tercero: Consiste en probar que la fórmula cumple para el número sucesor es decir para 'n +1', ésta expresa la fórmula en términos del número sucesor, y si así ocurre, se afirma que se ha demostrado la fórmula.

Como ilustración de lo mencionado demostraremos que la suma de los 'n' primeros números es igual a $\frac{n*(n+1)}{2}$. Empezamos calculando el valor de la fórmula para n

[58] En el texto de *Lógicas Condicionales y Razonamiento de Sentido Común*, Gladys Palau y colaboradores, tratan diversas formas sobre lógicas condicionales y sobre los formalismos no monótonos.

igual a 1 (uno) y evaluamos en la fórmula, resulta el valor 1, que se obtiene de reemplazar 1 en 'n' (cumple con el primer paso), luego afirmamos que la fórmula es correcta, como hipótesis que se incluye en la premisa (segundo paso). Ahora demostramos para el número 'n+1' (es el tercer paso, la última parte de la demostración), que resulta reemplazar 'n+1' en 'n', obteniendo la expresión $\frac{(n+1)*(n+2)}{2}$, concluimos que cumple con lo estipulado.

Hubert Kennedy[59] nos dice que Peano[60] entendió que el método axiomático permite hacer teorías matemáticas, que son de fácil aprendizaje. (Kennedy, 2002:39). En su artículo *Twelve Articles in Giuseppe Peano*, presenta los axiomas utilizados por Peano para la definición de los números naturales, basado en una publicación póstuma de 1958:

(1) Uno es un número.
(2) El signo + colocado después de un número produce un número.
(3) Si a y b son dos números, y si sus sucesores son iguales entonces estos son iguales.
(4) Uno no es sucesor de ningún número.
(5) Si s es una clase que contiene uno, y si la clase está hecha por los sucesores de s, está contenida en s, entonces todo número está contenido en la clase s.

Se puede reconocer que el axioma (2) es una definición recursiva de un número natural, define el número siguiente en relación al número anterior. El axioma (5) presenta

[59] Hubert Kennedy, nacido en Florida Estados Unidos, 1931. Profesor de Matemáticas en la Universidad de Providence (Rhode Island), donde su interés por la investigación fue la historia de las matemáticas. Tiene más de 200 publicaciones en varios idiomas. Resalto sus escritos sobre la biografía del matemático italiano Giuseppe Peano.

[60] Giuseppe Peano (1858-1932) matemático y filósofo italiano, conocido por sus contribuciones a la teoría de conjuntos, publicó más de doscientos libros y artículos, la mayoría en matemáticas. La mayor parte de su vida la dedicó a enseñar en la ciudad de Turín Italia.

el concepto de clase[61] como un conjunto que contiene a todos los números.

Peano presentó varias formas de axiomatización de los números naturales, así afirma Hubert Kennedy indicando que en el documento de Peano *Formulaire de mathématiques,* de 1901, el número inicial de los números naturales es el cero y no el número uno como lo enunció en 1889. Esto lo entendemos, debido a que Peano buscó definir el concepto de número.

La inducción matemática está relacionada con la Axiomática de Peano, así según Kennedy refiere al sentido utilizado por Peano cuándo trata de su quinto axioma: "un número es el número anterior más uno", es una definición de una función matemática, esta teoría es presentada en más detalle su libro *Arithmetices Principia Nova Methodo Exposita,* en 1889.

Cuando Peano presentó sus axiomas, consideró que su trabajo no era una respuesta a la pregunta ¿qué es un número? sino más bien, es un esfuerzo hacia la axiomatización (obtención de proposiciones verdaderas), así lo indica Huber Kennedy en su artículo *The Mathematica Philosophy of Giuseppe Peano*, ésta es una traducción de la cita de Peano en la que dice: "Estos conceptos (número, la unidad, el sucesor de un número) no se pueden obtener por deducción, es necesario obtenerlos por inducción (abstracción)" (Peano, 1891:85).

El caso es que según Kennedy, Peano se dedicó a la axiomatización, y no utilizó la teoría lógica matemática que fuera desarrollada por él mismo:

[61] Se entiende como clase al conjunto de elementos que tienen la particularidad de que cualquier elemento representa a todos.

> Hemos visto que el principal interés estaba en su axiomática, que nunca usó la lógica matemática desarrollada por él para la reducción de los conceptos matemáticos a conceptos lógicos, y que, en cambio, negó la validez de dicha reducción. (Kennedy, 2002: 11).

En cuanto a las formas de inferencias enunciadas como silogismos, la inducción matemática es una deducción, es entendible que así resulta en la medida que la demostración para el caso 'n+1' se encuentra contenida en el caso 'n', es decir la Regla y el Caso están presentes en las premisas, y lo que se está demostrando es el Resultado para el caso 'n+1', (en términos dados por Peirce). Se puede expresar la inducción matemática de la siguiente forma:

Fórmula(n) → Fórmula(n+1) (Regla, todo número es el anterior más uno)

Fórmula (n) (Caso, se acepta como válida la fórmula)

Por lo tanto: Fórmula (n+1) (Resultado, se obtiene de las dos premisas)

El concepto de número y el deseo de formalizar la aritmética son parte de los trabajos de investigación de Peano, siendo importante en la historia de las matemáticas, porque contribuyó a una teoría que luego tendría participación en la definición de expresiones de las funciones recursivas, así tenemos según Kennedy: "En 1898 la definición recursiva de adición requería de dos ecuaciones: $a + 0 = a$, $a + (b^+) = (a + b)^+$" (Kennedy, 2002:42).

La notación b^+ significa el siguiente número de b, la segunda fórmula dice "la suma de un número con el siguiente de otro número es igual al siguiente número de la suma de dos números", en esta expresión se remarca la definición recursiva.

1.4. Recursividad.

La recursividad de las funciones matemáticas, trata sobre funciones[62] que en su definición contiene su propia definición, es una definición que se contiene así misma.

Para profundizar con lo expresado, revisamos el concepto de función matemática como una relación entre los elementos de dos conjuntos: Dominio y Rango, a un valor del Dominio le corresponde según la definición de la función un valor en el Rango.

Como ilustración, la función de la suma de los n primeros números, se expresa de la siguiente forma: h (n) = $\frac{n*(n+1)}{2}$, y la evaluación para n = 5 seria h (5) = 15, para el caso, 5 pertenece al Dominio de la función y el valor 15 pertenece al Rango.

La composición de funciones se refiere a una aplicación de dos o más funciones, una después de la otra, en la que a un elemento del Dominio de la primera función aplicada, le corresponde un elemento del Rango de la segunda función aplicada. Para el caso, tenemos las dos funciones f y g:

g: $\text{Dominio}_g \rightarrow \text{Rango}_g$

f: $\text{Dominio}_f \rightarrow \text{Rango}_f$

La composición de las dos funciones (g y f) obtienen otra función, h, esta se describe de la forma f ° g, donde se expresa como h: $\text{Dominio}_h \rightarrow \text{Rango}_h$

[62] La noción de función recursiva precisa la noción de función computable, como la relación de conjuntos en que su evaluación requiere de valores obtenidos de pasos anteriores, en forma similar a entender un número natural expresado en relación a la suma de números anteriores.

Por ser una composición de funciones se tendría dos relaciones que parte del Dominio de g hacia el Rango de f: $f \circ g : Dominio_g \rightarrow X \rightarrow Rango_{f,}$

Donde X es parte del Rango de la función de g y también parte del Dominio de la función de f, entonces: $X \subseteq (Rango_g \cap Dominio_f)$.

Resulta que X es un subconjunto de la intersección del Rango de la función g con el Dominio de la función f. En un sentido más riguroso el Dominio de la función h es el subconjunto del Domino de la función g y el Rango de la función h es un subconjunto del Rango de la función f.

Supongamos que tenemos las funciones $g(x) = x^2$, y $f(x) = 2x+1$, entonces la función resultante de la composición de las funciones f, y g, seria: $h(x) = f(x) \circ g(x)$, que resulta de aplicar la función g y luego la función f, por eso suele también escribirse de manera equivalente $h(x) = f(g(x))$.

Si calculamos h (3) sería equivalente evaluar f (g (3)), esto significa calcular en dos pasos: Primero f (g (3)) = f (9) y luego f (9) = 19, entonces f (g (3)) = 19. Expresando en términos de la variable x, tenemos que $h(x) = f(g(x))$, reemplazando según definición de la función de g, se tendría $h(x) = f(x^2)$, y reemplazando nuevamente según definición de la función f, se tendría $h(x) = 2x^2+1$.

La función recursiva se define mediante la composición de funciones, en la que al menos una de las funciones que participa de la composición es en cierto sentido la misma función. Es mediante la composición de funciones donde se muestra la mayor capacidad expresiva de las funciones recursivas.

Para la definición de una función recursiva, se requiere establecer como definición por lo menos una condición inicial y reglas de naturaleza recursiva.

Como ejemplo, el cálculo factorial de un número, que consiste ser la multiplicación de todos los números naturales menores e igual al número, así por ejemplo el factorial de 5 es 5×4×3×2×1 que resulta 120. Si nos fijamos, 4×3×2×1 es el factorial de 4, entonces el factorial de 5 es igual al factorial de 4 por el número 5.

El factorial expresado mediante una función recursiva, define como condición inicial el factorial de 1 igual a 1; luego el factorial de otro número que es la regla recursiva, que dice que el factorial de un número, es el factorial del número anterior multiplicado por el número, así tendremos:

fact (1) = 1 (Condición inicial)

fact (n) = n × fact (n-1) Para n > 1 (Definición recursiva)

Las funciones recursivas utilizadas para expresar cálculos, exigen que para su evaluación requieran de definiciones de funciones que ya fueron evaluadas. Esto se conoce como "procedimiento" de computar, dado que para el cálculo de un determinado argumento, se recurre a obtener valores de funciones definidas en las condiciones iníciales.

Al respecto Mosterín y Torretti[63] nos dicen:

> La noción exacta de función recursiva precisa la noción intuitiva de función computable. La denominación de 'recursiva' alude al procedimiento de computar el valor de una función para un número recurriendo a sus valores para los números menores que él. La definición de función recursiva requiere diversas definiciones previas (Mosterín y Torretti, 2002:258).

Las funciones recursivas fueron introducidas por Gödel[64], el año 1931, en la demostración del teorema de la incompletitud, utilizándolas en un sentido diferente al empleado por Church, en 1936, que la presentó con el nombre de cálculo lambda. Gödel define la función recursiva de la siguiente forma:

> Decimos que una función numérica $f(x_1, x_2,\dots, x_n)$ está recursivamente definida a partir de las funciones numéricas $h(x_1, x_2,\dots, x_{n-1})$ y $q(x_1, x_2,\dots, x_{n+1})$ si para cada $x_2,\dots, x_n$, k vale lo siguiente:
>
> $f(0, x_2,\dots, x_n) = h(x_2, \dots, x_n)$,
>
> $f(k+1, x_2,\dots, x_n) = q(k, f(k, x_2, \dots, x_n), x_2, \dots x_n)$ (Gödel en 1931 en Mosterín 2006:62)

Como se observa, la primera ecuación es la condición inicial y la siguiente ecuación es una definición para el término 'k+1' en relación al término 'k'.

Kleene en su libro *Introduction to Metamathematics,* de 1952, define las funciones

---

[63] Roberto Torretti, nació el 15 de febrero de 1930 en Santiago de Chile, Chile. Filósofo chileno, escritor y académico reconocido por sus contribuciones a la historia de la filosofía, física y matemáticas. Ph.D. en Filosofía en la universidad de Freiburg, Alemania, 1954. Su obra *Manuel Kant. Estudio de los fundamentos de la filosofía crítica,* es considerada como una de las más importantes obras literarias sobre los pensamientos de Kant, además de publicaciones como *Philosophy of Geometry from Riemann to Poincaré* (1978), *Relativity and Geometry* (1983) y *El paraíso del cantor* (1998). Gran parte de las obras de Torretti están enfocadas en la física y las matemáticas, con un enfoque mayor en la teoría de la relatividad y en la geometría del siglo XIX.

[64] Kurt Gödel, nace el 28 de abril de 1906 en Brno, Imperio austrohúngaro (ahora República Checa) y muere el 14 de enero de 1978 Princeton, New Jersey. Lógico, matemático y filósofo austriaco-estadounidense. Se le conoce principalmente por sus dos teoremas de la incompletitud, publicados en 1931 a los 25 años de edad, un año después de finalizar su doctorado en la Universidad de Viena. El más célebre de sus teoremas de la incompletitud establece que para todo sistema axiomático recursivo auto-consistente lo suficientemente poderoso como para describir la aritmética de los números naturales (la aritmética de Peano), existen proposiciones verdaderas sobre los números naturales que no pueden demostrarse a partir de los axiomas. Demostró que la hipótesis del continuo no puede refutarse desde los axiomas aceptados de la teoría de conjuntos. Realizó importantes contribuciones a la teoría de la demostración.

recursivas primitivas, que resulta definiciones de funciones muy simples, así:

> Cada una de las ecuaciones y sistemas de ecuaciones siguientes (I)-(V) define una función de teoría de números φ, cuando n y m son enteros positivos, i es un número tal que $1 \leq i \leq n$, q es un número natural, y ψ, $x_1, \ldots x_m$, x son funciones en la teoría de números dadas con el indicado número de variables.
>
> (I) $\varphi(x) = x'$
>
> (II) $\varphi(x_1, \ldots, x_n) = q$
>
> (III) $\varphi(x_1, \ldots, x_n) = xi$
>
> (IV) $\varphi(x_1, \ldots, x_n) = \psi(\chi_1(x_1, \ldots, x_n), \ldots, \chi_m(x_1, \ldots, x_n)$
>
> (Va) $\varphi(0) = q$, $\varphi(y') = \chi(y, \varphi(y))$
>
> (Vb) $\varphi(0, x_2, \ldots, x_n) = \psi(x_2, \ldots, x_n)$, $\varphi(y', x_2, \ldots, x_n) = \chi(y, \varphi(y, x_2, \ldots, x_n), x_2, \ldots, x_n)$ (Kleene, 1952:203)

Las funciones recursivas primitivas son utilizadas para definir otras funciones recursivas. La ecuación (I) es la función del número siguiente (en el mismo sentido del quinto axioma formulado por Peano), la ecuación (II) es la función constante (que es una función que siempre calcula el mismo valor), la ecuación (III) es la función que proyecta un elemento del dominio.

Las ecuaciones (IV), (Va) y (Vb) definen el concepto de función recursiva utilizando la composición de funciones, en el caso (Va) es para una variable y la (Vb) es para más de una variable, Kleene nos dice:

> Las ante dichas ecuaciones y pares de ecuaciones (I) – (V), serán consideradas por nosotros como esquemas. Su función es análoga a la de los postulados, que desempeña (I)-(III) el papel de esquemas axiomáticos… y (IV) y (V) el papel de reglas de inferencia (Kleene, 1952:203).

Las funciones iniciales definidas en (I), (II) y (III) tienen correspondencia a expresadas como: función sucesivo $s(x) = x + 1$; función nula $n(x) = 0$ y función proyección de $u^i(x_1, \ldots, x_n) = x_i$.

En su libro de 1952, Kleene presenta algunos ejemplos que ilustran el sentido de computabilidad mediante las funciones recursivas, así tenemos la suma de dos números:

$$\begin{cases} a + 0 = a \\ a + b' = (a + b)' \end{cases}$$

La potencia de dos números:

$$\begin{cases} a^0 = 1 \\ a^{b'} = a^b * a \end{cases}$$

La suma de los números expresado según las funciones iniciales:

$$\begin{cases} \varphi\ (0, a) = a \\ \varphi\ (b', a) = (\varphi\ (b, a))' \end{cases}$$

La potencia de dos números expresada según las funciones iniciales:

$$\begin{cases} \varphi\ (0, a) = 1 \\ \varphi\ (b', a) = \varphi\ (b, a) * a \end{cases}$$

Kleene incorpora la definición de la función recursiva primitiva como aquella función que es una composición finita de funciones iniciales o dependiente de funciones precedentes, así tenemos:

> Una función φ es denominada una función inicial si φ satisface la ecuación (I) o la ecuación (II) para unos particulares n y q, o la ecuación (III) para unos particulares n e i… Una función φ es denominada una dependiente inmediata de otras funciones si φ satisface la ecuación (IV) para unos particulares n y m, siendo ψ, $\chi_1, \ldots , \chi_m$ las otras funciones, o las ecuaciones (Va) para un q particular, siendo χ la otra función, o las ecuaciones (Vb) para un n particular, siendo ψ, χ las otras

funciones (Kleene, 1952:203)

Ejemplo de funciones recursivas primitivas (en los números naturales), expresadas según funciones iniciales. Para la suma: x + y, la función sería equivalente a la composición de funciones, en la que u es la función proyección y s es la función sucesivo:

$$\begin{cases} f(x, 0) = u^1(x) \\ f(x, y+1) = s(f(x, y)) \end{cases}$$

Para el producto: x * y, la función recursiva según la composición de funciones en la que f es la función suma y n es la función nula:

$$\begin{cases} h(x,0) = n(x) \\ h(x, y+1) = f(h(x,y), x) \end{cases}$$

Para el caso del exponente x^y, la función recursiva sería equivalente a las siguientes funciones, en la que h es la función multiplicación y s la función sucesiva:

$$\begin{cases} e(x, 0) = s(0) \\ e(x, y+1) = h(e(x, y), x) \end{cases}$$

Hemos tratado sobre las funciones recursivas iniciales y las funciones recursivas primitivas en los números naturales, éstas se sustentan en la definición del número, tal como menciona Kleene:

> Cuando escribimos la secuencia de números naturales 0, 1, 2, 3, … empezamos por describir a los números naturales como aquellos objetos que pueden ser generados comenzando con un objeto inicial 0 (cero) y pasando sucesivamente de un objeto n ya generado a otro objeto n+1 o n’ (el sucesor de n) … resta solo explicar que 0, 1, 2, 3… hacen las veces de 0, 0’, 0’’, 0’’’ … respectivamente (Kleene, 1952:29).

El concepto de sucesor es básico en la definición – para Kleene los números naturales empiezan en el número 0 –, y permiten definir las operaciones como suma, multiplicación entre otras.

Kleene afirma que las funciones recursivas primitivas tienen correspondencia con la axiomatización de los números naturales realizada por Peano, como sigue:

> Estas cinco proposiciones 1-5, salvando una diferencia, fueron adoptadas por Peano (1889, 1891) como axiomas caracterizadores de la secuencia de números naturales. Dicho autor estableció por su parte la proposición 3 como el principio de inducción matemática, y la colocó en el quinto lugar de su lista (Kleene, 1952:30).

En la definición de las funciones recursivas primitivas se entiende intuitivamente el concepto de inducción matemática. En el caso de Gödel la recursividad tiene un sentido más potente dado que corresponde a relaciones, entendemos así dado que una función es un tipo particular de relación, así enuncian Mosterín y Torretti:

> En sus correrías aritméticas Gödel se limita básicamente a una clase especialmente manejable de relaciones y funciones numéricas: las relaciones y funciones recursivas primitivas, definidas por primera vez en este artículo y cuyo estudio daría lugar más tarde a la teoría de la recursión (Mosterín y Torretti, 2002:46).

Gödel utiliza para la demostración del teorema de la incompletitud las funciones recursivas y las relaciones. Esta definición es importante dejarla resaltada, dado que las funciones recursivas permiten un cálculo que obtiene un valor, mientras que en una relación se define la correspondencia de un valor hacia varios valores.

Gödel, en 1931, define la función recursiva introduciendo un operador llamado minimizador, así ж, η representan n-tuplas, en la que cumple:

Si la función f(ж) y la relación Rx, η son recursivas primitivas, también lo son las relaciones S, T definidas por:

$S(ж, \eta) \leftrightarrow \exists x \, (x \leq f(ж) \wedge Rx, \eta)$

$T(ж, \eta) \leftrightarrow \forall x \, (x \leq f(ж) \rightarrow Rx, \eta)$

Así como la función $q(ж, \eta) = \mu\, x \, (x \leq f(ж) \wedge Rx, \eta)$ donde $\mu\, x\, \varphi(x)$ significa el mismo número x para el que vale $\varphi(x)$, si hay algún tal, y 0, si no lo hay (Gödel en Mosterín, 2006:63)

El operador mínimo (μ) lo enuncia Mosterín y Torretti de la siguiente forma:

El operador μ (el mínimo… tal que) nos permite referirnos al mínimo número x que satisface la condición Φ, $\mu x\Phi$. Si hay algún número que satisface Φ, y para cada número natural x es decidible si Φ(x) o no entonces $\mu x\Phi(x)$ es computable. Decimos que una función n-aria h es definible por minimalización a partir de una función (n+1)-aria f en caso normal si y sólo si para cada $x_1, \ldots x_n$ existe al menos un w tal que $f(x_1, \ldots, x_n, w) = 0$ y ocurre que para cada $x_1, \ldots, x_n$: $h(x_1, \ldots, x_n) = \mu\, w\, [\, f(x_1, \ldots, x_n, w) = 0\,]$ (Mosterín y Torretti, 2002:259).

Kleene indica que Gödel utiliza el concepto de función recursiva general en referencia a la investigación realizada por su amigo Herbrand[65]. La definición de la función recursiva en Gödel no toma en cuenta el concepto de las funciones recursivas primitivas (constituyendo una diferencia importante en la teoría de la recursividad) las define como funciones recursivas generales:

La caracterización de todas las 'funciones recursivas' fue llevada a cabo, mediante la definición de 'función recursiva general' por Gödel 1934, quien se basó en una sugerencia de Herbrand. Esta definición tuvo éxito merced a una atrevida generalización (Kleene, 1952: 250).

Cuando Kleene refiere a la definición elaborada por Gödel, está considerando lo mencionado por el ilustre matemático en 1934, en donde refiere a que mediante la función

[65] Jacques Herbrand, nace en Paris el 12 de febrero, 1908 y fallece en La Bérarde, Isère, Francia el 27 de julio, 1931. Matemático.Matemático francés, que trabajó en lógica matemática. Egresado de la Escuela Normal Superior de París. Pasó un período en la Universidad de Göttingen. Introdujo la noción de función recursiva. Terminó su doctorado en la École Normale Supérieure de París. Se le concedió una beca Rockefeller que le permitió estudiar en Alemania en 1931, primero con John von Neumann en Berlín. Murió a los 23 años en un accidente de montaña en los Alpes.

recursiva es posible construir una expresión aritmética, introduciendo el valor que se obtiene del cálculo de una función recursiva.

> Usando esta definición de la noción de función recursiva podemos probar que si f $(x_1, \ldots, x_n)$ es recursiva, entonces hay una expresión aritmética α $(x_1, \ldots, x_n)$ talque f $(x_1, \ldots, x_n) = y \leftrightarrow$ α $(x_1, \ldots, x_n, y)$ (Gödel en Mosterín, 2006:196).

La definición de minimización en las funciones recursivas proporciona otra forma de definir funciones recursivas, Kleene la denominó el operador μ. "φ $(x_1, \ldots, x_n) = \mu$ y $[\chi(x_1, \ldots, x_n, y) = 0]$" (Kleene, 1952: 255).

El operador de minimización relaciona las ecuaciones de forma tal que define un conjunto de posibles valores, es el mínimo valor que resulta ser el que cumple con la condición de cálculo.

Ejemplo, al calcular el valor entero de $\frac{7}{3}$, sabemos que el resultado es 2. Éste se obtiene de la definición de la función valor entero denotado $[\frac{x}{y}]$ y el resultado es el número entero de la división de los números implicados.

La función mínimo para el valor entero:

$$[\frac{x}{y}] = \text{Min } \{t \leq x \ / \ (t+1) * y > x\}$$

En el caso de los números 7 y 3 tenemos:

$$[\frac{7}{3}] = \text{Min } \{t \leq 7 \ / \ (t+1) * 3 > 7\}$$

$$[\frac{7}{3}] = \text{Min } \{2, 3, 4, 5, 6, 7\}$$

El resultado es 2 por ser el valor mínimo

$$[\tfrac{7}{3}] = 2$$

Otro ejemplo, al calcular el residuo de $\frac{7}{3}$, sabemos que es 1. El resultado se obtiene de la definición de residuo según el algoritmo del residuo:

Si $[\frac{x}{y}]$ es el cociente de la división y, $\text{Res}(\frac{x}{y})$ es el residuo, entonces:

$$x = y * [\tfrac{x}{y}] + \text{Res}(\tfrac{x}{y})$$

Definimos como:

$$\text{Res}(\tfrac{x}{y}) = \text{Min } \{ t < y \,/\, (t+1) > x - y * [\tfrac{x}{y}] \}$$

En el caso de los números 7 y 3, tenemos:

$$\text{Res}(\tfrac{7}{3}) = \text{Min } \{ t < 3 \,/\, t+1 > 7 - 3 * [\tfrac{7}{3}]\}$$

$$\text{Res}(\tfrac{7}{3}) = \text{Min } \{ t < 3 \,/\, t+1 > 7 - 3 * 2\}$$

$$\text{Res}(\tfrac{7}{3}) = \text{Min } \{t < 3 \,/\, t > 0\}$$

$$\text{Res}(\tfrac{7}{3}) = \text{Min } \{1, 2\}$$

$$\text{Res}(\tfrac{7}{3}) = 1$$

Kleene explica la relación entre recursividad y la inducción matemática, considerando que una función es recursiva general si hay un sistema de ecuaciones que la definen recursivamente, de tal forma que la recursividad trata de procesos recurrentes que se basan en la inducción matemática: “Esta elección puede parecer inesperada, puesto que la palabra “recursivo” tiene su raíz en el verbo “recurrir”, y la inducción matemática es nuestro método para tratar procesos recurrentes” (Kleene, 1952: 251).

En la opinión de Kleene la recursividad de las funciones es la misma que se expresa mediante procesos recurrentes, para esclarecer esta expresión nos valemos de la opinión de Peirce en su artículo *On the logic of number,* de 1881, en la que relaciona dos conjuntos que tienen el mismo número de elementos, indicando que ambos conjuntos son finitos. Es en la cantidad de elementos de un conjunto donde el concepto de número natural resulta definible:

> Si todo S es un P y si los P son una agrupación finita que cuenta hasta un número tan pequeño como el número de los S, entonces todo P es un S. Porque si, contando los P, empezamos con los S (que son una parte de ellos), y habiendo contado todos los S llegamos al número n, no quedaran ni P ni S. Pues si hubiera alguno el número de los P contaría hasta más que n (Peirce, 1881).

Lo expuesto por Peirce resulta similar a la definición de número dada por Bertrand Russell, al mencionar la correspondencia entre los conjuntos que tienen la misma cantidad de elementos. Esta forma de incluir una definición en otra definición, es común en temas matemáticos, así lo indica en un ejemplo al final del mismo artículo, que refiere a relaciones entre texanos. Se explica como sigue:

> Todo texano mata un texano, nadie es muerto por más de una persona. Por lo tanto, todo texano es muerto por un texano, suponiendo que los texanos son una agrupación finita. Porque por la primera premisa. Todo texano muerto por un texano es un texano asesino de un texano. Por la segunda premisa, los texanos muertos por texanos son tantos como los texanos asesinos de texanos. De donde concluimos que todo texano asesino de un texano es un texano muerto por un texano, o, por la primera premisa, todo texano es muerto por un texano. Este modo de razonamiento es frecuente en la teoría de los números (Peirce, 1881).

1.5. Incompletitud.

Se suele entender por incompletitud a la ausencia de algo que hace falta para ser completo, tiene un significado contrario al de completitud. Aquí se presenta el concepto en correspondencia a un sistema axiomático, en el sentido que es incompleto cuando no se puede demostrar en el sistema la veracidad o la falsedad de cierta proposición que le pertenece al sistema.

Un sistema axiomático cumple con ciertos requisitos para que se diga que está bien definido, estas características básicas son las siguientes: consistente, independiente, adecuado y completo.

Es consistente si desde los axiomas deducen teoremas que no tienen contradicción con los axiomas o con otros teoremas, específicamente que los axiomas no sean contradictorios y que no propicien la contradicción. Tarski explica al respecto:

> Llamaremos consistente a una disciplina deductiva cuando no haya en ella dos enunciados que se contradigan mutuamente, o, con otras palabras: cuando dos enunciados contradictorios de ella, uno al menos no pueda demostrarse... llamaremos completa o integra cuando dos proposiciones formuladas en la misma, con ayuda exclusiva de expresiones de esta y de las disciplinas precedentes, y contradictorias entre sí, una al menos de ambas pueda demostrarse (Tarski, 1951:147).

Decimos que es independiente cuando los axiomas puede deducirse de los otros, resulta evidente que en la medida que simplifiquemos los axiomas se pierde claridad intuitiva[66].

[66] Tarski en su libro *Introduction to Logic and to the Methodology of Deductive Sciences* presenta un conjunto de axiomas para definir la aritmética de los números reales y muestra como los simplifica hasta encontrar un mínimo sistema axiomático equivalente.

Es adecuado[67] cuando el sistema axiomático corresponde a la naturaleza de lo que se busca representar, constituyendo la idea de lo que representa, está en referencia a la semántica de la teoría, actúa como una teoría que permite deducir los teoremas de la teoría, en correspondencia a lo que ha sido teorizado.

Decimos que es completo cuando se puede deducir desde sus axiomas todas las proposiciones verdaderas o falsas en el sistema, y en sentido contrario, es incompleto cuando se encuentra una proposición verdadera o falsa en el sistema axiomático que no se puede deducir.

En relación al concepto de incompletitud, se inició la base de lo que resultaría la ciencia de la computación, específicamente porque contribuyó a configurar el concepto de computabilidad, que en una primera instancia se materializó en la tesis de Church-Turing. En 1900, David Hilbert propone demostrar que las matemáticas son un sistema axiomático consistente, tomando como base lo propuesto por Cantor[68], en este sentido en que Berlinski[69] expone:

> George Cantor había defendido, veinte años atrás, consistencia como el único estándar probatorio para todas las matemáticas, las creaciones libres de la mente humana... porque no hacían ningún daño. Hilbert ahora demandaba que esa idea lírica fuera puesta bajo el control de una demostración matemática (Berlinski, 2007:182).

---

[67] Los escolásticos llaman adecuada a la idea que posee una correspondencia exacta con la propia naturaleza de la cosa objeto de la idea… Las ideas adecuadas son completas, es decir exhiben claramente las notas constitutivas del objeto ideado. En la palabra adecuado Tomo I (Ferrater Mora, 2004:61).

[68] Georg Cantor, nacido en San Petersburgo, 3 de marzo de 1845, fallece en Halle, 6 de enero de 1918. Matemático alemán, uno de los inventores de la teoría de conjuntos. Gracias a sus atrevidas investigaciones sobre los conjuntos infinitos fue el primero en *formalizar* la noción de infinito bajo la forma de los números transfinitos (cardinales y ordinales). Murió en una clínica psiquiátrica de monjas, aquejado de una enfermedad maníaco-depresiva.

[69] David Berlinski, nace en 1942 en Nueva York. Filósofo y matemático. Recibió su Ph.D. en filosofía en Princeton, becario postdoctoral en matemáticas y biología molecular en Universidad de Columbia. Profesor de filosofía, matemáticas en Stanford, Universidad de Nueva York y la Universidad de París.

En 1910 Bertrand Russell y Alfred Whitehead[70] publican el primer volumen de *Principia Mathematica,* con el objetivo de demostrar que los principios de la aritmética se podían derivar de la lógica, esto se encontraba en cierto sentido dentro del programa propuesto por Hilbert, al respecto resalta Berlinski: "Russell escribió en términos conmovedores su deseo, casi desesperado, de encontrar reposo para sus pensamientos turbulentos en una estructura matemática de certidumbre perfecta. Los *Principia Mathematica* eran la expresión de esta necesidad..." (Berlinski, 2007:183).

El concepto de la incompletitud refiere a imposibilidad de lograr la axiomatización de la matemática. Esta conclusión es significativa en el campo de la lógica y la filosofía de las matemáticas, e influye en diversos campos del conocimiento. La demostración de la imposibilidad de la axiomatización fue realizada por Gödel en 1930 y publicada en 1931, en la que prueba utilizando funciones recursivas, que en un sistema formal de cierto rigor siempre será posible encontrar una proposición verdadera que no puede ser demostrada en el mismo sistema. Gödel prueba que la aritmética es incompleta. Cabe destacar que la incompletitud de la aritmética no descarta su utilización, y que continúa su desarrollo en la definición de nuevas estructuras matemáticas.

[70] Alfred North Whitehead, (Ramsgate, 15 de febrero de 1861 - Cambridge, Massachusetts, 30 de diciembre de 1947), fue un matemático y filósofo inglés. Publicó trabajos sobre álgebra, lógica, fundamentos de las matemáticas, filosofía de la ciencia, física, metafísica, epistemología y educación. El trabajo más conocido, del que es coautor con Bertrand Russell, es *Principia Mathematica*. Fue profesor en las universidades de Londres y de Cambridge, donde destacó por sus estudios lógico-matemáticos. Luego en Estados Unidos fue director de la cátedra de filosofía en la Universidad de Harvard, y tuvo entre sus discípulos a Quine.

De otro lado Penrose[71] afirma que suele haber una interpretación equivocada del teorema Gödel, descartándola por ser pesimista, afirmando que en el campo de las matemáticas está abierto el capturar nuevos conocimientos:

> Existe la idea equivocada de que el teorema de Gödel nos dice que existen <proposiciones matemáticas indemostrables>, y que esto implica que existen regiones del <mundo platónico> de verdades matemáticas que, en principio, nos son inaccesibles. Esto está muy lejos de la conclusión que deberíamos sacar del teorema de Gödel. Lo que Gödel realmente nos dice es que cualesquiera que sean las reglas de demostración que hayamos establecido por adelantado, si ya aceptamos que dichas reglas son dignas de confianza (i.e., que no nos permiten deducir falsedades) y no son demasiado limitadas, entonces disponemos de un nuevo medio de acceso a ciertas verdades matemáticas para cuya deducción aquellas reglas particulares no son lo bastante potentes (Penrose, 2004:517).

Consideramos que la demostración de Gödel con respecto al teorema de la incompletitud nos lleva a reflexionar sobre el significado de un sistema axiomático que utiliza funciones recursivas, que tiene cierta complejidad y que es incompleta, por lo tanto siempre será posible, encontrar una proposición que no se pueda deducir en ella. En términos prácticos se resuelve el problema añadiendo la proposición como un nuevo axioma. Esto nos dice de los sistemas axiomáticos de este tipo como sistemas abiertos.

Hoy en las matemáticas existen fuertes tendencias hacia la unificación no atada al rigor de los axiomas, esta característica la enuncia Berlinski:

---

[71] Roger Penrose, nace el 8 de agosto de 1931 en Colchester, Reino Unido. Físico matemático, profesor emérito de matemáticas en la Universidad de Oxford. miembro de la Royal Society de Londres en 1972, ganó el Science Book Prize en 1990, y compartió el Premio Wolf en Física con Stephen Hawking en 1988. Fue nombrado caballero en 1994. Cursó estudios en la Universidad de Londres y en el St. John's College de Cambridge, desde 1973 ocupó la cátedra de matemáticas en la Universidad de Oxford. Sus publicaciones son diversas: *Techniques of Differential Topology in Relativity* (1973), *Spinors and Space-Time* (junto con Wolfgang Rindler; vol. 1, *Two-Spinor Calculus and Relativistic Fields,* 1984; vol. 2, *Spinor and Twistor Methods in Space-Time Geometry,* 1986), *La nueva mente del emperador* (1989), *Las sombras de la mente* (1994), *Lo grande, lo pequeño y la mente humana* (1997), y su último título*, El camino a la realidad* (2004).

> La Sociedad Matemática Americana enumera cincuenta especialidades matemáticas principales que van desde la topología algebraica a la teoría de conjuntos de Zermelo-Fraenkel. Estas cincuenta especialidades se dividen en más de trescientas sub especialidades. La dificultad de comunicación entre las fronteras es a veces alarmante (Berlinski, 2007:211).

1.6. Modelo.

El concepto de modelo suele tener diferentes interpretaciones, así lo confirmamos en lo expresado por Ferrater Mora quien agrega que "Epistemológicamente, la noción de modelo ha sido, a su vez, empleada en varios otros sentidos" (Ferrater Mora, 2004:2433).

Asimismo David Calvo[72] en su tesis de doctorado en filosofía sostiene que Ferrater Mora señala cuatro usos del concepto de modelo: Como modo de explicación de una realidad, como forma de presentación, como sistema que sirve para comprender otro sistema y como sistema real que la teoría trata de representar (David Calvo, 2006: 35-36), así encontramos lo mencionado por Ferrater Mora:

> Se ha hablado a veces (vagamente) de modelo como de un modo de explicación de la realidad, específicamente de la realidad física. Por ejemplo, se ha hablado de <modelo mecánico> equivalente al mecanicismo… Se ha hablado asimismo de modelo como de alguna forma de representación de alguna realidad o serie de realidades, de algún proceso o serie de procesos, etc. Ejemplo de un modelo puede ser un dibujo… Un modo muy común de entender 'modelo' es tomar como modelo un sistema que sirva para entender otro sistema, como cuando se toma el paso de un fluido por un canal como modelo de tráfico… Otro modo de entender ´modelo es tomar como tal un sistema del cual se trate de presentar una teoría. El modelo es entonces la realidad efectiva o supuesta- que la teoría trata de explicar (Ferrater Mora, 2004: 2433).

Es posible hablar de los diversos usos que se suele dar a la palabra modelo, David Calvo enumera una relación a la que denomina listado provisional en el sentido que se

[72] David Calvo, Doctor en filosofía de la Universidad Complutense de Madrid, sustenta su tesis doctoral sobre *Modelos teóricos de la Física* en el Departamento de Lógica y Filosofía de la Ciencia, becado por 4 años en el Instituto de la Comunidad Autónoma de Madrid, permitiéndole una estancia breve de investigación en el London School of Economics.

trata de una síntesis explicativa. La relación mencionada la consideramos oportuna, dado que elucida la amplitud de temas que están asociados al uso de la palabra modelo:

> … (1) Como modelo de una teoría matemática, la cual esta clausurada respecto de su relación de consecuencia lógica, en el sentido indicado por Tarski (1956). (2) Como interpretaciones de sistema axiomático, dentro del positivismo lógico, donde los modelos son interpretaciones parciales de la teoría a través de las reglas de correspondencia. Es la presentada por Carnap (1947)… (3) Como entidades no lingüísticas, donde la clase de modelos M de una teoría T define un conjunto de mundos compatibles con la verdad semántica de T. Esta concepción es la defendida por Suppe (1990)… (4) Como sistema real, físico, que comparte la misma estructura que la teoría. En la concepción estructuralista de las teorías, los modelos son ejemplos de la teoría. Sneed (1971) y Stegnuller (1981) desarrolla esta versión. (5) Como modelos icónicos, en el sentido de Campbell (1957) o Hesse (1966), que especifican el contenido empírico de la teoría por medio de sistemas conocidos de que pudiera señalarse si existe una analogía positiva, negativa o neutral. (6) Como modelos mediadores, donde el enfoque descansa en las aproximaciones y simplificaciones que permiten desarrollar y manejar los aspectos formales de las teorías, sin que los modelos formen parte de ellas. (7) Como modelos teóricos, que alcanzan mediante hipótesis de bajo nivel la base empírica donde llega la teoría a falta de leyes fundamentales capaces de explicar el fenómeno. (David Calvo, 2006: 39).

Al consultar sobre el significado de la palabra modelo en el diccionario de *Lógica y Filosofía de la Ciencia* de Mosterín y Torretti, encontramos dos formas de uso de la palabra Modelo: “dos acepciones muy diferentes y en cierto modo opuestas” (Mosterín y Torretti, 2002: 387). La primera corresponde a que un modelo es una realización de la teoría, esta concepción se encuentra en la denominada Teoría de Modelos, donde una interpretación M de una teoría T, es un modelo de T, de forma que los enunciados de T se cumplen en M. La segunda corresponde a que un modelo es una representación de una situación, el modelo resulta ser la teoría que explica una realidad.

En las ciencias fácticas (física, química y otras) como en las ciencias formales (lógica y matemáticas), es frecuente el uso de la palabra modelo, pero sin advertirlo, se utiliza con significados diferentes. Es conveniente revisar el sentido del término "modelo" utilizado en las ciencias y distinguir el significado que adopta de un lado en las ciencias formales y del otro lado en las ciencias fácticas.

La definición de modelo en las ciencias fácticas, se encuentra en la relación binaria entre modelo y realidad, a la que se pretende representar: No se trata de una relación donde cada elemento del modelo corresponde a un elemento del sistema real; por el contrario, entre ambos se establece una relación compleja, de sistema a sistema, donde algunas variables del sistema real pueden no aparecer en el modelo y a su vez, algunas variables del modelo pueden no tener su correlato en el sistema real, éste es el caso de modelos que introducen entidades teóricas, no directamente observables, cuyas propiedades no pueden ser determinadas por vía empírica en el sistema real.

En las matemáticas el modelo se relaciona a teorías (previamente desarrollada) en el sentido de contener términos y formas de inferencia pertenecientes a la teoría. En algunos casos se denomina "modelo matemático" de una teoría fáctica[73] a la teoría matemática, a la cual corresponde a dicha teoría fáctica. Se tiende a identificar la teoría fáctica con el modelo matemático asociado. Pero un modelo matemático, en tanto estructura puramente sintáctica, no constituye una teoría fáctica; para convertirse en tal requiere de una interpretación (semántica) en términos de la realidad expresada.

Consideramos adecuado el análisis que desarrolla David Calvo cuando indica la relación entre la realidad y el modelo, a través de ésta el modelo es tratado como la teoría. En un sentido más abstracto el modelo es una teoría:

[73] Factico en el sentido que refiere a una situación en el mundo, relativo a los hechos, basado en lo que ocurre, en posición a lo imaginario.

> La relación "ser modelo de" con la letra M, sería una relación binaria tal que: X M Y = "x es modelo de y". Si el modelo constituye la estructura que ejemplifica la teoría, los dos componentes x e y de esta relación son la realidad y la teoría, respectivamente. (Calvo, David 2006: 40).

La teoría describe a la "realidad", es decir la describe cómo funciona. Esta definición puede trasladarse a las dos orientaciones que suele referirse a modelo (en las ciencias fácticas y en las ciencias formales). Así en las ciencias fácticas el modelo explica una determinada realidad física, mientras que en las ciencias formales el modelo refiere a una realidad que es una teoría matemática, que es una construcción teórica que no tiene ninguna correspondencia a una realidad física y por lo tanto son reales en ella misma.

Otra forma de elucidar el concepto de modelo es revisando la relación binaria entre lo representado y su representación, en el sentido que al representado le corresponde la representación, como una imagen, no es lo representado es su representación. Significa que la representación está en correspondencia binaria a lo representado: Es el caso de lo representado como modelo, ejemplo: Definimos un objeto a producir industrialmente, este es un modelo que debe ser aprobado, para luego producirlo en cantidades. También puede darse el sentido contrario, en la que una representación es el modelo, ejemplo: Un mapa que contiene las rutas de la ciudad, las rutas dibujadas en el mapa son un modelo de las calles de la ciudad.

El modelo entendido como representado, ocurre al considerar al objeto que sirve de observación como modelo para obtener de ésta una imagen, sucede cuando se hace una pintura o para obtener una fotografía.

El modelo como representación ocurre en los casos en el que se formula una teorización que explica una situación real, como sucede en la teoría del modelo atómico (es una representación de un átomo).

El concepto de modelo puede ser entendida en los dos sentidos expresados previamente: el primer caso, el modelo como lo representado se refiere a lo que debe ser obtenido, lo obtenido denota lo que debe ser realizable en el sentido que será lo real (en potencia). En el segundo caso, el modelo como representación es la teoría que refiere a lo real (en acto). Consideramos adecuada la definición de modelo proporcionado por Calvo:

> Un modelo es una reconstrucción racional de la realidad, una representación ordenada de lo que hasta la fecha conocemos de un fenómeno. Cuando el modelo está justificado teóricamente, la teoría proporciona una serie de reglas metodológicas para construirlo, aunque el problema nunca se resuelva mecánicamente, sino a través de suposiciones físicas sobre la naturaleza del fenómeno (David Calvo, 2006:129).

El concepto de modelo supone la diferencia entre lo representado y la representación, estableciendo que la diferencia entre uno y el otro, es de forma tal, que no es posible afirmar la existencia de un isomorfismo en términos absolutos. Por ello consideramos que es pertinente marcar la diferencia entre el modelo y lo real, en el sentido que no se puede considerar plenamente como un isomorfismo, como lo expresa Calvo:

> El isomorfismo entre un modelo teórico de datos de la realidad nunca es exacto, y nunca podrá serlo. Aunque el objetivo de la ciencia sea lograr semejante isomorfismo, es una tarea imposible de realizar, por mucha mejora que se dé en los instrumentos de observación y en los modelos: siempre habrá un modelo más preciso que se adopte mejor a los datos experimentales (David Calvo, 2006: 238).

Es pertinente aclarar que la connotación de la definición de modelo según David Calvo contiene una carga debido a su experiencia con la ciencia física. En la presente investigación utilizamos el concepto de modelo en los dos sentidos mencionados, especialmente en relación a explicar los conceptos que se utilizan como computables, desde una perspectiva de aproximación a una realidad expresada a lo que ejecutan las computadoras, siguiendo en cierto sentido lo sostenido por Calvo:

> Un modelo es una fórmula matemática aplicada a la realidad física; como tal se caracteriza por el número ilimitado de sus predicciones. Si esas predicciones no se comparan con la realidad, con una serie de medidas, no es posible hablar ni de "verdad" ni de "aproximación" (David Calvo 2006: 299).

El concepto de modelo requiere de precisión, específicamente en el campo filosófico y lingüístico, de manera que resulta válido utilizar otra palabra en referencia a "modelo" en las ciencias formales, de manera que contribuya a la claridad expresiva. Estamos de acuerdo con Mosterín en el uso de la palabra "realización" en vez de modelo en el caso de las ciencias formales:

> De todos modos, y para terminar, hay que reconocer que también sería coherente usar la palabra "realización" en vez de "modelo" para lo que se llama modelo en la teoría de modelos y reservar la palabra "modelo" para la descripción teorizada de un sistema real (Mosterín, 2003:253)

Resaltamos que en el campo de la lógica la teoría de modelo, es el estudio de las relaciones entre las estructuras matemáticas y los lenguajes formales, en el prólogo escrito por Mosterín en el libro de María Manzano[74] *Teoría de Modelos* (1989), refiere a que "La teoría de modelos no es una teoría semántica que ponga en relación a los lenguajes naturales con la realidad física y social, sino una teoría matemática que pone en relación unos sistemas matemáticos con otros sistemas matemáticos". Aclara que el libro versa sobre la teoría clásica de modelos. Manzano relaciona un lenguaje con el significado de

[74] María Manzano Arjona, Doctora en Filosofía, profesora de Lógica la Universidad de Salamanca España, son diversas sus investigaciones en el campo de la teoría de modelos y sistemas, habiendo publicado diversos libros y artículos.

sistema, refiriéndose como sigue:

> El esquema abstracto de la Teoría de Modelos es así: Tenemos un lenguaje L y una clase de objetos M que son sistemas, y entre estos dos tipos de realidades tendemos un puente: la noción de verdad. Este planteamiento, aparentemente tan simple, proporciona una flexibilidad y alcance a la Teoría de Modelos. (Manzano, 1989:19).

Manzano para definir el concepto de modelo necesitó precisar una teoría como expresable en un lenguaje, como el conjunto de sentencias cerradas bajo la relación de deducibilidad, es decir, una teoría es un subconjunto de sentencias de un lenguaje y si una sentencia es deducida de la teoría, entonces pertenece a la teoría: “Sea L un lenguaje de primer orden. T es una teoría de L syss $T \in SEN(L)$ y para cada $\varphi \in SEN(L)$ se cumple: si $T \vdash \varphi$ entonces $\varphi \in T$” (Manzano, 1989:155)

Luego introduce el concepto de modelo M como una clase de sistemas, en la que un subconjunto K de M permite obtener una teoría, así nos dice que los elementos del subconjunto de M satisfacen las sentencias del lenguaje L, estas sentencias forman parte de la teoría de K: “Sea $K \in M$. Llamaremos teoría de K al conjunto de todas las sentencias de L verdaderas en todos los sistemas de K. Es decir, $TEO(K) = \{ \varphi \in SEN(L) / A \text{ sat } \varphi$, para cada $A \in K \}$” (Manzano, 1989: 157).

Define que para cada elemento A que pertenece al modelo M, hay una teoría que es completa en A. “Para cada $A \in M$, TEO(A) es una teoría completa”. (Manzano, 1989: 158).

Presenta la definición de modelo de Σ, como los elementos A que pertenecen a M y que satisfacen cada elemento de Σ, así nos dice: “Sea $\Sigma \in SEN(L)$. Llamaremos modelos de Σ a la clase de todos los sistemas que son modelo de Σ. MOD (Σ) = { A ϵ M / A sat φ, para cada $\varphi \in \Sigma$}" (Manzano, 1989: 159)

Es decir, en el lenguaje existen sentencias que constituyen una teoría, en el que dichas sentencias satisfacen determinadas definiciones, de forma que estas son parte del sistema. En este sentido, el modelo son las clases de sistemas que satisfacen las definiciones.

Cabe destacar que Manzano precisa que la teoría de modelos es una rama de la lógica que se ocupa de la relación entre los lenguajes formales y sus representaciones en estructuras, en su artículo *¿Qué es esa cosa llamada lógica?*, del 2005, dice: "El puente entre estos dos tipos de realidades es el concepto de verdad; concretamente la noción de fórmula φ es verdadera bajo la interpretación F".

Manzano afirma que una fórmula φ es consecuencia de un conjunto de fórmulas y que toda interpretación que hace verdadera a las fórmulas del conjunto hace verdadera a φ, en relación a esto, el modelo contiene las expresiones que deduce lo que está constituida en la teoría: "la estructura A es modelo de φ abreviadamente A $\Vdash \varphi$… Podemos definir la clase de todos los modelos de una teoría Mod (Γ) = {A/ A $\Vdash \gamma$, para cada $\gamma \in \Gamma$}" (Manzano, 2005:25)

Define como teoría Th(A) como una descripción de A, así nos dice: "… el conjunto de las sentencias verdaderas en una clase R de estructuras Th(R) = {$\varphi \in$ SENT(L) / A $\Vdash \varphi$ para cada A $\in$ R }" (Manzano, 2005:26).

Estos conceptos definen la lógica, como el conjunto común que está incluida en todas las teorías, lo define como el conjunto VAL, siendo $VAL = \cap \{Th(A) / A \in R\}$

Así mismo, Manzano, en su artículo *Sobre Razonamiento Formal*, publicado por la Universidad de Castilla-La Mancha, en el 2006, define la variación ocurrida en los últimos años en el campo de la lógica, habiendo ampliado su campo de acción, saliendo del formal razonamiento matemático para ser utilizado en diversos campos, así expresa:

> La lógica fue retomando su extensión y amplitud originales estudiándose en ella no solo el razonamiento matemático sino también fenómenos de gestión y transmisión de información, de toma de decisiones y de la acción, y en general en casi todos los contexto gobernados por reglas (Manzano, 2006:71).

Precisa aún más, en cuanto que la lógica no se agota en el cálculo, la extiende más allá lo que puede ser ejecutado por una máquina:

> No se agota en el cálculo que un humano o una máquina pueda efectuar ya que también le interesan las interacciones entre los agentes que participan en la conversación, el proceso de adquisición de conocimiento, la dinámica y el flujo de información (Manzano, 2006:71).

Resaltamos la definición del campo de acción de la lógica, en relación con la teoría de modelos, en su relación a las interacciones entre los diversos agentes que participan de la conversación y/o proceso, específicamente en la dinámica del flujo de información en relación a la denominada tecnología de la información[75].

El término modelo en relación a lo representado y su representación, contribuye a la interpretación sobre aspectos cognitivos en la construcción de programas de

[75] Según la Asociación ITAA (Information Technology Association of America). La tecnología de la información es el estudio, diseño, desarrollo, implementación, soporte o dirección de los sistemas de información computarizados, en particular de software de aplicación y hardware de computadoras.

computadora. Para iniciar la construcción de un programa se requiere definir los diversos componentes y ser expresados en reglas, para este fin, se suele construir un modelo denominado prototipo (no es el programa, pero nos dice sobre el programa) de forma que explique cómo va a funcionar. Sirve como herramienta para la aceptación de lo que se hará.

En sentido contrario también se utiliza el concepto de modelo, cuando se trata de un programa de computadora que está funcionando en una organización, entonces el programa es un modelo de un sistema en funcionamiento, que puede ser utilizado en otra organización, que requiera el programa en sentido similar en donde ya está funcionando.

La lógica formal, cuando es tratada por medio del método del establecimiento de un lenguaje formalizado, se llama lógica simbólica o lógica matemática o logística. Al método mismo lo llamaremos el método logístico.

Alonzo Church [76]

CAPÍTULO II

COMPUTABILIDAD Y RECURSIVIDAD.

El concepto de computabilidad en las matemáticas suele entenderse como la obtención de un valor mediante la ejecución de un cálculo. Este es definido mediante funciones que resultan ser operaciones aritméticas. Dado el desarrollo de las matemáticas en el área de las funciones y de la teoría de los números, permitieron reflexiones en temas que constituyeron la ciencia de la computación. Así en los siglos XIX y XX el razonamiento de tipo matemático se hace filosófico, esto expresado por Roberto Torretti[77]:

> En los siglos XIX y XX la matemática prolifera y florece como quizás ningún otro que hacer del espíritu. Movidos por la misma riqueza y audacia de sus invenciones... Su reflexión es lo que se llama filosofía, y así la entienden; pero la conducen como matemáticos que son, aunando libertad y rigor, fantasía ubérrima y precisión pedante, en el estilo propio de su disciplina (Torretti, 1998: XI).

Ferrater Mora define el concepto de computabilidad mediante funciones del tipo recursivo, incluye términos utilizados por Tarski en relación al concepto de teoría decidible. Afirma que un procedimiento es una secuencia de operaciones que permite la demostración de un teorema, y que la teoría es decidible si sus funciones son recursivas:

---

[76] Alonso Church su libro de 1956, *Introduction to Mathematical Logistic*

[77] Roberto Torretti, autor del libro *El Paraíso de Cantor*, es un documento que contiene diversos temas que están expuestos en la tradición conjuntista en la filosofía de la matemática.

> Se llama decidible a un cálculo C cuando puede forjarse un método o un procedimiento mecánico mediante el cual sea posible decidir – en una serie de operaciones finita – si una fórmula bien formada de C es o no un teorema de C… Si se encuentra tal procedimiento o método, el cálculo o la teoría formalizada reciben el nombre de decidibles; si no, el de indecidible… Para una definición formal suficiente del término 'decidible' aplicando a una teoría formalizada T usaremos la formulación de A. Tarski (*Undecidable Theories*, 1953). Una teoría es llamada decidible si el conjunto de todas sus funciones válidas es recursivo; de lo contrario, es llamada indecidible. (Ferrater Mora, 2004, 786).

Precisa también el concepto de decisión en el mismo sentido que el utilizado por Hilbert en 1900 (al plantear 23 problemas de investigación). Específicamente en el enunciado del tercer problema que dice: Encontrar un procedimiento para la solución de las ecuaciones diofánticas[78]. El programa sugerido por Hilbert trata sobre la búsqueda de la axiomatización de las matemáticas, este sería conocido como *Entscheidungsproblem,* que significa: Problema de Decisión. Hilbert acota al respecto:

> La compatibilidad de los axiomas aritméticos. Cuando estamos inmersos en la investigación de los fundamentos de una ciencia, debemos establecer un sistema de axiomas que contiene una descripción exacta y completa de las relaciones que subsisten entre las ideas elementales de esta ciencia. Los axiomas para configurar son al mismo tiempo, las definiciones de las ideas elementales, y ninguna declaración en el ámbito de la ciencia cuya fundamentación nos están poniendo a prueba se considera correcto a menos que pueda derivarse de esos axiomas por medio de un número finito de pasos lógicos... quiero designar los siguientes como las más importantes de las numerosas preguntas que se le puede pedir en lo que respecta a los axiomas: probar que no son contradictorios, es decir, que un número definido de pasos lógicos basados en los mismos nunca pueden conducir a resultados contradictorios. (Hilbert, 1900:31-32).

El concepto de computabilidad en relación al de recursividad se formuló antes del planteamiento de Hilbert, y fue madurando hasta lograr la definición de una teoría, que se llamaría teoría de la recursión, contribuyendo en los años 1960 al nacimiento de la ciencia de la computación (en Estados Unidos) o informática (en gran parte de Europa).

[78] Ecuación diofántica, es una ecuación algebraica con coeficientes enteros, donde la solución de las variables son números enteros. Ésta debe su nombre al matemático griego Diofanto de Alejandría (200 DC – 280 DC).

En este capítulo revisamos el concepto de computabilidad expresado en la tesis de Church, publicada en el artículo *An Unsolvable Problem of Elementary Number Theory,* de 1936 (Un problema insoluble de la teoría del número elemental), donde define que toda función es efectivamente calculable si es una función recursiva. Nos proponemos esclarecer el concepto de recursividad utilizado por Church en relación al de la computabilidad.

2.1. El número.

Utilizamos los números como expresiones que representan cantidades, en la que suele denominarse numeral a la expresión que indica un número, así por ejemplo: uno, dos, tres, etc. El enunciar los números es parte del lenguaje natural, indican cantidad y son tema del debate filosófico-matemático, en referencia a los fundamentos de las matemáticas, así Ferrater Mora dice sobre las posiciones filosóficas en las matemáticas:

> En efecto, es posible defender una posición formalista, una logicista y una intuicionista… Estas posiciones pueden ser llamadas (latamente) ontológicas. A ellas se agregan las posiciones predominantemente epistemológicas, entre estas destacamos la radicalmente empirista, la apriorista y la conceptualista (Ferrater Mora, 2004: 2597).

Ferrater Mora empieza la definición de número desde el tiempo de los antiguos griegos, indica que los Pitagóricos decían que los números son elementos representativos de la realidad, mientras que para Platón los números forman parte del mundo de las ideas. Para Aristóteles no se puede concebir la unidad como un número, ya que la unidad de medida y el uno son principios, el número no es causa de las cosas. Para Ferrater Mora, el número se define: “como la multitud medida, y como la multitud (o multiplicidad) de las medidas” (Ferrater Mora, 2004: 2595).

En la edad media se enfrentarán dos concepciones ontológicas, por un lado la de los empiristas que consideran que el número se obtiene de la experiencia por abstracción de las cosas particulares, y de otro lado, la concepción de los racionalistas que consideran que el número es apriorístico. En este enfrentamiento de ideas, Ferrater Mora considera que resulta difícil encontrar representantes puros de ambas tendencias, menciona a Dedekind[79] como ejemplo de fundamentación lógica del concepto de número, también a Bertrand Russell que define el número como un modo de agrupar, de esta forma consigue una fundamentación y una aclaración lógica de número.

Cuando revisamos algunos textos matemáticos suele encontrase la definición de tipos de números, como conjuntos. El primer conjunto son los números Naturales, luego siguen los números Enteros, Racionales, Irracionales, Reales, Imaginarios y los Complejos. Cada uno de estos sistemas de números los explicamos en un contexto histórico, nos basamos en el libro de Cesar Trejo, *El Concepto de Número* incorporando al marco teórico la definición de número, porque el concepto de función recursiva trata de números naturales y enteros.

El sistema de números Naturales N es definido extensivamente como el conjunto que tiene la siguiente forma: {0, 1, 2, 3,…}, mostramos el 0 como el primer número (si

[79] Julius Wilhelm Richard Dedekind (6 de octubre de 1831 - 12 de febrero de 1916), matemático alemán, nació en Brunswick. En 1848 entró en el Colegium Carolinum de su ciudad natal, y en 1850 en la Universidad de Gotinga. Su tesis doctoral, supervisada por Gauss, sobre la teoría de las Integrales Eulerianas. Recibió su doctorado en 1852. Estudió la teoría de los números y otras materias. Se dice que fue el primero en impartir clases universitarias sobre la teoría de las ecuaciones de Galois. Fue el primero en comprender el significado de las nociones de grupo, cuerpo, ideal en el campo del álgebra, la teoría de números y la geometría algebraica. Sus cortaduras zanjan definitivamente el problema de la fundamentación del análisis al definir el conjunto de los números reales a partir de los racionales. Caracterizó los números reales como un cuerpo ordenado y completo.

fueran una sucesión). Como mencionamos, una forma de definir a estos números corresponde a los trabajos realizados por Peano mediante cinco axiomas. Otra forma de definir el número Natural es mediante el concepto de cardinalidad de un conjunto, que consiste en agrupar los conjuntos que tienen la misma cantidad de elementos, este método se precisa de la siguiente forma:

> Nos muestra que el concepto de cardinalidad o sea de tener el mismo número de elementos puede darse independientemente del concepto de número de elementos. En esta observación se basa precisamente el método de Cantor-Frege-Russel: en lugar de utilizar el número para verificar la cardinalidad, se utiliza la cardinalidad de conjuntos para definir el número (Trejo, 1968:8).

Los números desde el tiempo de los antiguos griegos fueron considerados desde una perspectiva geométrica, como ocurre en la multiplicación y división de una magnitud por un número natural[80]. El producto de dos magnitudes se trata de áreas de rectángulos comprendido por los dos segmentos, esto se encuentra en el *libro II de Euclides*. Al mostrar la igualdad $\frac{a}{b} = \frac{c}{d}$ para a, b, c y d que resultan ser longitudes de lados de dos rectángulos, esto es equivalente a: a × d = c × b, como una igualdad de áreas.

El sistema de números Enteros (Z) es la unión de los números Naturales {0, 1, 2, 3,…} con sus números negativos {-1, -2, -3,…} resultando {…, -3, -2, -1, 0, 1, 2, 3,…}. Resaltamos que en la sucesión de estos números no es posible especificar el primer número, es decir, su extensión va del menos infinito al infinito.

Mosterín y Torretti en su libro *Diccionario de Lógica y Filosofía de la Ciencia* indican que el uso de los números negativos está documentado en textos babilónicos y

[80] La división de un segmento en *n* partes iguales, recibió un tratamiento particular apoyada en el Teorema de Thales, las áreas de las figuras geométricas corresponden a la multiplicación de números naturales.

chinos, pero recién se aceptan como tales en la edad media, al respecto:

> … está documentado en textos babilónicos y chinos anteriores a nuestra era y el indio Brahmagupta (s. VI a.C) dio reglas explícitas para operar con ellos. Por otra parte solo hacia fines de la edad media europea se llegó a admitirlos como soluciones de problemas aritméticos (Chuquet, 1484), y todavía Descartes los llamaba "número falsos" (Mosterín y Torretti, 2002:404).

Para definir los números Racionales (Q), se precisa la correspondencia de dos conjuntos A y B, expresando el producto cartesiano, al que simbolizamos $A \times B$, éste contiene todas las posibles correspondencias entre los elementos de ambos conjuntos. Una relación R entre los dos conjuntos, es un subconjunto del producto cartesiano, así lo expresamos $R \subset A \times B$, de forma tal que se establece cierta correspondencia entre los elementos del primer conjunto con el segundo. La relación entre los elementos x e y es denotada por $x R y$ (un elemento (x, y) pertenece a la relación R) y nos dice que $(x, y) \in R$

Una relación de equivalencia es la relación que define una partición del conjunto, la divide en subconjuntos disjuntos, de tal forma que la unión de todos los subconjuntos obtenidos da el conjunto, ésta relación de equivalencia define el teorema de la partición, donde R es de equivalencia si cumple con las siguientes condiciones:

- Reflexiva: Si $\forall x \in A \Rightarrow ((x, x) \in R)$, todo elemento se relaciona consigo mismo.
- Simétrica: Si $\forall x, y \in A \Rightarrow ((x, y) \in R)\ (y, x) \in R)$, si dos elementos se relacionan en un sentido, también se relacionan en sentido contrario.
- Transitiva: Si $\forall x, y, z \in A \Rightarrow (((x, y) \in R \wedge (y, z) \in R))\ (x, z) \in R)$, si un elemento se relaciona con un segundo elemento y este con un tercer elemento, entonces el primer elemento se relaciona con el tercer elemento.

La relación de equivalencia R sobre un conjunto A, trata de que para cada elemento de A (a ϵ A), una clase de equivalencia a, denotada como [a], son conjuntos de forma tal que la unión de todas las clases constituyen el conjunto A, así cada elemento de la clase puede ser identificado con un único elemento. En el caso de los números Racionales Q que tienen la forma de quebrados $\frac{n}{m}$, tal que n, m ϵ Z (son números Enteros), la relación divide a un número entre otro, es una relación de equivalencia en Z: "La relación ≈ entre pares ordenados a/b, c/d,… de números Enteros… es una relación de equivalencia, es decir… tiene las propiedades: reflexiva: a/b ≈ a/b, simétrica: a/b ≈ c/d → c/d ≈ a/b, y transitiva a/b ≈ c/d y c/d ≈ e/f → a/b ≈ e/f" (Trejo, 1968:56).

Como ejemplo de una clase de equivalencia: la clase [$\frac{1}{2}$] sería el conjunto de todas las combinaciones del par de números enteros que expresan $\frac{1}{2}$, así el conjunto estaría formado por $\{\ldots \frac{-3}{6}, \frac{-2}{4}, \frac{-1}{2}, \frac{1}{2}, \frac{2}{4}, \frac{3}{6}, \ldots\}$

Los números Racionales Q se definen mediante una clase y tienen la propiedad que dado dos números racionales diferentes, existe un tercer número racional entre ambos. Esta propiedad dice que los números Racionales son densos con respecto a la relación de orden. Si imaginamos los números Racionales en una recta numérica, éstos no llenan todos los puntos, aunque podríamos suponer que se obtiene la densidad en toda la recta.

Los números inconmensurables[81] suelen ser definidos por la combinación de

[81] En matemática, la conmensurabilidad es una característica de dos números. Dos números reales, *a* y *b*, que no sean cero, son conmensurables sólo cuando la razón *a/b* es un número racional. Si la razón de *a/b* es irracional, entonces se dice que es inconmensurable, sinónimo de que no puede ser medido.

números racionales, esto fue analizado por los antiguos griegos y se le atribuye su estudio a Eudoxo.[82] Euclides en el libro VII de los *Elementos* expone la teoría de Eudoxo, es así que en la definición 20 observamos: "(Cuatro) números son proporcionales cuando el primero es el mismo múltiplo, la misma parte o las mismas partes del segundo que el tercero del cuarto". Al comparar con una notación moderna, es equivalente a decir que dado cuatro números naturales *a, b, c y d* son proporcionales si, cada vez que se tengan dos números m y n tales que $m \times a = n \times b$, resulta también que $m \times c = n \times d$.

Los números Irracionales (I), no pueden ser expresados en relación a los números Enteros, suelen ser expresados en relación a números Racionales aprovechando la propiedad de la densidad, resultando que todo número Irracional puede ser una aproximación de operaciones de números racionales, así por ejemplo el teorema de Lagrange[83] que dice que los números irracionales cuadráticos son representados en términos de fracciones continuas. En el libro de Penrose *El Camino a la Realidad,* en la página 112, presenta ejemplos de irracionales cuadráticos $\sqrt{2}$, $\sqrt{3}$, $\sqrt{5}$ .... en la página 110, presenta la forma de una fracción continua para un número, siendo éstos:

[82] Eudoxo (408 -335 a. de J.C.) Nació en Cnido -en la península de Resadiye, Turquía- viajero y maestro en las costas orientales del Mediterráneo y Egipto hasta Grecia. Se le atribuye la fundación de la astronomía matemática, contribuyó a la teoría de la proporción. En geometría, su contribución a la teoría de la proporción (Aristóteles, *Segundos Analíticos*; Euclides, Elementos, V), la prueba de dos teoremas avistados por Demócrito: el volumen de una pirámide es un tercio del volumen de un prisma; el volumen de un cono es un tercio del volumen de un cilindro (Arquímedes, prefacios de *Sobre la esfera* y *el cilindro y del Método*). Las magnitudes que considera Eudoxo son todas aquéllas que, siendo homogéneas, pueden guardar razón entre sí y satisfacer la condición: si A < B, hay un número (entero positivo) n tal que n+ A > B.

[83] Joseph Louis Lagrange, bautizado como Giuseppe Lodovico Lagrangia, también llamado Giuseppe Luigi Lagrangia o Lagrange (25 de enero de 1736 en Turín - 10 de abril de 1813 en París) fue un matemático, físico y astrónomo italiano que después vivió en Prusia y Francia. Trabajó para Federico II de Prusia, en Berlín, durante veinte años. Lagrange demostró el teorema del valor medio, desarrolló la mecánica Lagrangiana y tuvo una importante contribución en astronomía.

Para la raíz cuadrada del número dos: $\sqrt{2} = 1 + \cfrac{1}{2+\cfrac{1}{2+\cfrac{1}{2+\cfrac{1}{2+\cfrac{1}{2+\dots}}}}}$

Para un cuarto del número Pi: $\frac{1}{4}\prod = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$

En el año 825 D.C. Muhamad Al-Khowarizmi[84], en su libro *Al-jabr w'al muqabala* utilizó magnitudes irracionales a las que ha llamado *gidr asamm* (raíz muda o ciega)[85]. Progresivamente los números Irracionales fueron incluyéndose en el álgebra y en la aritmética.

En el año 1077 Omar Al-Khayyam[86] retomó la teoría de proporción expuesta por Euclides en el libro V de los *Elementos*. Aconsejó la utilización del método de las fracciones continuas para comparar razones entre segmentos inconmensurables e intentó establecer una equivalencia lógica entre ambas teorías. Al-Khayyâm se preguntó también por la naturaleza de las razones y su relación con los números, sin embargo esto permaneció sin difusión por casi cinco siglos, hasta el año 1594 cuando fue publicada en

---

[84] Abu Abdallah Muḥammad ibn Muss Al-Khowaizmi (أبو عبد الله محمد بن موسى الخوارزمي ابو جعفر), conocido generalmente como al-Juarismi, fue un matemático, astrónomo y geógrafo persa musulmán chií, que vivió aproximadamente entre 780 y 850. Poco se conoce de su biografía, algunos sostienen que nació en Bagdad, otros sostienen que nació en la ciudad Corasmia de Jiva, en el actual Uzbekistán. Es considerado como el padre del álgebra y como el introductor de nuestro sistema de numeración, su nombre dio origen a la palabra algoritmo a su nombre y las palabras álgebra y guarismo al de su obra principal:"Kitab al-jabr wa'l muqabala"

[85] En el siglo XII ese término fue traducido al latín como *surdus* y hasta el siglo XVIII los números irracionales eran llamados también números sordos.

[86] Omar Al-Khayyam. se cree que nació en Nishapur, al norte de Persia, hacia el 1050. Viajó mucho y vivió en Samarkanda, Ispahan, Merv. Llamado por el sultán turco de Ispahan, dirigió el observatorio. En su libro Álgebra, amplía la obra del también matemático Al-Khowarizmi generalizando la resolución de ecuaciones cúbicas con alguna raíz positiva. Hay una clara la relación entre el álgebra y la geometría.

Europa en idioma árabe para ser traducida al latín en el año 1657.

Destacamos los trabajos de definición del número Real realizado por Cantor y Dedekind, en que el significado de número tiene relación con el concepto de computar, como el de obtener un valor dado un argumento. Definimos de manera intuitiva a los números Reales como aquellos que son la unión de los números Racionales con los números Irracionales.

El número Real representado mediante la notación decimal, a la izquierda del punto contiene una serie de dígitos que representa la parte entera y a la derecha del punto otra serie de dígitos para la parte decimal.

Una forma de presentar la definición de los números Reales, es mediante el concepto de Encaje de Intervalos, formulado por Cantor. Trejo describe: Un número real A es una clase de equivalencia de encajes de intervalos racionales como $[a_1,b_1]$, $[a_2,b_2]$, …, $[a_{n-1},b_{n-1}]$, $[a_n,b_n]$ en la que $a_i < b_j$ para cualquier par de índices i, j.

> Llamaremos encaje de intervalos racionales a toda sucesión de intervalos cerrados de extremos racionales, cada uno contenido en el anterior, y tal que dado un número positivo $\varepsilon > 0$ arbitrario (tan pequeño como se quiera), resulta $b_n - a_n < \varepsilon$ con tal de tomar n suficientemente grande (mayor que un número n_0 que depende del ε dado) (Trejo, 1968: 70).

Como ejemplo de la definición de un número real mediante el encaje de intervalos tenemos el caso de $\sqrt{2}$ = 1.414…, en la que los intervalos serían: [1, 2], [1.4, 1.5], [1.41, 1.42], [1.414, 1.415]… Los intervalos se construyen de forma que el siguiente intervalo está incluido en el intervalo anterior.

Cantor especificó que dado un encaje de intervalos, encontramos una sucesión de números, en los números de la izquierda de cada intervalo, o del mismo modo en los la parte derecha. El valor al que tiende la sucesión es el número Real.

Siguiendo con el ejemplo del caso $\sqrt{2}$ la sucesión de números (los números a la izquierda de cada intervalo) según lo mostrado el encaje de intervalos sería: 1, 1.4, 1,41, 1,414… resultando como valor de $\sqrt{2}$ el número 1.414…

Dedekind define a los números Reales[87] utilizando los números Racionales basado en el concepto de la cortadura, y dice que dado dos conjuntos A y B tales que forman una cortadura, si cumple con: Para todo x $\in$ Q se tiene que x $\in$ A ó x $\in$ B (x solo pertenece a uno de los conjuntos), siendo A $\bigcap$ B = $\varnothing$. Para todo x $\in$ A, y $\in$ B, se tiene x $\leq$ y, existiendo un único número γ, para todo x $\in$ A, é y $\in$ B, resultando que x $\leq$ γ $\leq$ y.

Ejemplo: Definimos la cortadura x $\leq$ 3, en los números Reales, formándose dos conjuntos, el primero con los números menores e igual a tres y el segundo conjunto con los números mayores de 3. El número irracional sería el que se encuentra en el límite izquierdo del segundo intervalo.

El número complejo C, es el número que tiene la forma a + b i, donde a y b son números reales e i es el número definido como $\sqrt{-1}$ (llamado imaginario). Penrose

[87] Información adicional puede obtenerse del trabajo publicado por Karen, García, Yanelys Zaldívar y Cecilia Gálvez de la Universidad de la Habana, Cuba http://www.monografias.com/trabajos-pdf/números-reales/números-reales.pdf

menciona que Bombelli[88] introdujo el imaginario en 1572 en su obra *L` Algebra*, estos números son definidos como puntos cartesianos entre las rectas de los números reales y los números imaginarios, así tendríamos en la figura siguiente el número complejo a + b i.

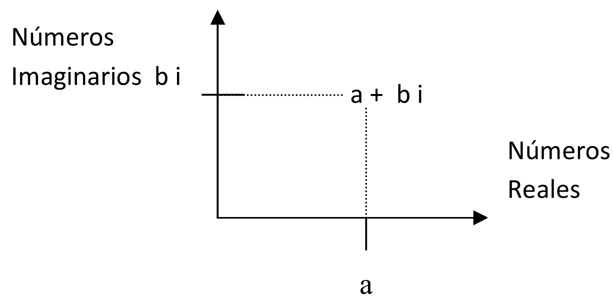

Figura 1. Representación de un número complejo a+bi

Resaltamos la importancia de los números en los procesos de cálculo aritmético, especialmente cuando se trata de la relación de orden de los números, que nos dice que dado dos números diferentes, uno es menor al otro. Afirmamos que los números no pueden ser definidos únicamente como conjuntos, debido a que éstos contienen por definición la relación de orden.

Desde la perspectiva del cálculo algebraico, específicamente en la solución de las

[88] Raphael Bombelli nació en enero de 1526 en Bolonia (Italia). No recibió educación universitaria, adquirió su formación con el ingeniero y arquitecto Pier Francesco Clementi. Adquirió reputación en ingeniería hidráulica. Antonio María Pazzi, profesor de matemáticas en la universidad de Roma, le enseñó a Bombelli un manuscrito de la Aritmética de Diofanto, los dos decidieron hacer conjuntamente una traducción. La obra de Bombelli titulada "Álgebra" está dividida en cinco libros. Los tres primeros fueron publicados en 1572, y anunciaba que los libros IV y V, dedicados a la geometría, aparecerían seguidamente. Desgraciadamente Bombelli nunca llegó a publicar estos volúmenes. Murió en 1573, probablemente en Roma. En 1923, un manuscrito de Bombelli fue descubierto en una biblioteca de Bolonia. Además de una versión manuscrita de los tres libros publicados, había un manuscrito inconcluso de los otros dos libros. Fue el primero que escribió las reglas para la suma, resta y multiplicación de los números complejos. Además demostró que usando el cálculo de los números complejos podían resolverse ecuaciones. Se reconoce a Bombelli como el inventor de los números complejos y su Algebra tuvo una influencia en Leibniz.

ecuaciones algebraicas, cito el teorema fundamental del álgebra demostrado por Gauss[89], en 1799, en el que dice que todo polinomio[90] de una variable de grado n puede descomponerse en el producto de n factores de primer grado, cada una representa una solución de la ecuación si el polinomio se compara con el cero.

Si tenemos el polinomio P(x) de grado n, éste puede ser representado de la forma: $P(x) = (x-a_1)(x-a_2)\dots(x-a_n)$ donde cada valor a_i es un número real o un número complejo. Ejemplo, para el caso del polinomio $P(x) = x^2+1$, puede ser expresado como $P(x) = (x - \sqrt{-1})(x + \sqrt{-1}) = (x - i)(x + i)$.

Añadimos a la argumentación el que la función recursiva es definida en base a los números naturales, teniendo la dificultad de expresar a los números inconmensurables, dado que no es posible obtener cálculos exactos, aunque hemos logrado obtener resultados mediante aproximaciones.

2.2. Los lenguajes formales.

El lenguaje matemático tiene entre sus características el utilizar fórmulas y símbolos que se ajustan a reglas de sintaxis en correspondencia a significados, estas reglas son necesarias para la comunicación entre los matemáticos, gracias a esto, la actividad lingüística adquiere cierta formalidad, de tal manera que elimina en cierto grado la ambigüedad, que es natural en el lenguaje cotidiano.

---

[89] Johann Carl Friedrich Gauss (30 de abril de 1977, Brunswick- 23 de febrero de 1855, Göttingen) matemático, astrónomo y físico alemán. Contribuyó en diversos campos: teoría de los números, análisis matemático, magnetismo, óptica entre otros.

[90] En este caso refiero a polinomio a la expresión algebraica de una variable de la forma: $P(x) = x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_n$ en la que el exponente mayor es el grado.

El lenguaje matemático es parte del lenguaje natural, y su desarrollo está en relación directa al avance de las matemáticas, específicamente en la forma como se sostiene la demostrabilidad de los teoremas, así nos dice Yuri Manin[91]: "… el lenguaje matemático está subordinado a los rígidos principios de la ordenación correcta. Estas reglas han de garantizar la veracidad de las conclusiones" (Manin, 1979: 9).

Kleene en su libro *Introduction to Metamathematics,* introduce un sistema formal para el estudio de la teoría clásica de números, define la Tesis de Church en relación a las funciones recursivas parciales. El sistema formal es construido sin procurar ninguna interpretación, así expresa:

> Nuestra tarea presenta dos aspectos distintos. En primer lugar, ha de ser descrito e investigado el sistema formal mismo con métodos finitistas y sin hacer uso de interpretación alguna del sistema. Esto es la metamatemática. En segundo lugar, ha de ser precisada una interpretación del sistema bajo la cual constituya este una formalización de la teoría de números. (Kleene, 1952: 71).

En los lenguajes formales se busca eliminar la ambigüedad de los términos, diferenciándose de los lenguajes naturales, en la que el significado de los términos muchas veces es ambiguo. Las reglas de formación de las expresiones están constituidas por las reglas sintácticas, dado que contienen estructura de las mismas.

El lenguaje formal se confronta con la realidad según reglas semánticas, las cuales determinan el uso del mismo, en correspondencia a los significados. La descripción de las reglas sintácticas y semánticas del lenguaje formal son parte del metalenguaje

[91] Yuri Ivanovich Manin, nació en 1937, en Simferopol Rusia, matemático, conocido por su trabajo en geometría algebraica y la geometría diofantica, y en muchas obras que van desde la lógica matemática a la física teórica. Obtuvo su doctorado en 1960 en el Instituto de Matemáticas Steklov como un estudiante de Igor Shafarevich. Actualmente es Profesor y Director del Instituto Max-Planck-Institut Mathematik en Bonn y profesor de la Universidad Northwestern.

(entendiendo al lenguaje formal como el lenguaje del lenguaje). Yuri Manin considera que determinada clase de razonamientos matemáticos o procesos de cálculo son la realidad para los lenguajes de las matemáticas, precisando la división entre los lenguajes formales y los algorítmicos, en relación al uso del lenguaje natural, el primero en su forma indicativa y el segundo en su forma imperativa: "En consonancia con tal o cual destinación, estos lenguajes se dividen en formales y algorítmicos. (Compárense, en los lenguajes naturales, la contraposición del modo indicativo e imperativo, al nivel de textos, la comunicación y el orden)" (Manin, 1979:14).

Es interesante comparar los lenguajes formales y los algorítmicos, en el primero le corresponden las expresiones de forma indicativa, que son expresiones aseverativas (afirmativa o negativa) mientras en los segundos, son órdenes y/o instrucciones a ser ejecutadas. Los lenguajes formales son de diversos tipos, dependen de lo que están formalizando, a teorías concretas, por ejemplo, el lenguaje utilizado en la teoría de conjuntos de Zermelo[92] – Fraenkel[93], que utiliza símbolos para representar a los conjuntos y sus elementos.

Yuri Manin considera que las fórmulas son parte del lenguaje formal y que al escribirlas en forma abreviada se busca cumplir objetivos psicológicos asociados a la comprensión de lo expresado: "Al elegir una escritura abreviada, se persiguen, en lo general, objetivos psicológicos: la rapidez de lectura… la ligereza de surgimiento de

[92] Ernst Friedrich Ferdinand Zarmelo (27 de julio 1871- 21 de mayo de 1953). Matemático y filósofo alemán. Cursó estudios en las universidades de Berlín, Halle y Freiburg. Se doctoró en 1894 y completó sus estudios en Göttingen en 1899. En 1902 publica sobre la adición de cardinales transfinitos. En 1908 prueba el teorema del buen orden mediante el axioma de elección. en 1910 deja Göttingen y se va a la universidad de Zúrich. En 1908 publica sus resultados sobre la axiomatización de la teoría de conjuntos.

[93] Adolf Abraham Fraenkel (17 de febrero de 1891 – 15 de octubre de 1965). Matemático alemán – israelí. Estudió en las universidades de Múnich, Berlín y Breslan. Se doctoró en 1914. En 1929 se traslada a Jerusalén. Mejora los axiomas sobre la teoría de conjuntos de Zermelo en 1924 y 1925. Demuestra la independencia del axioma de elección.

asociaciones útiles y la dificultad de surgimiento de asociaciones nocivas, la coincidencia de las costumbres del autor y del lector” (Manin, 1979: 22).

Es característica de los lenguajes formales el que sus reglas sintácticas no sean equívocas, es decir, no permiten la ambigüedad de sus significados, siendo utilizado por las matemáticas. Desde cierto punto de vista se podría decir que las matemáticas son un lenguaje, en el caso de un físico que utiliza las matemáticas para expresar leyes físicas. Sin embargo, si las matemáticas fueran un lenguaje, éste estaría definido por una gramática, y no es el caso, porque debería incorporar las reglas de un lenguaje natural. Las fórmulas matemáticas en muchos casos son insuficientes para expresar conceptos matemáticos, se necesita del lenguaje natural para abarcar las diversas áreas de las matemáticas.

Supongamos que juntamos todas las palabras técnicas que utilizan las personas que trabajan en el campo de la programación de computadoras. Estos utilizan términos tales como: programa, archivo, SQL, dato, pantalla, etc. Consideramos que estas palabras no forman otro lenguaje, que sea diferente al español.

Las matemáticas, son una ciencia que se expresa en el lenguaje de la comunidad de las personas que la utilizan, sus temas son diversos y sus fórmulas están en relación a los temas. No es una unidad lingüística diferente a la que pertenecen, de forma tal que se puede decir que utilizan el lenguaje hablado en el lugar que habitan. La necesidad de un lenguaje formal, radica en la claridad de exposición de sus teoremas.

Kleene define para un sistema formal la necesidad de las reglas de transformación que permitirán hacer una demostración, estas se constituyen explícitamente: “El propósito de la formalización de una teoría es obtener una definición explícita de lo que constituye una demostración dentro de la teoría. Una vez conseguida esta definición, no hay necesidad de recurrir siempre directamente a ella" (Kleene, 1952: 86).

Entendemos que la lógica es una estructura como un sistema axiomático que no tiene significado semántico, en cuanto verdad a una realidad que depende de sus postulados y proporciona la ventaja de que podemos incorporar un significado, de forma tal, que se utiliza la estructura en la lógica para el uso en el significado asignado, Yuri Manin dice: "El objeto de la lógica no es el mundo exterior, sino sus sistemas de comprensión. La lógica de uno de tales sistemas – de las matemáticas – debido a su normalización representa una especie de plantilla rígida, la cual puede ser aplicada sobre cualquier otro sistema" (Manin, 1979: 63).

Con respecto al rol de la lógica en las matemáticas, es tema de discusión filosófica, resultando de estas tres escuelas u orientaciones en las matemáticas, las cuales se originan por la concepción ontológica en la que perciben las definiciones matemáticas o los objetos estudiados en esta materia de la ciencia. José Babini[94] , en el noveno capítulo de su libro *Historia de las Ideas Modernas en Matemática,* sustenta la polémica como consecuencia de las paradojas, como la de Russell: "conjunto de todos los conjuntos que no se contienen a sí mismos como elementos". Así también Babine sostiene "Las cuestiones que suscitaron estas paradojas desataron la polémica, que culminó hacia 1930, en la que se perfilaron tres tendencias: logicista, formalista e intuicionista" (Babini, 1967: 51).

El debate se enmarca en la intención de hacer de las matemáticas una ciencia que

[94] José Babini (10 de mayo de 1897, 18 de mayo de 1984). Historiador argentino, ingeniero y matemático. Historiador de la ciencia. Se le considera el padre de esta disciplina en su país.

tenga una sola raíz, en este sentido se presenta Hilbert como el representante del formalismo, siendo esta corriente la más afín a los matemáticos de profesión, pretendiendo eliminar la intuición, predominando el signo sin ningún contenido empírico.

> En definitiva, el lema del formalismo es: "Al principio fue el signo", lo que equivale a concebir la matemática como un variado juego de signos y de símbolos, de carácter formal y sin contenido empírico alguno. Estas "formas vacías" obedecen a una serie de reglas de estructura y de deducción que, en último análisis, descansan en un sistema de axiomas (Babini, 1967:54)

En el caso del logicismo, se pretende que los conceptos matemáticos sean definidos únicamente con criterios lógicos, de forma tal que las matemáticas sean parte de la lógica, así nos dice:

> … debió su nombre al hecho de pretender que los conceptos básicos de la matemática podían definirse mediante recursos puramente lógicos, con lo cual la matemática perdía su autonomía para convertirse en una parte de la lógica o, en el mejor de los casos, constituía con la lógica una única y misma disciplina (Babini, 1967:52)

El logicismo surgido desde Dedekind y Frege luego reactivado por Russell en el célebre libro *Principia Mathematica*[95] escrito conjuntamente con Alfred Whitehead, en el que proponen la lógica como fundamento matemático, en la que los términos matemáticos pueden reducirse a conceptos y leyes puramente lógicos. La forma de eliminar los problemas presentados en las paradojas fue el admitir el llamado "principio del círculo

[95] *Principia Mathematica* es un conjunto de tres libros de la matemática escritos por Bertrand Russell y Alfred North Whitehead y publicados entre 1910 y 1913. Este constituye un intento de derivar la mayor parte de los conocimientos matemáticos de la época a partir de un conjunto de principios o axiomas. La principal motivación para esta obra provenía del trabajo anterior de Gottlob Frege en lógica que contenía algunas contradicciones descubiertas por Russell. Éstas eran evitadas en los Principia construyendo un sistema elaborado de "tipos".

vicioso"[96] y resolverlo mediante la teoría de los tipos.

La teoría de los tipos es una jerarquía, de forma similar a como se enuncia en la teoría de conjuntos, trata sobre la diferenciación entre los elementos y los conjuntos, expresado cuando referimos a un conjunto y a su conjunto potencia. La primera corresponde a un elemento del conjunto: $a \in A$ (a pertenece a A), y la segunda a un sub conjunto de un conjunto: $A \subset B$ (el conjunto A incluido en el conjunto B). Si decimos que un conjunto pertenece a otro conjunto ($A \in B$), estamos en el caso de que el conjunto B tiene la forma de un conjunto Potencia (es el conjunto de los sub conjuntos de un conjunto): $B = P(C)$, diferenciándose en la forma de los elementos del conjunto A.

El intuicionismo debe su nombre a la naturaleza intuitiva que se asigna al conocimiento matemático. En los textos que tratan sobre estos temas se considera como padre de esta escuela a Brouwer[97] quien propone una forma distinta de hacer matemáticas. Esta teoría no acepta la regla lógica del tercio excluido: en la que una proposición es verdadera o falsa, porque en las matemáticas es posible la aceptación de proposiciones en la medida que se sostienen en un sentido constructivista[98]. En cierto sentido se puede decir que la lógica multivaluada sería una forma lógica aceptada. También el intuicionismo

[96] Circulo vicioso, es un argumento circular que consiste en intentar probar una cosa mediante otra, y esta segunda mediante la primera. En un sentido general, se usa para referirse al paralogismo en que se cae cuando se introduce en la definición la palabra que se pretende definir o bien cuando se da como prueba de una proposición otra proposición que, a su vez, se prueba por la primera. No todo paralogismo es falaz, hay fenómenos que solo se pueden ser explicados de forma circular, ya que se retroalimentan.

[97] Luitzen Egbertus Jan Brouwer matemático holandés (1881-1966), graduado en la Universidad de Ámsterdam. Sus trabajos versan sobre Lógica, Topología, Teoría de la Medida y Análisis Complejo. Promovió la escuela matemática Intuicionista. funda el Intuicionismo Matemático.

[98] En constructivismo es una corriente en las matemáticas que afirma que es necesario encontrar o (construir) un objeto matemático para poder probar su existencia.

considera no válido el principio de la refutación de la falsedad[99]. Suele considerarse al intuicionismo una forma de constructivismo, en el sentido que las definiciones o demostraciones matemáticas requieren ser construidas en una definición y a partir de esta la demostración para aceptar el enunciado matemático. Babini afirma que:

> El intuicionismo debe su nombre al carácter intuitivo inmediato que asigna al conocimiento matemático… En efecto para esta escuela la existencia matemática ya no equivale a no contradicción como en el formalismo, sino que significa constructividad (Babini, 1967: 56)

El intuicionismo es considerado por cierta literatura filosófica como una nueva forma del pensamiento kantiano, dado que prioriza su carácter a priori. Gladys Palau[100] dice: “Es sabido que el intuicionismo, en tanto filosofía de la matemática, se inspira en las tesis kantianas acerca del carácter a priori de los principios de la aritmética y de su conocimiento por medio de la intuición” (Palau, 2002:79).

David Berlinski en su libro *Ascenso Infinito Breve Historia de las Matemáticas*, 2007, en el noveno capítulo sostiene que el proyecto formalista propuesto por Hilbert en la famosa conferencia en Paris, en la que expone sobre la consistencia basada en los axiomas, ponderando lo expresado por Cantor, indicando que la consistencia es el único estándar probatorio para las matemáticas, así también, cuando Russel y Whitehead presentan el libro *Principia Mathematica* en 1910, Hilbert no lo consideró contrario a los propósitos de la formalización, con la observación de que no había resuelto el problema planteado sobre la consistencia.

---

[99] En la lógica clásica se tiene la equivalencia de la fórmula p $\rightarrow$ q con la formula -q $\rightarrow$ -p, en el intuicionismo esta equivalencia es rechazada.

[100] Gladys Palau, doctora en Filosofía (UBA). Profesora de la Universidad de Buenos Aires. Profesora de la Universidad Nacional de La Plata. Profesora de la Maestría en Epistemología e Historia de la Ciencia de la Universidad Nacional de Tres de Febrero. Directora del Seminario Permanente de Lógica “Carlos E. Alchourrón” de la Sociedad Científica Argentina. Miembro de la Sociedad Argentina de Análisis Filosófico, de la Asociación de Filosofía de la República Argentina y de Sociedad de Lógica, Metodología y Filosofía de la Ciencia en España.

> Como Hilbert llego a comprender, Russell y Whitehead no habían demostrado la consistencia de su sistema porque no habían reconocido la importancia de la pregunta de 1910 y no estaban en condiciones de responder después (Berlinski, 2007:184).

En medio del debate entre las diferentes tendencias filosóficas de las matemáticas se encuentran los trabajos de Kurt Gödel quien en 1929[101], diserta su tesis doctoral, sobre la demostración de la completitud de la lógica de primer orden, siendo bien recibida por los formalistas y los logicistas. Sin embargo en 1931 publicará sus célebres teoremas de la incompletitud *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme* (Sobre proposiciones formalmente indecidibles de *Principia Mathematica* y sistemas relacionados). Demostró que el sistema de *Principia* era incompleto.

> A partir de sus símbolos siempre era posible construir una proposición tal que si los Principia eran consistentes, no se podía encontrar una demostración ni para la proposición ni para su negación. Lo que es más esta proposición es cierta. (Berlinski, 2007: 191).

Así Gödel da un golpe mortal a la escuela formalista, con sus dos teoremas. El primero, que se obtiene de la proposición VI de su publicación en 1931, en el que demuestra que cualquier formalización consistente de las matemáticas que sea lo bastante fuerte para definir el concepto de números naturales, pueden construir una afirmación en la que no se puede demostrar que sea verdadera o falsa dentro del sistema; el segundo teorema prueba que ningún sistema consistente puede demostrarse en sí mismo.

En opinión de Berlinski, el teorema de Gödel estuvo sin difusión hasta 1961,

[101] En 1929 Gödel termina su tesis doctoral y escribe su artículo Ŭber formal unesntscheidbare Sätze der principia Mathematica and verwandter Systeme, publicado en 1931, es la proposición VI de este artículo que se conoce como el primer Teorema de Gödel.

(según su experiencia cuando estudió en Princeton). El teorema solía ser comprendido a partir de notas cuidadosamente elaboradas por Church y el popular texto de Ernest Nagel[102] y James Newman[103]. Hoy tenemos un libro didáctico de Douglas Hofstadter[104], *Gödel, Escher, Bach, Un Eterno y Frágil Bucle,* que ayudan a una mejor comprensión del teorema de Gödel, con la aclaración necesaria, de que en el libro, el concepto de bucle se entiende en el mismo sentido al de recursivo.

El debate entre las tres escuelas contribuyó al avance de la lógica. El formalismo considera que el intuicionismo destruye las bases de las matemáticas y pondera en cierto sentido el límite que tiene la lógica clásica, en contradicción con el logicismo que pretende incluir a las matemáticas dentro de la lógica o en el mejor de los casos asignarle el mismo estatus. El logicismo tiene como rival al intuicionismo, este último considera que las matemáticas no son bivalentes (verdad o falsedad) sino polivalente (múltiples valores que van desde el falso hacia la verdad), asimismo, el intuicionismo critica la utilización de la lógica clásica, por considerarla una reducción de la creación de la mente humana, de otro lado, se ha formalizado un tipo de lógica conocida como lógica intuicionista[105], es decir, en cierto sentido se ha transformado en una forma de lógica.

Cabe destacar el beneficio obtenido del debate sobre de la crisis en los

---

[102] Ernest Nagel (1901 – 1985). Filósofo estadounidense de origen checoslovaco. Profesor en Columbia (New York), influido por el positivismo lógico. Publicó diversos artículos y libros.

[103] James Roy Newman (1907-1966). Abogado, matemático e historiador de las matemáticas. Ocupó diversos cargos entre ellos el de jefe de inteligencia en la embajada en Londres. Escribió diversos artículos y libros sobre temas matemáticos.

[104] Douglas Richard Hafstadter (15 de febrero, 1945) filósofo, matemático y físico, se gradúa en matemáticas en la universidad de Stanford y recibe su Ph.D en física en la universidad de Oregón en 1975. es parte del equipo del laboratorio de Inteligencia Artificial del MIT y es titular de la cátedra de ciencias cognitivas en la universidad de Michigan.

[105] La lógica intuicionista, es el sistema lógico originalmente desarrollado por Arend Heyting para proveer una base formal para el proyecto intuicionista de Brouwer. El sistema enfatiza las pruebas, en vez de la verdad, a lo largo de las transformaciones de las proposiciones. Rechaza el principio del tercero excluido.

fundamentos de las matemáticas, contribuyendo a la aparición y definición de diversas lógicas. Las tres escuelas matemáticas mencionadas, utilizan explícitamente como herramienta demostrativa la lógica, no siendo la misma para un matemático intuicionista que para un formalista. Se utiliza el método axiomático y en muchos casos se usan expresiones que no son necesariamente bivalentes. Resulta el aporte de la escuela intuicionista en la unificación de las matemáticas, no en el sentido de Hilbert, ampliando la relación entre las diversas ramas de las matemáticas, al respecto Babini refiere:

> … después de la atenuación y declinación de la polémica acerca de los fundamentos de la matemática que esta adquiere los rasgos actuales: unificación y vinculación a través de las estructuras de temas muy diferentes que en la matemática clásica constituían capítulos separados, y una armonía cada vez más estrecha entre lógica, algebra y método axiomático. (Babini, 1967: 58).

Consideramos que lo expuesto se resume según expresiones dadas por Charles Peirce en el artículo *La Lógica de las Matemáticas un Intento de Desarrollar mis Categorías desde Dentro,* en la que menciona a la lógica y las matemáticas, así nos dice:

> … las matemáticas llevan a cabo su razonamiento mediante lógica utens (lógica para el uso) desarrollada por ella misma, y no necesita acudir a la lógica docens (lógica teórica), pues ninguna disputa acerca del razonamiento hace surgir en las matemáticas que necesite ser sometida a los principios de la filosofía del pensamiento por decreto (Peirce, 1897:1).

2.3. Enumerabilidad

Se dice que un conjunto es enumerable cuando sus elementos pueden ser colocados en una sucesión, con un primero y así sucesivamente, de forma que a todo elemento del conjunto le corresponde un único lugar en la mencionada sucesión. Esta forma de presentar los conjuntos, establece el concepto de contar los elementos del conjunto.

Otra forma de plantear la enumeración es mediante la relación de orden, ésta se define si cumple con lo siguiente:

- Reflexiva: Si $\forall$ x ϵ A $\Rightarrow$ ((x, x) ϵ R)
- Anti simétrica: Si $\forall$ x, y ϵ A $\Rightarrow$ ((x, y) ϵ R) (y, x) $\notin$ R)
- Transitiva: Si $\forall$ x, y, z ϵ A $\Rightarrow$ (((x, y) ϵ R ^ (y, z) ϵ R)) (x, z) ϵ R)

La relación de orden participa en la de la computabilidad, en el sentido de enumerar los elementos de un conjunto según lo establecido en la relación. En las matemáticas, existe la orientación conocida como conjuntista, considera importante la enumeración mencionada por estar presente en cierto sentido en la definición de la recursividad. En cuanto a las matemáticas existen otras corrientes, destacando que la gran mayoría de los matemáticos no toman en cuenta estas orientaciones, así lo indica Torretti:

> Esta corriente se autodenomina 'clásica', pero la llamaré 'conjuntista' porque coloca al centro de la matemática, en una forma u otra, la noción de conjunto y trabaja en fortalecerla. Iniciada por Dedekind... y Cantor… incorpora logros de Frege… Peano… Whitehead… y Russell… y recibe aportes de Hilbert… Zermelo… Tarski… Von Neumann… Gödel… Gentzen y muchos otros. Por otra parte, están los adversarios del conjuntismo – ilustres matemáticos como Kronocker… Poincaré… Brouwer… y Weyl… filósofos como Wittgenstein… y Lorezen… - que impugnan con poderosas razones sus ideas y prácticas más arraigadas, sin que la masa de los matemáticos les preste atención (Torretti, 1998: capítulo XI).

La enumerabilidad de un conjunto pone en correspondencia biunívoca a los elementos del conjunto con los números naturales, de forma que en un conjunto finito los elementos corresponderán hasta llegar a un número, que resulta ser el número cardinal. En los conjuntos infinitos, la enumerabilidad sólo puede sostenerse en los conjuntos que tienen por cardinalidad al número transfinito χ_0, así nos dicen Mosterín y Torretti:

> Un conjunto es numerable si y solo si es finito o contiene tantos elementos como hay números naturales. Todos los conjuntos finitos son numerables. De los infinitos, solo son numerables los más pequeños, es decir los que tienen la misma cardinalidad transfinita χ_0 (Mosterín y Torretti, 2002:402).

Para Cantor el infinito en los números no es en potencia, sino en acto, así:

> Cantor sabía bien que la matemática tradicional – lo que había llamado 'matemática clásica' en 1880… admitía el infinito solo como una potencialidad inalcanzable y rechazaba de plano el infinito actual (Torretti, 1998:28).

La forma de enumerar de Cantor en aplicación a los números naturales no es nuevo, ya se encuentra en el razonamiento de Galileo en 1638, lo menciona Penrose, cuando indica la relación del conjunto de los cuadrados de los números naturales hacia los números naturales: "Ya advertida por el gran físico y astrónomo Galileo Galilei... el conjunto de los cuadrados {0, 1, 4, 9…} también debe tener la misma cardinalidad que N" (Penrose, 2004:500).

Los números naturales tienen la propiedad de que pueden definir diversas correspondencias en sí misma, estas son de subconjuntos infinitos de los números naturales a los números naturales. Las correspondencias son definidas mediante funciones inyectivas[106]. Al respecto Penrose dice: "¿Qué es un número cardinal? Básicamente es el número de elementos en cierto conjunto, y consideramos que dos conjuntos tienen el mismo número de elementos si y solo si pueden ponerse en correspondencia 1-1 entre sí" (Penrose, 2004:5001).

[106] Función inyectiva es la función que hace corresponder a un elemento del dominio con un elemento del rango, si $x \neq y$ entonces $f(x) \neq f(y)$ y si $x = y$ entonces $f(x) = f(y)$

La cardinalidad de un sub conjunto finito es siempre menor o igual a la cardinalidad del conjunto, pero en el caso de los números naturales se encuentran subconjuntos infinitos que su cardinalidad es la misma que la de los números naturales.

Los números naturales son utilizados por Cantor para establecer la enumerabilidad de los números racionales, hace corresponder el conjunto NxN (números naturales por números naturales) a los racionales mayores de cero, así tenemos:

NxN	1	2	3	4	5	….
1	1/1	1/2	1/3	1/4	1/5	….
2	2/1	2/2	2/3	2/4	2/5	….
3	3/1	3/2	3/3	3/4	3/5	….
4	4/1	4/2	4/4	4/4	4/5	….
5	5/1	5/2	5/3	5/4	5/5	….
….	….	….	….	….	….	….

A esta correspondencia de números le dió un orden, según el sentido de las flechas mostradas. Esta es una enumerabilidad, que significa poner en un orden los números asociados de NxN a los números naturales.

NxN	1	2	3	4	5	….
1	1/1	1/2	1/3	1/4	1/5	….
2	2/1	2/2	2/3	2/4	2/5	….
3	3/1	3/2	3/3	3/4	3/5	….
4	4/1	4/2	4/4	4/4	4/5	….
5	5/1	5/2	5/3	5/4	5/5	….
….	….	….	….	….	….	….

La sucesión de los números según el orden establecido, nos proporciona la siguiente secuencia 1/1, 2/1, 1/2, 1/3, 2/2, 3/1, 4/1, …, si eliminamos aquellos números equivalentes[107] tendremos una enumerabilidad de los números racionales positivos, así nos dice Kleene: "El conjunto de los números racionales es también enumerable" (Kleene, 1952:16). Según la enumeración de Cantor, se concluye que la cardinalidad de los números Naturales es el mismo a la cardinalidad de los números Racionales.

En el caso de los números Enteros se establece un orden que se asocia la secuencia de números naturales impares a los números enteros negativos y la secuencia de números pares a los números enteros positivos, de esta forma se demuestra que la cardinalidad de los números enteros es la misma a la de los naturales: "El conjunto de los enteros puede ser enumerado alistando a estos en el siguiente orden 0, 1, -1, 2, -2, 3, -3,…" (Kleene, 1952:16).

De lo expuesto deducimos que en un conjunto, si lo separamos en partes disjuntas, cada una de estas es menor al todo y la unión de todas las partes resulta el todo. En el caso de los conjuntos infinitos, una parte puede tener la misma cantidad del todo, así también la unión de las partes en la que cada parte tiene la misma cantidad del todo, resulta el todo. Una ejemplificación de estas ideas es equivalente al de sumar infinitos, que resulta siempre un número infinito, como si el infinito se contuviese así mismo.

[107] En los números racionales se tienen diferentes números que expresan el mismo valor, por ejemplo 4/2, es equivalente a 2/1, 8/4…, así en la forma un número racional tiene infinitas representaciones. Como ya se mencionó, se deja un elemento de cada clase en los números Racionales.

2.4. Diagonalización

La diagonalización de Cantor es una forma de prueba matemática que muestra que los números reales no son numerables con respecto a los números naturales. La prueba original de Cantor es trabajada en el intervalo [0,1] de la recta numérica de los números reales, así lo indica Roberto Torretti: “El conjunto de los números reales comprendidos en un intervalo finito cualesquiera es más numeroso que el conjunto de los números enteros positivos” (Torretti, 1998:22).

La demostración de Cantor utiliza el método de reducción al absurdo y sigue el siguiente procedimiento:

1. Fijamos para la demostración los números reales comprendidos entre el 0 y el 1.
2. Los números reales seleccionados los ordenamos en una secuencia infinita, siendo esta: r_1, r_2, r_3...
3. Sabemos que los números reales entre 0 y 1, tienen la forma de tener un digito cero seguido de un punto y luego dígitos que expresan la parte decimal, tomando la forma: $0.d_1d_2d_3\ldots$, en la que cada d_i es un digito decimal.
4. Los números seleccionados los colocamos en una lista, uno seguido del otro, tal como se muestran en el siguiente ejemplo:

$$r_1 = 0.\ \underline{\mathbf{5}}\ 1\ 0\ 5\ 1\ 1\ 0...$$

$$r_2 = 0.\ 4\ \underline{\mathbf{1}}\ 3\ 2\ 0\ 4\ 3...$$

$$r_3 = 0.\ 8\ 2\ \underline{\mathbf{4}}\ 5\ 0\ 2\ 6...$$

$$r_4 = 0.\ 2\ 3\ 3\ \underline{\mathbf{0}}\ 1\ 2\ 6...$$

$$..\ = 0.\ ....$$

5. Los números mostrados tienen la forma: 0. x_{i1} x_{i2} x_{i3} x_{i4}…, donde cada número r_i está el intervalo numérico [0,1], y cada x_{ij} es un digito.
6. Construimos el número r de forma que contenga los dígitos que ocupan el lugar x_{nn}, de cada número r_i, aquellos que están en la diagonal (ejemplo los dígitos mostrados en negrita). El número que contiene los dígitos de la diagonal seria de la forma 0. x_{11} x_{22} x_{33}…, que en el caso resulta ser r = 0.5140…
7. Luego modificamos cada digito x_{nn} por otro, sumamos una unidad a cada digito y si resulta 9 lo cambiamos a 0, para nuestro ejemplo el número r' = 0.6251….
8. El número r' no podría estar en intervalo [0,1], dado que hemos tomado un digito de cada uno de los números (los de la diagonal) y hemos cambiado todos los dígitos, por lo tanto hemos definido un número que por su forma se encuentra en el intervalo [0,1] pero no está en la enumeración original, dado que todo los elementos de la diagonal han sido modificados. Concluyendo que los números Reales no son numerables.

> … hay infinitos conjuntos, considerados en la matemática, que no pueden ser enumerados fue mostrado por el famoso ´método de la diagonal´ de Cantor. El conjunto de los números reales es no-enumerable (Kleene, 1952:17).

El método en sí mismo se explica en palabras de Kleene: "Seleccione la fracción diagonal mostrada… cámbiese en ella cada uno de los sucesivos dígitos x_{nn}… la fracción resultante x'_{00} x'_{11} x'_{22} x'_{33}… representa un número real x que pertenece al intervalo pero no a la enumeración" (Kleene, 1952:12).

La enumerabilidad en los diversos conjuntos se debe a que contienen en su definición una relación de orden, esto identifica a los elementos del conjunto en una secuencia. Así nos dice Roberto Torretti.

> Cantor explica que un conjunto bien ordenado es un conjunto bien definido cuyos elementos están ordenados linealmente de tal modo que: B01 Hay un primer elemento, esto es, un elemento que precede a todos los otros. B02 Todo elemento que precede a otros tiene un sucesor inmediato, esto es, un elemento que le sigue y precede a cualquier otro elemento precedido por él. B03 Si A es una parte no vacía – finita o infinita – del conjunto y los elementos del conjunto que siguen a todos los elementos de A forman otra parte no vacía B, entonces B tiene un primer elemento (Torretti, 1998:34).

En la investigación de Cantor sobre la cardinalidad de los números reales es de 2^n, donde n es la cardinalidad de los números naturales, demuestra que la relación 2^N es mayor que N, expresa que la cardinalidad de los números reales es mayor a la cardinalidad de los números naturales, Berlinski lo dice en esta forma:

> Lo que Cantor llamo la potencia del conjunto es mayor que el propio conjunto… χ_0 es menor – debe ser menor – que el conjunto de sus subconjuntos que tiene cardinalidad $2^{\chi 0}$. Por tanto hay un número cardinal más allá de χ_0 y, por supuesto, por medio de ese mismo razonamiento hay otro, y otro más. Se da paso a una jerarquía trascendental, los números cardinales alineados en alguna parte del espacio más allá de los números naturales (Berlinski, 2007:172).

La conclusión a la que llega Cantor es que en un conjunto infinito y numerable, su conjunto potencia es innumerable, tal como ocurre con los números naturales, enteros y racionales (que ya mencionamos que son numerables y tienen la misma cardinalidad), mientras que los números reales no son numerables y su cardinalidad es la cardinalidad del conjunto potencia de los números naturales. Nos dice que dos conjuntos infinitos, uno puede ser más grande que el otro, como demostró que el conjunto de números reales tiene χ_1 elementos y es mayor a la cantidad de elementos del conjunto de los números naturales χ_0, expresado como: $\chi_1 > \chi_0$.

Esto nos lleva a colocar en el límite del marco teórico, que la función computable se define en los conjuntos numerables, en la medida que se establezca la enumerabilidad, asimismo, esta correspondencia se puede definir en universos no numéricos que son numerables, así lo expresa Boolos[108], Burgess y Jeffrey.

> El conjunto de todas las cadenas de finitas letras del alfabeto provee un ejemplo de una enumerabilidad de conjunto finito, el cual no es un conjunto de enteros positivos. Este conjunto es enumerable porque sus miembros pueden ser arreglados en una lista simple (Boolos, Burgess y Jefrey, 1988:5)

La enumerabilidad definida por Boolos, Burgess y Jeffrey, es una extensión de la definición de Cantor con la consideración que no solo trata de números, también se da en conjuntos de caracteres alfabéticos o símbolos que cumplen perfectamente con lo enunciado en cuanto a una relación de orden.

2.5. Funciones computables.

Kleene enuncia la tesis Church-Turing en el capítulo trece de su libro *Introduction to Metamathematics* (1952), identificándose y resaltando las conclusiones del trabajo de Church: *An Unsolvable Problem of Elementary Number Theory*, publicado en 1936, a la que denominó *Tesis de Church* haciéndola equivalente a lo expresado por Turing en su publicación: *On computable numbers with application to the Entscheidungsproblem* (1936).

El concepto de computar refiere al de calcular el resultado de una función

[108] George Boolos (nace en New York el 04 de setiembre de 1940, fallece el 27 de mayo de 1996). Se gradúa en matemáticas en Princeton, en 1961. Obtiene su PH.D en filosofía en Massachusetts Institute of Technology, bajo la dirección de Hillary Putnam.

mediante la ejecución de instrucciones, éstas son efectivas por que se ejecutan en acto, de forma que cualquier persona podría ejecutarlas sin mayor entendimiento, sólo siguiendo lo precisado, así lo enuncia Kleene:

> Supongamos que una persona tiene que computar el valor de una función para un conjunto dado de argumentos, siguiendo instrucciones efectivas pre asignadas. Al efectuar la computación, utilizará un número finito de símbolos o marcas distintas de alguna clase… Una sucesión de actos tales, debe deducirle desde una expresión simbólica que representa los argumentos, a otra expresión simbólica que representa el valor de la función. (Kleene, 1952:322)

El calcular el valor de una función para un argumento determinado en relación al procedimiento que lo ejecuta es la idea de computabilidad, para este fin, Kleene incorpora el concepto de la máquina de Turing, (será analizado en el siguiente capítulo). El concepto de computabilidad de Kleene, refiere al de función calculable.

La función calculable es asociada a un procedimiento efectivo que define la ejecución de los cálculos. El procedimiento está formado por una secuencia de instrucciones que obtienen el valor. La función f y los argumentos $r_1, r_2, \ldots, r_m$, tienen correspondencia con las instrucciones Ψ del procedimiento P.

Ejemplo: La suma de los n primeros números naturales $f(n) = 1 + 2 + \ldots + n$, asignamos el valor 1 a una variable que actuará como contador, iniciamos en cero al que contendrá el valor de la suma, luego realizamos la suma de los números de forma que cada vez que incrementamos en uno la variable, éste se suma al acumulador, el contador irá incrementando en uno hasta llegar al valor n.

En el caso de la composición de funciones también podemos establecer el criterio de computabilidad, dado que la composición de funciones define una función, así que dada una función es posible expresarla en composición de otras funciones, expresando en forma implícita un procedimiento.

Si g una función de k variables y g_1, g_2,…, g_k son funciones de n variables, entonces definimos función h de variables como la composición de las funciones f, g_1, g_2,…, g_k. $h(x_1, x_2, \dots ,x_n) = f(g_1(x_1,x_2,\dots , x_n), g_2(x_1, x_2, \dots , x_n), \dots , g_k(x_1, x_2, \dots , x_n))$.

Ejemplo: Expresamos la función h (x1, x2), según la composición de las funciones f, g_1, g_2 y g_3, tales que $f(x, y, z) = (x + y) \cdot z$, $g_1(x_1, x_2) = x_1 + x_2$, $g_2(x_1, x_2) = x_1 \cdot x_2$, $g_3(x_1, x_2) = x_1 - x_2$.

Entonces: $h(x_1,x_2) = f(x_1 + x_2, x_1 \cdot x_2, x_1 - x_2) = (x_1 + x_2 + x_1 \cdot x_2) \cdot (x_1 - x_2)$

Habíamos indicado que las funciones recursivas primitivas son expresadas mediante las funciones recursivas iniciales, pero no toda función recursiva es una función recursiva computable que se pueda reducir a funciones recursivas primitivas, como ejemplo las funciones recursivas de Ackermann[109].

La función recursiva de Ackermann es una función en la que sus valores iniciales son de fácil cálculo, pero a medida que se sigue calculando, los valores se hacen enormes y

[109] Wilhelm Ackermann (29 de marzo 1896 - 24 de diciembre 1962) matemático alemán. Es conocido, por la función de Ackermann nombrada en su honor, un ejemplo importante en la teoría de la computación. Se doctoró en 1925 con su tesis Begründung des "tertium non datur" mittels der Hilbertschen Theorie der Widerspruchsfreiheit, que fue una prueba de consistencia de la aritmética sin inducción. Desde 1928 hasta 1948 fue profesor en el instituto Arnoldinum en Burgsteinfurt, y desde entonces hasta 1961 enseñó en Lüdenscheid. Además, fue miembro de la Akademie der Wissenschaften (Academia de las Ciencias) en Göttingen, así como profesor honorífico de la Universidad de Münster en Westfalia.

exigen mayor potencia de cálculo. Así tenemos definida la función:

$$A(m, n) = \begin{cases} n + 1, & \text{si } m = 0 \\ A\,(m - 1, 1), & \text{si } m > 0 \text{ y } n = 0 \\ A\,(m - 1, A(m, n - 1)), & \text{si } m > 0 \text{ y } n > 0 \end{cases}$$

Si evaluamos las funciones A (m,n) para valores pequeños de m, tales como 1, 2, ó 3, esta crece lento con respecto a n (a lo sumo exponencialmente), pero para m = 4, crece más rápido:

$$A(4, 2) \approx 2 \times 1'019{,}728$$

$$A(4, 3) > \text{cantidad muy grande}$$

Esta función recursiva es interesante, dado que cada vez que se avanza en el cálculo, van creciendo los valores en proporciones mayores. En la actualidad esta función se utiliza para medir la capacidad o velocidad de cálculo que tiene una determinada computadora. Por otro lado es un ejemplo emblemático que nos introduce en el tema de la complejidad de cálculo mediante la recursividad.

Siguiendo con la definición de computabilidad, Kleene, demuestra la equivalencia del concepto de la computabilidad con el concepto de la recursividad parcial[110] o funciones completamente definidas con la recursividad general. Esto es importante, ya que establece el criterio de función computable como sinónimo de función recursiva.

[110] Recursividad total corresponde a la definición de una función que está definida en todo el dominio de esta, mientras que recursividad parcial corresponde a una función que abarca solo una parte del dominio.

> Nuestro principal objetivo es ahora demostrar la equivalencia de la computabilidad con la recursividad parcial o, cuando solo se consideran funciones completamente definidas, con la recursividad general. (Kleene, 1952:326).

En el análisis de Kleene sobre el concepto de computabilidad nos proporciona una expresión funcional que está referida a las funciones recursivas, así tenemos:

> Extendemos ahora nuestras nociones al caso de computación a partir de l funciones completamente definidas $\psi_1,\ldots,\psi_l$ (brevemente ψ), de $m_1, \ldots, m_l$ variables, respectivamente. Aquí la idea se modifica por la suposición de que cualquier valor de una de las funciones ψ, si es exigido en el curso de la computación. Será desde luego facilitado (Kleene, 1952:327)

Kleene refiere al problema de decisión en relación a lo propuesto por Hilbert de acuerdo a obtener un procedimiento. Para este fin utiliza las funciones en relación a las funciones recursivas.

> En el caso de que ψ sean funciones y predicados, cabe discutir la reducción del problema de decisión para P (el problema de cálculo para φ) a los respectivos problemas para ψ en el sentido de obtener un procedimiento uniforme en ψ así como en $x_1,\ldots, x_n$; o, dicho brevemente, de establecer que P es efectivamente decidible (φ efectivamente calculable) uniformemente a partir de ψ (Kleene, 1952:287).

Kleene extiende el concepto de computabilidad a símbolos, específicamente a palabras como cadenas de letras que forman parte de un alfabeto, así tenemos:

> Supongamos, además, ahora que es dada una lista finita $(A_1, B_1),\ldots, (A_n, B_n)$ $(n \geq 1)$ de pares de palabras; llamamos a esta lista el diccionario. Decimos que dos palabras R y S son inmediatamente equivalentes (por el diccionario dado) si R y S son de la forma respectiva UA_iV y UB_iV o de las formas respectivas UB_iV y UA_iV, para algunas palabras U y V (Kleene, 1952:345).

La recursividad y la computabilidad tienen aplicación práctica en la formalización de los lenguajes computacionales, éste es uno de los campos en la ciencia de la computación que ha mostrado el éxito del planteamiento de Church y Kleene. De otro lado, el análisis realizado por Kleene toma en cuenta la opción de sustitución de palabras, definiendo un criterio de equivalencia:

> En otros términos, si R se puede transformar en S mediante la sustitución de una parte de A_i por su correspondiente Bi en el diccionario, o a la inversa. Llamamos a dos palabras P y Q equivalentes (por el diccionario dado) si hay una secuencia finita $R_1,\ldots, R_l$ de palabras tal que R_1 es P y R_l es Q (Kleene, 1952:345).

Kleene enuncia que el problema de las palabras para semigrupos es insoluble esto lo hace en el teorema XXXI, destacándose la existencia del límite en los procesos computacionales para obtener la traducción efectiva de palabras: "El problema de las palabras para semigrupos es insoluble; de hecho, hay un alfabeto y diccionario particular tales que no hay algoritmo para decidir si cualesquiera dos palabras son equivalentes" (Kleene, 1952:345).

2.6. Cálculo Lambda.

El cálculo lambda[111] fue introducido por Church en el año 1930, en su etapa inicial tenía la dificultad expresiva para combinar los términos, luego en 1934 Church restringe la libertad introduciendo el cálculo lambda con tipos, que define de manera precisa la secuencia de introducción de los valores en las variables como "lenguaje", permitiendo la descripción de las funciones matemáticas y sus propiedades para evaluarlas. Este se fundamenta en su formalización de una manera recursiva de calcular una función basada en valores anteriores.

Desarrollamos el tema del cálculo lambda, porque contiene el sentido del concepto

[111] El cálculo lambda fue desarrollado por Alonso Church con ayuda de C. Kleene describe funciones matemáticas, un texto que trata sobre las funciones lamba se encuentra en http://lml.ls.fi.upm.es/rsd/Slides/lambda.pdf. Lunes, 30 de junio de 2008, 12:00 horas.

de computabilidad manifestado por Church, así damos algunas precisiones que explican las formas que tiene para realizar el cálculo de funciones.

Se dice que es una α-equivalencia (alfa equivalente) de dos funciones cuando las dos funciones son la misma, y se representa la equivalencia con el símbolo ≡. Ejemplo $\lambda x. x \equiv \lambda y. y$

La evaluación de una expresión se compone de pasos de reducción donde cada uno de los pasos se obtiene por reescritura, llamándolos δ-reducciones (delta reducciones), las reducciones con reglas que se transforman en constantes.

Ejemplo, el cálculo de * (+ 1 2) (- 4 1) sería como sigue:

$* (+ \ 1 \ 2) (- \ 4 \ 1) \rightarrow_{\delta} * (+ \ 1 \ 2) \ 3$

$\rightarrow_{\delta} * \ 3 \ 3$

$\rightarrow_{\delta} 9$

Se dice que es una β-reducción (beta reducción): Al proceso de copia del valor del argumento en la variable, en el rango donde corresponde el reemplazo. El reemplazando de todas las ocurrencias de la variable equivale a una sustitución, si la regla se utiliza en sentido contrario se dice β-expansión.

Ejemplo: $(\lambda x. * x x) \ 2 \rightarrow_{\beta} * \ 2 \ 2 \rightarrow_{\delta} 4$

También se puede expresar directamente: $(\lambda x. * x x) \ 2 \rightarrow_{\beta\delta} 4$

Ejemplo: $((\lambda\ x.\ \lambda\ y.\ *\ x\ y\ )\ 7)\ 8 \rightarrow_{\beta} ((\lambda\ y.\ *\ 7\ y\ )\ 8 \rightarrow_{\beta} *\ 7\ 8 \rightarrow_{\delta} 56$

También se puede expresar directamente: $((\lambda\ x.\ \lambda\ y.\ *\ x\ y\ )\ 7)\ 8 \rightarrow_{\beta\beta\delta} 56$

En el cálculo lambda se permite expresiones que operan valores lógicos, así tenemos las siguientes definiciones: λ fxy. x = true, λ fxy. y = false

Dado la expresión true y false, definimos el caso de negación.

λ t.t (λ fxy. y) (λ fxy. x) = λ t.t true false = not

Ejemplo: Probar que not true = false

(λt.t false true) true	por definición de not
(λt.t (λab.b) (λcd.c)) (λxy.x)	por las equivalencias de true y false
(λxy.x) (λab.b) (λcd.c)	haciendo t = λxy.x
λy.(λab.b) (λcd.c)	haciendo x = (λab.b)
(λab.b)	como y no figura como variable de reemplazo
false	

Así tenemos las funciones lógicas and y or, definidas como sigue:

λxy.x y (λxy.y) = and

λxy.x (λxy.x) y = or

Ejemplo: Probar que or true false = true

(λxy.x true y) true false	por la definición de or
(λxy.x (λab.a) y) (λcd.c) (λef.f)	por equivalencias de true y false
(λcd.c) (λab.a) (λef.f)	haciendo x = (λcd.c), y = (λef.f)

(λab.a)	haciendo c = (λab.a), la variable d no figura
true	

Church en su trabajo publicado en 1936, presentó como definición de números naturales la siguiente notación: $C_0 = \lambda$ fx.x, $C_1 = \lambda$ fx.fx, $C_2 = \lambda$ fx.f(fx) = λ fx.f^2x. ..., $C_n = \lambda$ fxf^n (x), $C_{n+1} = \lambda$ fx. f^{n+1}(x). Los C_i representan el numero i y permiten definir funciones aritméticas, así tenemos la función sucesivo: λ n f x. n f (f x), o la función suma de números naturales: λ m n f x. m f (n f x).

Ejemplo: Probar que sucesivo de 1 es 2

(λnfx.nf(fx)) (λga.ga)	considerando la definición de sucesivo y C_1
λfx.((λga.ga) f (fx))	haciendo n = λga.ga
λfx.f(fx)	haciendo g = f, a = (fx)
2	definición del número 2

Ejemplo: Probar que la suma de 1 y 2 resulta 3

λmnfx.mf(nfx) (1) (2)	considerando la definición de suma
λfx.1f(2fx)	haciendo m = 1, n = 2
λfx.((λga.ga) f (2fx))	haciendo C_1= λga.ga
λfx. f(2fx)	haciendo C_2 = λhe.h(he)
λfx.f(λhe.h(he) (f) (x))	haciendo h = f, e = x
λfx.f(f(fx))	simplificando
λfx.f^3x	
3	definición del número 3

2.7. Tesis de Church.

El trabajo de investigación de Church la ubicamos dentro del proyecto propuesto por Hilbert, en relación contraria a los ideales que son fórmulas que no es posible obtener un valor mediante un conjunto de pasos, Martínez y Pineiro lo explican:

> Hay fórmulas cuya verdad o falsedad no pueden ser determinadas mecánicamente, en una cantidad finita de pasos. A estas fórmulas Hilbert las llamaba ideales en contraposición con las que llamaba fórmulas con sentido, que son aquellas cuya verdad o falsedad se puede determinar en una cantidad finita de pasos. (Martínez, Pineiro, 2009: 154).

Church en su artículo *An Unsolvable Problem of Elementary Number Theory*, presenta como procedimiento para obtener el valor de una función mediante la función recursiva a la que llamó cálculo lambda. Así lo expresa en la introducción del documento "Hay una clase de problemas de la teoría elemental de números que pueden estar en la forma requerida a encontrar una función de cálculo efectivo *f* de n enteros positivos tal que $f(x_1, x_2, \ldots, x_n) = 2$" (Church, 1936:345).

Cuando se refiere a la función y el valor 2 como resultado, al píe de nota indica que el valor no es trascendental que es accidental, es decir se orienta a la obtención de un resultado. Aclara para el caso de la solución en el teorema de Fermat, la función *f* es efectivamente calculable y que es parte esencial del problema.

Church define como propósito el presentar una definición de función efectivamente calculable, indicándolo en el pie de nota en la segunda página.

> Como se verá, esta definición de calculabilidad efectiva puede ser una de las dos formas equivalentes, (1) que una función de enteros positivos se llama efectivamente calculable si es definible en el sentido de $2 , (2) que una función de enteros positivos que puede ser llamado

efectivamente calculable si es recursivo en el mismo sentido de $4 (Church, 1936,346).

El trabajo se orientó en demostrar que si el sistema de Bertrand Russell y Whitehead expresado en *Principia Matemática* es consistente, entonces es insoluble, entendiéndose que la demostración tiene relación con la investigación sobre la incompletitud formulada por Gödel y la propuesta formalista de Hilbert. Para demostrar que existen proposiciones no solubles en la teoría elemental del número menciona como ejemplo el teorema de Fermat[112] el que fuera demostrado[113] en 1995, con resultado de que no es posible encontrar una solución para exponentes mayores a dos.

Indica que la finalidad de su documento es el proponer una definición de calculabilidad efectiva, en la que demuestra que no todos los problemas de la teoría de números elementales son solubles.

Utiliza símbolos que son una forma de enumeración en referencia a los números naturales, estableciendo el concepto de función bien formada, así para el número "n" que resulta ser: $n \rightarrow \lambda ab\,.\,a\,(\,a\,(\,...\,a\,(\,b\,)\,...\,)\,)$, en la que "a" se repite "n" veces.

Para la conversión de una expresión A por B, define la notación $S_n^* M$, en la que se sustituye el valor n para la variable x en todo M.

En el primer teorema se indica que si una formula está en forma normal, no es

112 Pierre Fermat (1601-1665), ilustre matemático que afirmó haber probado varios teoremas, específicamente $x^n + y^n = z^n$, que dice que se puede obtener un número n que resulta de elevar a la potencia tres números enteros, ejemplo $x^2 + y^2 = z^2$que es expuesto en el teorema de Pitágoras en la geometría, el caso es que Fermat escribe al margen de su ejemplar de la *aritmética de Diofanto* que ha resuelto el problema en una parte de la hoja del texto que propone el problema habiendo poco espacio en la hoja. El problema se conoce como el último teorema de Fermat

113 Andrews Wiles publica la solución del teorema de Fermat, Annals of Mathematics Vol 142, 1995 pp. 443-551, hay un texto interesante sobre la historia de la demostración de Amir D. Aczel (2005).

posible una reducción de esta. Con respecto a los dos teoremas siguientes, expresa: Si una fórmula tiene una forma normal, ésta es única y cualquier secuencia de reducciones de la fórmula terminará en una forma normal.

El tercer teorema manifiesta que si una fórmula tiene forma normal toda parte bien formada de la fórmula tiene forma normal, resultando este teorema ser una definición de lo que se define como forma normal.

Church afirma que cualquier función lambda definible de números enteros positivos, proporciona un algoritmo, en el proceso de reducción de fórmulas. La evaluación mediante las funciones recursivas es un cálculo efectivo. Por lo tanto el cálculo lambda expresa adecuadamente el concepto de algoritmo.

> Esto es claro que, en el caso de cualquier función definible de enteros positivos, el proceso de reducción de fórmulas a formas normales provee un algoritmo para los cálculos efectivos de valores particulares de la función (Church, 1936:349).

Define como fórmula la sucesión de símbolos $r_1r_2...r_n$ que equivalen a la representación de un número que resulta estar en la forma del producto de números primos: $2^{t1}3^{t2}...p_n^{tn}$, estos son números Gödel[114], con la observación de que Gödel utiliza los números 11 y 13 para expresar variables en su teorema de incompletitud, mientras que Church indican que pueden referirse a más de un tipo de fórmula, porque corresponden a más de un símbolo "Gödel asignó dichos números para referirse a variables numéricas y ... a variables proposicionales" (Nagel y Newman, 1959:52).

---

[114] Número Gödel, es el número utilizado por Gödel en la demostración del teorema de la incompletitud, este número se construye como el producto de potencias de números primos.

Church considera que en una función de números de enteros positivos sobre ecuaciones recursivas se determina que son recursivas si existe un algoritmo que las evalué, así para cualquier valor de la función que puede ser efectivamente calculado puede ser expresada mediante una función, tal como: $f_{ni}^{i}(k_1^{i},k_2^{i}, \ldots ,k_{ni}^{i})= k^{i}$, en la que los valores de k son números enteros.

En el teorema IV indica que si F es una función recursiva de dos números enteros positivos y si para cada x existe un número y tal que F(x,y) > 1, entonces la función F* para F*(x) es igual al menor entero positivo y para F(x,y) >1, entonces es recursivo.

En el siguiente teorema indica que si F es una función recursiva de un entero positivo, y si existen infinitos números enteros positivos x para el que F(x) > 1, entonces la función F^{o} para cada entero positivo n, $F^{o}(n)$ es igual a la n-esimo número entero positivo x para el F(x), estableciendo una relación entre las funciones F(x) y $F^{o}(n)$.

Church incluye en su trabajo teoremas que fueron demostrados por Kleene, con la diferencia de que los trata como números enteros positivos, mientras Kleene refiere a los números naturales, estos teoremas tratan sobre propiedades de funciones recursivas, así indica que si existe una fórmula bien formada es una representación de un número Gödel en el sentido de ser expresado como producto de números primos, entonces es recursiva.

Define como fórmula bien formada a las expresiones del cálculo lambda, presentando en el teorema VII como que son recursivas enumerables y en el teorema VIII enuncia la función de dos variables, cuyo valor, cuando se trata de fórmulas bien formadas F y X, la fórmula {F}(X) es recursiva.

Precisa en el teorema X que es recursiva una función para cada una de las fórmulas bien formadas de un número entero positivo, también en sentido contrario. En los teoremas XI y XII indica que la convertibilidad inmediata entre fórmulas bien formadas es recursiva, y que es posible asociar simultáneamente las fórmulas bien formadas de una enumeración de fórmulas.

En el teorema XIII y XIV se define la propiedad de una fórmula bien formada que se encuentra en lo que denomina principal forma normal que resulta ser recursiva, y que el conjunto de fórmulas bien formadas se encuentran en la principal forma normal y son recursivamente numerables.

Church concluye en los teoremas XVI y XVII que cada función recursiva de enteros positivos es lambda definible. Es en esta parte donde relaciona directamente las funciones recursivas con el cálculo lambda que contiene el concepto de algoritmo.

En tal sentido Church define el concepto de cálculo efectivo, como que es un algoritmo para obtener los valores en una función de enteros positivos. Concluye que el cálculo efectivo puede ser obtenido de cualesquiera de los dos métodos: (1) mediante la definición de una función efectivamente calculable si existe un algoritmo para el cálculo de sus valores, (2) por la definición de una función F para calcular efectivamente si para cada entero positivo m existe un entero positivo n tal que F(m)=n. En el criterio de función recursiva toma en cuenta el concepto de invariantes[115] para extender lo que ha expuesto en

[115] La teoría de invariantes permite obtener información de un objeto a través del estudio de otro objeto más sencillo. Las conjeturas de isomorfismos nos relacionan diferentes invariantes y facilitan el cálculo de los mismos.

su documento.

Después de que Church presenta los teoremas mencionados, en opinión de Torretti "está listo para dar la definición de calculabilidad efectiva" (Torretti, 1998:373). Así Church lo expresa:

> Ahora definimos la noción, ya comentada de una función efectivamente calculable de enteros positivos identificándola con la noción de una función recursiva de enteros positivos (o de una función) definible de enteros (positivos). Pensamos que esta definición se justifica por las siguientes consideraciones, en la medida en que sea posible obtener una justificación positiva para la elección de una definición formal correspondiente a una noción intuitiva (Church, en 1936, tomado de Torretti, 1998:356)

Torretti afirma que Church respalda la definición de que la función recursiva y la de lambda–definible están en la misma extensión, Church lo dice:

> El hecho de que dos definiciones de calculabilidad efectiva sean tan distantes entre sí y sin embargo, igualmente naturales resulten equivalentes refuerza las razones aducidas más abajo para creer que ellas caracterizan dicha noción de modo más general que es computable con nuestra habitual comprensión intuitiva de la misma (Church, 1936: 346)

Church continúa con la demostración y enuncia el Lema: El problema para encontrar una función recursiva de dos fórmulas A y B cuyo valor es 2 o 1, es equivalente al problema para encontrar una función recursiva de una formula C cuyo valor es 2 ó 1.

En el teorema XVIII afirma que no hay función recursiva de una formula C, cuyo valor es de 2 o 1. Luego presenta el Corolario 1 que dice: El conjunto de fórmulas bien formado que no tienen forma normal no es recursiva numerable. Luego el Corolario 2 que dice: Que una función F de un entero positivo definido por la regla de que F(n) será igual a 2 o 1, es una representación del número Gödel para la fórmula que tiene forma normal.

Entonces F es un ejemplo de una función no recursiva de enteros positivos. Concluyendo en el teorema XIX que dice que no hay función recursiva de dos fórmulas A y B, cuyo valor es 2 o 1.

Church argumenta su investigación valiéndose del cálculo lambda en el mismo sentido de la función recursiva general enunciada por Gödel, así Torretti enuncia: “Se vale para ello del concepto de función λ-definible desarrollado por él y Kleene – cuya extensión demostrablemente coincide con la del concepto gödeliano de función recursiva general” (Torretti, 1998: 369).

Observamos que las funciones recursivas permiten definir cálculos que se ejecutan en un número determinado de pasos que están en correspondencia a las funciones recursivas generales. De lo expuesto por Church se deduce que una función es efectivamente calculable si y sólo si es recursiva.

Church en su trabajo propone que el cálculo lambda es equivalente al de recursividad correspondiéndole al de algoritmo que se encuentra intuitivamente en el cálculo, enunciando: “Ya se ha señalado que, para cada función de enteros positivos que es efectivamente calculable sólo en el sentido definido, existe un algoritmo para el cálculo de sus valores” (Church, 1936:356).

De otro lado, la tesis de Church nos lleva a reflexionar sobre la negación de lo expresado, en el sentido que si una función no es recursiva entonces no es efectivamente calculable, específicamente se considera la calculabilidad como aplicación de las funciones recursivas en los números naturales, por lo tanto la reflexión intuitiva de algoritmo se sustenta en la propiedad de los números definidos en funciones recursivas.

Se interpreta lo expuesto como un criterio que supera la definición solo matemática, pero refiere a la calculabilidad, Al respecto, Roberto Torretti afirma: “Por eso, no parece justo describir la tesis de Church como una conjetura matemática que aguarda ser demostrada. La veo más bien como una decisión de aceptar la computabilidad como criterio de calculabilidad” (Torretti, 1998:376).

Kleene en su libro *Introduction to Metamathematics* dedica dos capítulos (XII y XIII) para sustentar la evidencia de la tesis de Church, afirmando que el concepto de calculabilidad efectiva de una función es de naturaleza intuitiva, por lo tanto la tesis no puede ser demostrada, es una hipótesis: “La tesis puede ser considerada como una hipótesis acerca de la noción intuitiva de calculabilidad efectiva, o, una definición matemática de calculabilidad efectiva” (Kleene, 1056:289).

Kleene argumenta sobre cuatro evidencias que sostienen la tesis Church-Turing, siendo éstas:

Primera evidencia: Heurística, como aquella que significa el obtener un valor de una función, como se explica el concepto de algoritmo según el cálculo lambda.

Segunda evidencia: La equivalencia con diversas formulaciones, como la explicada en la recursividad general, lambda definibilidad[116], y computabilidad[117], precisándola de

[116] Kleene indica que la recursividad general y lambda definibilidad fueron dados por Church en 1933 y Kleene en 1935.
[117] Kleene sostiene que el concepto de computabilidad fueron dados por Turing en 1936 y Post en 1936.

la siguiente manera: “La equivalencia de las funciones definibles con las funciones recursivas generales fue mostrada por Church 1936 y Kleene 1936” (Kleene, 1952:290).

Tercera evidencia: Al concepto formulado por Turing en una máquina de computar. Para Kleene la tesis de Church es coextensiva con la definición de calculabilidad proporcionada por Turing.

Cuarta evidencia sobre las lógicas y algoritmos simbólicos indicando que Church presenta la esencia del significado de computable.

La tesis de Church no puede demostrarse porque se describe en términos de funciones recursivas que resultan ser su propia definición; por lo que la demostración caería en un círculo vicioso, pero la definición de cálculo efectivo mediante el cálculo lambda en relación al de algoritmo tiene sentido en la lógica, en la medida de estudiarlo como un lenguaje formal que hace cálculos en los números enteros.

2.8. El método logístico y Turing.

Church consideró que la máquina de Turing expresaba adecuadamente el concepto de procedimiento y recogía de mejor manera el concepto intuitivo de algoritmo, en su libro *Introduction to Mathematical Logic*, de 1956, en el capítulo de introducción refiere al método logístico, como a un lenguaje formalizado con un vocabulario y reglas, resaltando el interés de los lenguajes formalizados con la capacidad expresiva de una teoría:

> Nuestros intereses en los lenguajes formalizados no se centran en su uso real y práctico como lenguajes, sino en la teoría general de dicho uso y en sus posibilidades. Siempre que empleemos un lenguaje, llamaremos a este lenguaje el lenguaje objeto y a aquel el meta lenguaje (Church, 1956:

48)

Church define los requisitos de efectividad como sistema logístico.

> (I) La especificación de los símbolos primitivos será efectiva en el sentido de que hay un método por el cual, siempre que un símbolo es dado, puede ser determinado efectivamente si es uno de los símbolos primitivos. (II) la definición de fórmula bien formada será efectiva en el sentido que hay un método… (III) La especificación de los axiomas será efectiva en el sentido que hay un método, en que siempre hay una inferencia inmediata (Church, 1956: 50-51).

Church nos dice que después de establecer un sistema logístico aún no se tiene un lenguaje formalizado. Para esto se requiere de una interpretación, nos dice:

> Esto requerirá un metalenguaje más extenso que la parte restringida del inglés, usada para establecer el sistema logístico. En todo caso, no se procederá por medio de traducciones en las fórmulas bien formadas a las expresiones el inglés, sino más bien por medio de las reglas semánticas (Church, 1956:54)

Para Church un método logístico es un lenguaje que requiere de reglas semánticas. En el caso del cálculo lambda, resulta un sistema logístico en cuanto y cuando permiten obtener un cálculo numérico, siendo las reglas semánticas las que interpretan la notación, estableciendo una jerarquía de ejecución.

El trabajo realizado por Church a lo largo de su vida, en relación a temas académicos se centró en diversos campos de las matemáticas y de la lógica, al respecto María Manzano dice: "suyo es también Introduction to Mathematical Logic, Vol. I, el libro que definió que era la lógica, el enfoque y los temas básicos" (María Manzano, 1999: 107). Manzano cuenta que visitó en 1996 el CSLI de Stanford y John Etchemendy le relató una anécdota:

> Ocurrió en el curso de 1983-84, habían invitado a Alonzo Church al CSLI para dar una conferencia y Etchemendy lo fue a recoger y le estuvo enseñando el centro. En aquel momento tenían bastantes ordenadores Xerox Dandelion con procesadores LISP. Y puesto que dicho lenguaje está basado en el cálculo Lambda de Church (extremo que parece negar su creador, John Mc. Carthy, asegurando que fue un descubrimiento paralelo), John le hizo una demostración en una de aquellas Dandelion, explicándole que el lenguaje LISP está inmediatamente basado en el cálculo lambda. Church no pareció entusiasmarse y al final, para justificar su evidente falta de interés, le comentó que él no sabía nada de ordenadores, pero que había tenido un alumno, llamado Alan Turing, que sabía bastante de eso: (María Manzano, 1999: 108-109).

El desconocimiento de Church sobre temas de la ciencia de la computación queda evidenciado en una entrevista realizada por William Aspray en 1948 sobre el tema de la comunidad matemática en Princeton de 1930, Church no recuerda inmediatamente a Alan Turing. Tomándolo en cuenta, después de varias interrogaciones, en la que Aspray le pregunta directamente por Turing, y lo recuerda como el que tenía su propio proyecto, así tenemos:

> Aspray: dirigiste la tesis de Alan Turing?
> Church: bien, él estuvo en Princeton, pero no solo bajo mi supervisión, porque, por supuesto había trabajado con M.H.A. Newman en Inglaterra. Fue mientras que trabajaba con Newman en el que salieron sus ideas originales (Aspray, 1985).

Church no consideró a Turing entre sus alumnos, a pesar que dirigió su tesis doctoral, así lo afirma en la entrevista:

> Church: si me olvidé de él y cuando yo estaba hablando de mis propios estudiantes de postgrado. La verdad es que él no era realmente mío. Llegó a Princeton como estudiante de Postgrado y escribió su tesis ahí. Este fue su trabajo acerca de Lógica Ordinal (Aspray, 1985)

La idea detrás de las computadoras digitales puede explicarse diciendo que se trata de máquinas cuyo objetivo es ejecutar cualquier operación que puede realizar una computadora humana[118].

Alan Turing

CAPÍTULO III

AUTÓMATAS Y TEORÍA DE TURING.

En el presente capítulo examinamos el sentido de lo computable en la teoría de Turing, con fines de elucidar la relación con el concepto de algoritmo, estas resultarán necesarias para sustentar el mecanismo del funcionamiento de la máquina de Turing, destacando la importancia del procedimiento como secuencia de instrucciones, que para ser ejecutado, no requieren de ninguna comprensión.

El algoritmo en la teoría de Turing está expresado implícitamente, cuando define el concepto de máquina, en el sentido de ser un dispositivo que ejecuta instrucciones muy básicas. Así lo dice Wittgenstein[119]: "Si el cálculo nos aparece como una actividad maquinal, entonces la máquina es el ser humano que realiza el cálculo" (Wittgenstein, 1978:195), esta expresión resulta equivalente a decir que las máquinas de Turing son personas que calculan.

[118] En la *Máquina de Computación y la Inteligencia*, de 1950.

[119] Ludwig Josef Johann Wittgenstein (Viena, Austria, 26 de abril de 1889, Cambridge, Reino Unido, 29 de abril de 1951) filósofo y lingüista austríaco, posteriormente nacionalizado británico. Publicó el libro: *Tractatus logico-philosophicus*, influyó en los positivistas lógicos del Círculo de Viena, del que nunca se consideró miembro. Discípulo de Bertrand Russell en el Trinity College de Cambridge, donde llegó a ser profesor.

El concepto de computabilidad que manifiesta Turing se encuentra en sus diversas publicaciones, realizadas a lo largo de su vida científica, hemos escogido ocho documentos, porque consideramos que explican suficientemente el concepto computable en la teoría de Turing[120]. Los documentos son los siguientes:

1) On computable numbers, with application to the Entscheidungsproblem (1936).

2) Systems of logic based on ordinals (1938),

3) Intelligent Machinery (1948),

4) Computing machinery and intelligence (1950),

5) Can Digital Computers Think (1951),

6) The Chemical Basis of Morphogenesis (1952)

7) The Chess (1953)

8) Solvable and Insolvable Problems (1954).

Afirmamos que la definición de computabilidad en Turing está en relación a la noción intuitiva de algoritmo, que trata sobre operar lógicamente dispositivos "físicos". Si bien los conceptos utilizados por Turing son abstractos y corresponden a dispositivos que son ejecutados siguiendo reglas, en términos físicos, las instrucciones manejan automáticamente una máquina, en función del resultado deseado.

El concepto de computabilidad de la máquina Turing se relaciona al de "inteligencia", como naturaleza operacional con características diferentes a la cognitiva

[120] Los documentos corresponden a los trabajos de investigación de Alan Turing, publicados en los años: 1936 (define la máquina Turing), 1938 (su tesis doctoral en matemáticas), 1948 (informe cuando trabajaba en NPL) y 1950 (presenta el conocido Test de Turing), 1951 (respuesta al debate sobre el tema de inteligencia en las máquinas, 1952 (estudio sobre morfogénesis), 1953 (estudio sobre el ajedrez) y 1954 es un regreso al tema tratado en 1936).

como ocurre en el hombre, como una simulación de tareas que hace un hombre.

La opinión de Turing sobre la "inteligencia" en las computadoras ha propiciado debates en el campo de la filosofía, así también se ha aprovechado y obtenido resultados interesantes en el campo de la teoría de los lenguajes de la computación,[121] en relación a la teoría de los autómatas en aplicación del concepto de máquina. En cuanto al concepto de "inteligencia" observamos que se dan diversas interpretaciones, debido a esto la lectura tiene varias orientaciones que pretendemos esclarecer. Si consideramos las publicaciones de Turing en su conjunto, observamos un cuerpo teórico, al que denominamos Teoría de Turing.

Sostenemos como hipótesis que Turing realizó su investigación en el campo de la ciencia de la computación, en relación a máquina e instrucciones, habiendo tratado diversos temas que consideramos que forman parte de un cuerpo teórico, iniciándose en la solución negativa al problema *Entscheidungsproblem*, sobre funciones computables hasta llegar a sus estudios sobre aspectos relacionados al funcionamiento del cerebro del hombre.

3.1. La Máquina de Turing.

En 1936, Turing publica su trabajo en la Sociedad de Matemáticas de Londres, titulado: *On Computable Numbers, with application to the Entscheidungsproblem,* en el que define un dispositivo abstracto, que fue nombrado máquina de Turing. Éste es un

---

[121] Para mayor información consultar: Gramáticas de Lenguajes Formales por Chomsky N. http://www.chomsky.info/articles.htm. Consultado el Sábado, 26 de Junio del 2010, 11:30 horas, así también y el documento sobre autómatas elaborado en la Universidad de Murcia, España. http://perseo.dif.um.es/~roque/talf/Material/apuntes.pdf

concepto matemático basado en un procedimiento secuencial que se ejecuta en la máquina. La investigación de Turing concluye en la demostración de que no es posible obtener un procedimiento secuencial que calcule todas las proposiciones matemáticas, contradiciendo el planteamiento formulado por Hilbert, quien sostenía que las matemáticas son decidibles. Se supone que Turing conoció el artículo, publicado por Church en 1936, observando la equivalencia de los resultados con los que él había obtenido sobre el *Entscheidungsproblem;* en el mismo sentido iniciado por Gödel. Así lo indica Coello[122]:

> Turing tenía lista su investigación en abril de 1936, pero debió retrasarla, porque al mismo tiempo Church había llegado a la misma conclusión, en forma diferente, utilizando un cálculo basado en funciones recursivas. Así que Turing decidió hacer una nota de forma que se incluya como referencia el trabajo de Church. Ambos habían llegado al mismo resultado simultáneamente pero de formas diferentes. (Coello, 2004: 112).

La máquina de Turing es estudiada en la teoría de autómatas: Los autómatas son modelos abstractos de máquinas lógicas que ejecutan instrucciones mediante procedimientos. Su aplicación es diversa y es conocida como fundamento en los algoritmos, estudiados en la teoría de los lenguajes de computadoras[123]. La máquina de Turing es un autómata base, todos los demás tipos de autómatas se definen como deducidas de ella. La definición contiene la especificación de un procedimiento secuencial, que ejecuta instrucciones de naturaleza elemental; también se le suele asociar la formulación del cálculo lambda[124]. Turing en su tesis doctoral en la universidad de Princeton, trata sobre el sistema de lógica basado en ordinales. En esta investigación

[122] Carlos A Coello, nace el 18 de octubre de 1967 en Tonalá Chiapas, México. Ingeniero civil con maestría en Ciencias de la Computación en la universidad de Tulane, Lousiana, EEUU y doctorado en Ciencia de la Computación en la misma universidad.

[123] Libro clásico de la teoría de compiladores *Formal Languages and their Relation to Automata*, por Hopcroft, J, Ullman, J, Hill, M, en http://portal.acm.org/citation.cfm?id=1096945. Sábado, 5 de abril 2008. 10:00 horas.

[124] El cálculo lambda fue presentado en el capítulo anterior. este fue formulado por Alonzo Church con ayuda de C. Kleene, quienes describen funciones matemáticas.

introducirá el concepto de Oráculo en la máquina, brillante concepto que permite encapsular conceptos sobre la complejidad en relación a los dispositivos automáticos.

La máquina de Turing suele ser descrita, como si fuera un dispositivo físico, que no es posible fabricarla debido a la definición abstracta de sus componentes. La máquina está constituida por una unidad de control (que determina el estado en que se encuentra la máquina y actúa sobre la memoria), una memoria que es una cinta de longitud infinita en ambos sentidos (izquierda y derecha), está contiene celdas una a continuación de otra (cada celda puede contener un símbolo).

Hoy la máquina de Turing se representa mediante dispositivos mecánicos eléctricos, también suele ser mostrado en programas[125] de computadora, todos éstos ejemplos son solo de interés académico. En el transcurso del tiempo se han presentado diversas definiciones para la máquina manteniéndose el sentido original.

Una definición que consideramos adecuada la enuncia e Haugeland[126], que dice:

> Una máquina de Turing consta de dos partes: una cabeza y una cinta. La cinta es sólo un medio de almacenamiento pasivo: a lo largo está dividida en casillas, cada una de las cuales puede contener un elemento. … La cinta se usa también para las entradas y salidas, escribiendo elementos en ella antes que empiece la máquina, y leyendo los resultados después que para. La cabeza es la parte activa de la máquina que va, a saltitos, hacia atrás o hacia delante a lo largo de la cinta, casilla por casilla, y, mientras lee o escribe los elementos. En cada paso dado, la cabeza va "explorando"… También en cada paso, la propia cabeza se encuentra en estado interno…. este estado cambia de un paso al siguiente. … La cabeza entrará en un estado especial llamado "para", en cuyo caso la

[125] Ver http://www.microsiervos.com/archivo/ordenadores/premio-máquina-turing-si.html. Sábado, 5 de abril de 2008, 18:00 horas.

[126] John Haugeland (13 de marzo 1945, 23 de junio 2010) Profesor de filosofía de la Universidad de Chicago. Presidente del departamento de filosofía desde 2005 hasta 2007. Enseñó en la Universidad de Pittsburgh y la Universidad de California en Berkeley, y fue miembro del Palo Alto Research Center. Profesor visitante en la Universidad de Helsinki, Finlandia. Investigador de la Fundación Nacional para las Humanidades y del Centro de Estudios Avanzados en Ciencias de la Conducta.

máquina se detiene – dejando sus salidas en la cinta (Haugeland, 1999:127-129)

Otra definición es la de Penrose[127].

Una cinta infinita… prefiero pensar la cinta como la representación de un entorno por el cual puede moverse nuestro dispositivo finito. (Por supuesto, con la electrónica moderna ni la cinta ni el dispositivo tienen que moverse realmente en el sentido físico ordinario, pero tal idea de movimiento es una manera conveniente de representar las cosas. Desde este punto de vista, el dispositivo recibe todo su input desde el entorno; utiliza el entorno como el "papel", y al final escribe su output en este mismo entorno. … En la imagen de Turing la cinta consiste de una secuencia lineal de cuadros que se considera infinita en ambas direcciones. Cada cuadro de la cinta está en blanco o contiene una sola y única marca. El uso de cuadros marcados o sin marcar ilustra el hecho de que estamos admitiendo que nuestro entorno (es decir, la cinta) puede ser descompuesto y descrito en términos de elementos discretos (y no continuos). Esto es razonable si queremos que nuestro dispositivo funcione de un modo fiable y perfectamente definido. Estamos admitiendo que el entorno sea (potencialmente) infinito como consecuencia de la idealización matemática que estamos utilizando, pero en cualquier caso particular el input, el cálculo y el output deben ser siempre finitos. De este modo, aunque la cinta se considera infinitamente larga, en ella debe haber sólo un número finito de marcas reales. Más allá de un cierto punto en cada dirección la cinta debe estar completamente en blanco. (Penrose, 1989: 53-55)

La máquina de Turing tiene: una unidad de Control, una cinta de memoria infinita, una lectora grabadora que puede leer, borrar y grabar en cada celda, además la lectora puede desplazarse a la derecha o al izquierda de la celda de memoria en que este ubicada.

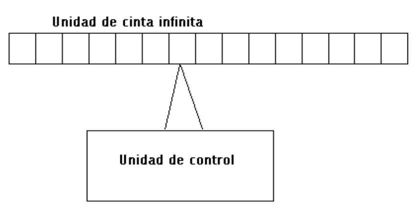

Figura 2. Componentes de la máquina de Turing

[127] Roger Penrose, (nace el 8 de agosto de 1931) es físico matemático nacido en Inglaterra y Profesor Emérito de Matemáticas en la Universidad de Oxford. Son famosas sus contribuciones a la relatividad general y la cosmología. También ha dedicado su tiempo a las matemáticas recreativas y a la filosofía. Es miembro de la Royal Society de Londres en 1972, ganó el Science Book Prize en 1990, y compartió el Premio Wolf en Física con Stephen Hawking en 1988. Fue nombrado caballero en 1994.

La máquina de Turing ejecuta un procedimiento de manera precisa. Está definida de forma que es posible verificar el resultado del funcionamiento mediante la utilización de lápiz y papel, con la necesaria actitud para no equivocarnos, explicado en las instrucciones del procedimiento.

Se deduce que Turing conceptúa la máquina como un procedimiento que puede ser ejecutado por un hombre, sin ninguna ambigüedad, garantizando que se hará lo que está especificado. Es una definición que está relacionada al operando de la mente humana, no decimos que corresponda a la totalidad de cómo opera la mente, pero hay parte de ésta en la especificación dada: "…Nosotros podemos comparar a un hombre en el proceso de cálculo de un número real a una máquina que sólo es capaz de un número finito de condiciones q_1, q_2,… q_R que se llama m-configuraciones" (Turing, 1936: 2)

En la definición de la máquina de Turing se especifica una unidad de memoria en forma de cinta, que contiene casilleros sucesivos, no tiene límites ni a la izquierda ni a la derecha, es infinita. ¿Por qué plantearse una unidad infinita, al haber casilleros que nunca serán accesibles? Tal vez con esta definición Turing buscó representar el concepto de una ejecución que nunca termina, por el solo hecho de intentar leer el casillero que se encuentra en la cinta tan lejos a la izquierda como a la derecha, también es posible concluir que hay espacio para almacenar la cadena de caracteres que se desee calcular.

La unidad de control tiene la funcionalidad de leer o grabar un carácter en un casillero, además se desplaza en la cinta en dirección a la derecha o a la izquierda, y siempre estará en un estado determinado. ¿Cómo ejecuta las instrucciones sin ningún significado semántico? Aquí nos encontramos con la respuesta, se encuentra en los estados, cada uno indica una situación determinada, y se puede pasar de un estado a otro en función del estado y del símbolo que se está leyendo, de esta manera se especifica qué hacer en la cinta (grabar o leer) y desplazarse según corresponda (izquierda, derecha o parar). Para Turing es posible establecer un cálculo en los números, así lo expresa:

> Se puede pensar que los argumentos que demuestran que los números reales no son enumerables también se podrían demostrar que el cálculo de números y secuencias no pueden ser enumerables. Podría ser, por ejemplo, que el límite de una secuencia de números calculables debe ser calculable. Esto es una clara verdad si la secuencia de números calculables está definida por algunas reglas. (Turing, 1936:17).

En cuanto a los números reales Turing nos dice que no es posible que todos sean numerables, indica que hay números decimales que sí son calculables, en alusión al resultado dado por Gödel en su teorema de la incompletitud (Turing, 1936:2) "Según mi definición, un número es computable si este es decimal y puede ser escrito por una máquina" (Turing, 1936:1)

Definimos la máquina de Turing como la 7 u-pla M, de siete componentes de la máquina, de forma tal que M sería $(Q, \Sigma, \Gamma, Qo, b, T, \delta)$, donde:

Q, es el conjunto de estados de la máquina.

Σ, es el conjunto de caracteres de entrada en cinta de memoria.

Γ, es el conjunto de caracteres de la cinta.

Q_o, es el estado inicial de la máquina.

b, es el símbolo en blanco o nulo, que indica que no es ningún carácter.

T, es el conjunto de estados finales.

δ, es la función de transición: $Q \times \Gamma \rightarrow Q \times \Gamma \times \{I, D, M\}$

La función δ nos dice que si estamos en un estado y leemos un símbolo de la cinta, se transforma en un nuevo estado, cambiando el símbolo de la cinta y movemos la ubicación de la unidad de control (izquierda, derecha o deteniéndose).

El momento en que Turing publica su trabajo en 1936, no existían máquinas electrónicas, se encontraba en plena discusión la propuesta de Hilbert con respecto a la decibilidad de las matemáticas producto de los resultados del teorema de incompletitud de Gödel. En medio de este contexto Turing presenta una forma de definir el algoritmo en correspondencia al procedimiento para ser ejecutado utilizando lápiz y papel, como modelo abstracto para obtener la solución a cálculos numéricos en su forma más simple.

La propuesta de Turing al formular una máquina abstracta que ejecuta instrucciones elementales, capaces de ser ejecutadas por un hombre de tal forma que siga el procedimiento, sin necesidad de razonar lo presentamos mediante un ejemplo: Sea una cadena de símbolos "OOOO" que deseamos cambiar los símbolos que ocupa el lugar par (de izquierda a derecha) por el símbolo "1". Para el caso la máquina de Turing obtendría como resultado la cadena de caracteres "O1O1", como condición inicial la unidad de control se encuentra frente a la celda que contiene el primer carácter "O" (está a la izquierda de la cadena), y el estado inicial Q_0.

Las reglas de transformación son como sigue:

1. $\delta(O, Q_0) \rightarrow (Q_1, O, D)$ Empieza y se desplaza una celda a la derecha

y pasa del estado Q_0 a Q_1.

2. $\delta(O, Q_1) \rightarrow (Q_2, 1, D)$ Si lee el símbolo “O” en la cinta y está en el estado Q_1, cambia a “1” el símbolo, se desplaza una celda a la derecha y pasa al estado Q_2.

3. $\delta(O, Q_2) \rightarrow (Q_1, O, D)$ Si lee el símbolo “O” en la cinta y está en el estado Q_2, se desplaza una celda a la derecha y pasa al estado Q_1

4. $\delta(b, Q_2) \rightarrow (Q_f, b, D)$ Si lee el símbolo b que indica blanco o nada y está en el estado Q_2 se detiene la máquina y el proceso ha terminado.

Siguiendo con el ejemplo, los componentes que definen la máquina de Turing son:

$Q = \{ Q_0, Q_1, Q_2, Q_f \}$ Conjunto de estados de la máquina.

$\Sigma = \{ O, b\}$ Conjunto de símbolos de inicio en la cinta.

$\Gamma = \{ O, 1, b \}$ Conjunto de símbolos de la cinta.

Q_0, Estado inicial

b es el símbolo blanco.

$T = \{ Q_f \}$ Conjunto de estado finales, en este caso solo hay un estado.

δ, es la función de transformación de la máquina.

Para el caso de la cadena de caracteres “OOOO” según la definición que acabamos de construir de máquina de Turing, su transformación seria:

$OOOO \rightarrow Q_0OOOO \rightarrow OQ_1OOO \rightarrow O1Q_2OO \rightarrow O1OQ_1O \rightarrow O1O1Q_2 \rightarrow O1O1Q_f \rightarrow O1O1$

Cuando se presenta el estado Q_f (estado final), la máquina de Turing se detiene, e indica que se ha obtenido la solución. Si la máquina no se detiene, entonces decimos que se está en un caso no resoluble.

Turing adiciona una nota a su primer trabajo de 1936, trata sobre el documento publicado por Church unos días antes, acerca de la idea de cálculo efectivo, manifestando que los resultados mostrados son equivalentes, pero que son definiciones diferentes. Esta similitud se entiende en el contexto de la solución del *Entscheidungsproblem*, es decir en relación a la búsqueda de un método efectivo que se encuentra expresado en un algoritmo para resolver las ecuaciones diofánticas.

> En un reciente documento de Alonzo Church ha introducido un concepto de "cálculo efectivo", lo que equivale a mí "computabilidad", pero es definida muy diferente. Church también llega a conclusiones similares acerca de la *Entscheidungsproblem*. La prueba de equivalencia entre "computabilidad" y "calculo efectivo" se expone en un anexo al presente documento. (Turing, 1936:2).

En el apéndice titulado *Computabilidad y Cálculo Efectivo,* Turing utiliza el concepto de fórmula bien formada, mediante el cálculo lambda, tal como lo entienden Kleene y Church. “Para demostrar que toda secuencia computable γ es λ-definible, debemos mostrar cómo encontrar una fórmula M”. (Turing, 1936:35)

De otro lado, la máquina de Turing contiene definiciones que permitieron elucidar el desarrollo de las computadoras, así tenemos la opinión de Claude Shannon[128] quien

[128] Claude Elwood Shannon (nace el 30 de abril de 1916 y fallece el 24 de febrero del 2001). Ingeniero electricista y matemático estadounidense conocido por sus estudios de la información. Es conocida su publicación de 1948 sobre *A Mathematical Theory of Communications*. En 1950, escribe sobre la programación de la computadora para jugar ajedrez.

publica en 1956, su artículo *A Universal Turing Machine With Two Internal State* en la que hace una demostración de que en una máquina de Turing se puede reducir el número de estados hasta llegar al mínimo de dos y que los estados eliminados pueden ser sustituidos por símbolos que estarían en la unidad de cinta (en la memoria). Esta demostración matemática relaciona las reglas que tiene la máquina con los espacios de memoria en la cinta y sus contenidos. Al final de la demostración plantea la interrogante si se puede simplificar la cantidad de símbolos en la cinta por estados de la máquina.

Resaltamos la demostración realizada por Shannon porque relaciona los estados de la máquina de Turing con símbolos que pueden estar en la cinta. Obtenemos la conclusión de que en un programa, las instrucciones pueden ser reemplazadas por símbolos en la memoria, es decir, que existe una correspondencia entre las instrucciones del programa y los datos que están en la memoria (unidad de cinta).

Von Neumann, en su libro publicado póstumamente *The Computer and The Brain*, en 1958, reflexiona como metáfora sobre la similitud de la computadora con el funcionamiento del cerebro. En el caso de las computadoras refiere a los códigos y su rol en el control del funcionamiento de una computadora. Precisa que estos códigos adquieren la forma de ser completos, en el sentido que permiten el funcionamiento de la máquina e indica que la máquina de Turing se refiere al concepto de código corto, como código elemental de la máquina.

> El lógico ingles A. M Turing, mostró en 1937… que es posible desarrollar sistemas de códigos de instrucciones para una máquina computadora… Entonces el sistema de instrucciones que hizo una imitación de máquina como el comportamiento de otra es conocido como código corto (von Neumann, 1958:71)

La observación que hace von Neumann sobre la máquina de Turing es que esta captura la esencia del código corto que hace que la máquina se pueda comportar como si fuera otra máquina.

3.2. Máquina de Turing y el Oráculo.

Turing en su tesis doctoral en 1938, en la universidad de Princeton, la que fuera supervisada por Church, introduce la idea de un mecanismo del tipo "caja negra", al que denomina "Oráculo", éste dispositivo se añade a la definición de máquina. El mencionado mecanismo tiene la capacidad de determinar si un proceso terminará el cálculo que está por ejecutar sin haberlo realizado. El Oráculo resuelve el problema de que la máquina no pueda detenerse, como consecuencia de una ejecución al infinito, dado que no encuentra la solución, conocido como el *problema de la parada.*

El trabajo de Turing fue titulado *Systems of logic based on ordinal*, documento clásico en la teoría de la recursividad. Con el dispositivo "Oráculo", su máquina se convierte en una máquina-O, que es un dispositivo abstracto que identifica cuándo una tarea es o no es computable. En el documento no explica cómo sería el dispositivo. Conociendo el interés del trabajo realizado por Church (las funciones recursivas), debe haber sido el propósito de Turing el resaltar las características de la recursividad y aislar los problemas que tiene la máquina de Turing con respecto a lo que no puede calcular. Es el caso que utiliza el concepto de formula bien formada como el cálculo lambda para referenciar a los números: "$\lambda f\ [\lambda x[\{f\}(\{f\}(x))]]$ representa el entero positivo 2, más adelante nos permitiremos representar ordinales" (Turing, 1938:164).

Turing indica que las máquinas con Oráculo (máquinas-O) pueden ser descritas por tablas del mismo tipo de las que usó para su máquina a la que denominó máquina-A (definida en 1936), de manera que para la definición de la máquina-O le asigna números a las configuraciones internas, que son estados en el mismo sentido al de la máquina-A.

> Con la ayuda del Oráculo podremos formar un nuevo tipo de máquina (llamada o-máquina), teniendo en éste uno de sus procesos fundamentales que resuelve un determinado problema teórico numérico. Más aún estas máquinas son el comportamiento en esta vía. Los movimientos de la máquina son determinados como usuales por una tabla, excepto en el caso de movimientos desde una cierta configuración interna **o.** Si la máquina está en la configuración **o** y si la secuencia de símbolos marcados con ***l*** son la fórmula bien formada **A**, entonces la máquina va de la configuración interna **p** o **t** según como ésta es o no verdad que **A**. (Turing, 1938:173).

Penrose trata el tema de la máquina-O en el capítulo de la *Teoría cuántica y el cerebro* en su libro *Sombras de la Mente*, como argumento de una nueva máquina basada en nuevos principios físicos en la arquitectura de las computadoras, así nos dice:

> Turing introdujo un importante concepto de relevancia para esta cuestión, al que denomino oráculo. La idea de oráculo consistía en algo (presumiblemente algo ficticio, en su mente que no tendría que ser físicamente construible) que podía resolver de hecho el problema de la detención. (Penrose, 1994: 401).

En la sustentación de la tesis doctoral de Turing, demuestra que es lo que calcula una máquina de Turing en relación a las funciones recursivas, basada en números ordinales. Ésta contribuye a la formulación de la Tesis de Church-Turing que dice: Si una función de enteros positivos es calculable si es recursiva, implica que ninguna máquina puede realizar un procesamiento que se encuentre más allá del alcance de una máquina de Turing. Todo cálculo que puede ser ejecutado en la máquina es expresable en funciones recursivas.

El Oráculo en la máquina de Turing define una máquina-O, que contiene una potencia de cálculo mayor (puede determinar que es computable o no), suele ser interpretada como una máquina con capacidades relacionadas al cómputo de la inteligencia del hombre, al respecto Penrose nos dice sobre la posibilidad de que las máquinas puedan desarrollar inteligencia artificial, para esto fija cuatro orientaciones con respecto a la posibilidad de asociar el pensamiento humano con el acto de computar:

> Creo que hay al menos cuatro puntos de vista…… A) Todo pensamiento es computación; en particular, las sensaciones del conocimiento consiente son provocadas simplemente por la ejecución de computaciones apropiadas. B) El conocimiento es un aspecto de la acción física del cerebro; y si bien cualquier acción física puede ser simulada computacionalmente, la simulación computacional no puede por sí misma provocar conocimiento. C) La acción física apropiada del cerebro provoca conocimiento, pero esta acción física nunca puede ser simulada adecuadamente de forma computacional. D) El conocimiento no puede explicarse en términos físicos, computacionales o cualesquiera otros términos científicos. (Penrose, 1994: 26).

Según lo expresado, Penrose clasifica las creencias sobre la posibilidad de inteligencia en las máquinas, lo hace desde uno optimista hasta llegar a un criterio pesimista, siendo estas:

(A) Creencia fuerte de Inteligencia Artificial.

(B) Creencia débil de la Inteligencia Artificial.

(C) Creencia de que es posible mediante una nueva física que está en desarrollo.

(D) Creencia que no es posible la Inteligencia Artificial.

La opción (A) corresponde a la capacidad de cómputo en las máquinas-A, en la opción (B) expresa la capacidad de cómputo en el sentido de que las máquinas de Turing no pueden desarrollar conocimiento, en todo caso el Oráculo es un mecanismo que puede ser interpretado como que el hombre es quien escribe el programa y ejecuta la máquina, o en su defecto la existencia de condiciones iniciales que puede tener un programa (proceso de consistencia)[129] que imposibilita la ejecución hacia al infinito, la opción (C) entiende que la máquina con capacidad para lograr inteligencia artificial es la máquina-O y ésta no puede ser lograda con la física actual, en la opción (D) refiere a que la inteligencia artificial no es posible con una máquina-A, ni una máquina-O.

Una forma de clasificar las máquinas, está en relación a la simplificación del esfuerzo del hombre para operar o controlar la ejecución, como el operar una herramienta, tomando en cuenta a la máquina de Turing sin Oráculo (máquina-A), como máquina que ejecuta siguiendo instrucciones. Encontramos en el libro de Piscoya[130], que trata sobre la energía de mando como la capacidad para dirigir la máquina y energía de ejecución que es el cómo ejecuta, estos conceptos son también mencionados por Pierre de Latil[131], en relación a la capacidad para realizar el trabajo con máquinas, según estos conceptos se establece una clasificación.

> Una máquina es de tipo 1 cuando la energía de mando está integrada a la energía de ejecución, es el caso de la palanca, el arado, o la bicicleta, entre otros. En estas máquinas el hombre tiene que hacer un esfuerzo sostenido para lograr que realicen el trabajo...... Una máquina es de tipo 2 cuando la energía de mando es independiente de la energía de ejecución que puede ser, según sea el caso, la fuerza hidráulica, eólica o la de motor. Es el caso del molino tradicional, de un tractor, de una tejedora o de una impresora. Estas máquinas requieren de un operador... manipular palancas o botones... Una máquina es de tipo 3 cuando su energía de mando ha sido reducida a la introducción de un conjunto fijo de instrucciones, expresadas en un lenguaje L, que la máquina reconoce y acepta, realizando actos que, a su vez son ordenes.... Esta máquina se conoce como ordenador o computador. (Piscoya, 2009: 264).

[129] Son procesos de computación que verifican las condiciones iniciales del programa. Son adecuados para realizar una operación. Ejemplo si se divide dos números a/b, se verifica que el número b no sea cero. De lo contrario no habría solución.
[130] Luis Piscoya, profesor de la Unidad de Post Grado de Filosofía en la Universidad Nacional Mayor de San Marcos. Refiere en su texto sobre *Tópicos en Epistemología*.
[131] Pierre de Latil, francés investigador, estudioso y divulgador de la cibernética con varias publicaciones y traducciones al castellano. Trata el tema de las máquinas en cuanto a espera de mando y de ejecución en su libro *El Pensamiento Artificial: Introducción a la Cibernética*, 1958.

La clasificación mencionada por Piscoya, corresponde a tres tipos de máquinas y entendemos que en un sentido amplio ubica a la computadora en el tipo 3. Nos permite reflexionar, en el aspecto de cómo y para qué, el hombre construye máquinas, con el objetivo de reducir la energía de ejecución, incluso la energía de mando dado que se encontraría en la máquina. El hecho de introducir instrucciones a las máquinas y que las ejecute, permite sugerir la posibilidad de que las máquinas del tipo 2 pueden ser transformadas en máquinas de tipo 3 (en la medida de que la tecnología lo haga posible).

Piscoya presenta las máquinas de tipo 3 en correspondencia a una máquina de Turing sin Oráculo, en el sentido de que la computadora es la que daría los mandatos para la ejecución de las acciones mecánicas. Encontramos lo expuesto dentro de la línea de estudio de la cibernética, en la que Pierre de Latil fuera un excelente investigador y difusor de la cibernética, en relación a computadoras acopladas a herramientas, ampliando el criterio de interpretación hacia campos no necesariamente de cálculos numéricos.

3.3. Máquina de Turing que aprende.

Turing argumenta sobre lo que entiende por máquina "inteligente" y lo manifiesta en el documento *Intelligent Machinery*, escrito en los tiempos en que laboraba en el *National Physical Laboratory* de Londres, en 1948, donde hace una crítica a las ideas que considera que atacan el concepto que él manejaba de "inteligencia" en las máquinas. Su documento presenta una crítica a cinco inclinaciones, que las rechaza por considerarlas superadas, así las enuncia:

(Objeción a) Una indisposición para admitir la posibilidad de lo que se pueda hacer, es que el

> hombre pueda tener un rival con poder intelectual… (Objeción b) Una creencia religiosa en la que cualquier idea a construir una máquina es una clase de promesa irreverente… (Objeción c) El límite máximo de caracteres de máquina que se pueden usar, según tiempos recientes (hasta 1940). Estimuló a creer que la máquina es limitada ante las exigencias extremas …(Objeción d) El teorema de Gödel y sus resultados relacionados (Gödel, Church, Turing) muestran que si uno usa las máquinas para determinar la verdad o falsedad de teoremas matemáticos, no está dispuesta a tolerar un inesperado resultado errado, entonces cualquier máquina estará en el mismo caso de no proporcionar una respuesta a todo. De otro lado la inteligencia humana parece ser capaz a encontrar métodos e incrementar el poder de tratamiento con tales problemas. (Objeción e) Hasta donde una máquina pueda mostrar inteligencia y es contemplada de que si es posible, será como reflejo de la inteligencia de su creador (Turing, 1948: 1).

Turing refuta cada una de las objeciones, afirmando que es posible la “inteligencia” en las máquinas, sosteniendo desde una perspectiva matemática, que las máquinas abstractas en relación al concepto de algoritmo, contienen un procedimiento que consiste de instrucciones que permiten operar lógicamente dispositivos de la máquina.

Turing afirma que un procedimiento puede ser cambiado, siendo un concepto temprano sobre la modificación de la secuencia de ejecución de las instrucciones del algoritmo, indicando que la máquina “aprende” para no repetir una situación no deseada, esto se logra modificando la secuencia de instrucciones en el programa, para este fin el programa debe tener elementos almacenados en la memoria de la máquina, de forma tal que puedan ser modificados en el momento de que se está ejecutando.

Sobre las objeciones (a) y (b) nos dice que son de naturaleza emocional, por lo tanto no requiere mayor sustentación en su contra. Aquí marca distancia con los conceptos a los que considera que tienen contenido del tipo psicológico o religioso.

En la objeción (c), Turing la descarta con el ejemplo de la máquina ENIAC, indicando sobre sus capacidades de operación en cuanto a velocidad y almacenamiento, el argumento se hace más fuerte si consideramos las capacidades actuales de las computadoras. En la objeción (d), referida a los argumentos del teorema de Gödel y que la máquina no debería cometer errores, afirma como criterio de refutación y además como una línea de acción del pensamiento del diseño de las computadoras y programas para estas máquinas, el que errar no es un requisito para negar la posibilidad de “inteligencia”. Sostiene como ejemplo del aprendizaje, que este se basa en el error, como un elemento importante para incorporar experiencia y asimilarla al proceso cognitivo asociado a la inteligencia.

Finalmente en la objeción (e) sostiene que esta se contradice con el siguiente ejemplo: Si un profesor ayuda a sus alumnos con métodos de enseñanza y luego abandona la comunicación con el pupilo, siendo el pupilo quien presenta los resultados y no el profesor, la decisión le corresponde al pupilo y no al profesor, en clara alusión de que es la máquina la que presenta el resultado y no el creador del programa.

No tratamos las primeras objeciones (a) y (b), que fueron calificadas como débiles porque consideramos que se refieren al significado de inteligencia, tal como se trata en la definición que se daría a la de inteligencia en el hombre, aclaramos que Turing trata de un concepto de “inteligencia” que refiere a la ejecución de procedimientos en forma automática. En la objeción (c), las máquinas de hoy en día tienen más potencia y capacidad de las que tenía la computadora ENIAC (citada por Turing), así que en este caso, sólo cabe decir que se espera tener máquinas con más capacidades. En lo que refiere a la objeción (d), hoy se tiene experiencias exitosas de aplicación de la inteligencia artificial en determinados campos, ha posibilitado argumentar que la incompletitud no es un problema para desarrollar programas de computadoras que "simulan" el raciocinio del hombre en temas específicos, además como lo indica Turing hay que incorporar el error como parte del proceso, lo asimila, como parte del aprendizaje, como mecanismo para que no vuelva ocurrir el error.

Siguiendo con el documento del año 1948, *Intelligent Machinery,* define una máquina abstracta, a la que denomina máquina desorganizada, en la que sus estados se encuentran relacionados, con la característica de que las reglas de incidencia son definidas como funciones y representadas mediante una matriz, de forma que el programa que contiene reglas resulta ser en cierto sentido la mencionada matriz, de manera que si se altera los valores de la matriz, se está modificando las relaciones entre los estados, y por lo tanto se cambia la secuencia de las instrucciones de la máquina. Esta definición de cambiar la relación entre los estados, es una modificación a la máquina de Turing. Sabemos que en la máquina-A, la función de transformación es fija, es decir sólo puede ser ejecutada tal como está definida desde el inicio del proceso; esta secuencia no puede ser alterada; mientras en la máquina desorganizada es posible modificar la relación de los estados que existían al inicio del proceso.

Turing nos muestra un ejemplo teórico sobre la máquina desorganizada, con un autómata que está definido mediante pulsos eléctricos que se encuentran sincronizados y que determina sus estados según los valores en un instante dado. Indica que se hacen muy complejos en la medida que aumentan las capacidades de la máquinas, en el ejemplo, nos presenta una relación de cinco estados y refiere que ésta es una idealización, porque en la práctica las computadoras están definidas con múltiples estados.

La máquina desorganizada presentada por Turing está constituida por estados, y las relaciona mediante salidas: i (r) y j (r), que indican la salida hacia el estado r. Así nos presenta el siguiente el diagrama.

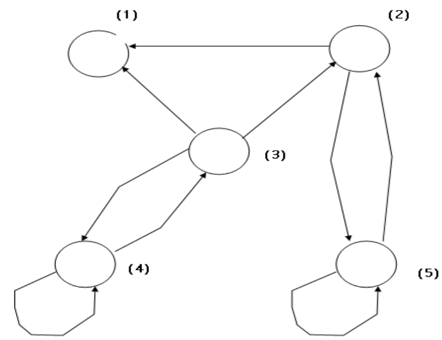

Figura 3. Máquina desorganizada (Turing, 1948: 6)

Las conexiones corresponden a las relaciones de incidencia, expresadas en las siguientes funciones:

r (estado)	i(r)	j(r)
1	3	2
2	3	5
3	4	5
4	3	4
5	2	5

Figura 4. Relaciones en la máquina desorganizada (Turing, 1948: 6)

Turing revisa la variedad de mecanismos y puntualiza sobre el significado del significado de discreto (secuencia de instrucciones del tipo paso a paso) y del continuo como un flujo (cuando es un proceso que no tiene interrupción) como ocurre en el flujo de la electricidad. Al respecto dice:

> Nosotros podemos denominar máquina discreta cuando ésta es natural para describir sus posibles estados como un conjunto discreto, la noción de la máquina se obtiene por saltos de un estado hacia otro. El estado de una máquina continua forma en otro sentido una multiciplicidad y en el comportamiento de la máquina es descrita por una curva de sus varias formas (Turing, 1948:3)

Turing incorpora adicionalmente los conceptos de *control* y *active* (controlado y activo). El de controlando corresponde a que proporciona información mientras el concepto de activo es de sentido contrario, no proporciona información, así nos da algunos ejemplos de máquinas:

Un tractor es continuo y activo.

Un teléfono es continuo y controlado (proporciona información).

Una calculadora mecánica es discreta y controlada.

Un cerebro es probablemente continuo y controlada, pero es muy similar a una mecanismo discreto (proporciona información).

La ENIAC, ACE, etc. son discretos y controlados (proporcionan información).

Un analizador diferencial es continuo y controlado (proporciona información).

Cuando refiere al ejemplo de la máquina de computación lógica, describe a la máquina-A, que es la definición que se remonta al año 1936.

> … un cierto tipo de máquina discreta fue descrita, ésta tenía una capacidad infinita de memoria obtenida en la forma de una infinita cinta marcada en cuadrados en el que cada símbolo podrá estar impreso. En un momento determinado hay un símbolo en la máquina, éste es llamado el símbolo leído (Turing, 1948:3)

Turing menciona su máquina como mecanismo inteligente, con una estructura lógica (en relación a un sentido físico), que realiza lo que puede ejecutar un hombre mediante lápiz y papel. Define que las máquinas de Turing no tienen límites en la duración de la ejecución y tampoco en la capacidad de la memoria de la máquina.

> Estas máquinas que aquí llamamos máquinas de computación lógica, son de principal interés cuando nosotros consideramos en principio que una máquina puede estar designada a hacer, permitiendo lo ilimitado del tiempo y lo ilimitado en la capacidad de almacenamiento (Turing, 1948:4)

Turing afirma que es posible hablar de una máquina que "aprende", porque puede cambiar la relación de sus estados. En contraposición a estas ideas, encontramos lo expuesto por Penrose, que considera posible obtener una máquina "inteligente" si se desarrolla una nueva[132] física. Así resulta que lo manifestado por Turing en 1948 es un concepto temprano de lo que sería expresado en la inteligencia artificial, y que sería un tema controversial.

Hoy el concepto de estados en las máquinas y que puedan ser autos modificados en la ejecución de un procedimiento, tiene diversas aplicaciones, específicamente en los algoritmos genéticos, que son programas que modifican su comportamiento inicial. Es una lástima que el documento haya sido divulgado en 1964, dieciséis años más tarde. Hoy se estudia lo propuesto por Turing en las redes neurales en la computación[133].

3.4. Teoría de autómatas y lenguajes.

Incluimos los autómatas y su relación con los lenguajes de programación de computadora, por la exitosa utilización de la máquina de Turing en temas sobre la

---

[132] Penrose, en su libro *Las Sombras de la Mente,* da su punto de vista sobre la posibilidad de inteligencia artificial: "El punto de vista C es el que yo personalmente creo que se acerca más a la verdad", está en relación al desarrollo de una mecánica cuántica, de forma tal que permita la construcción de un computador cuántico con un procesador que se acerque a la forma de cómo funciona un computador.

[133] Carlos Coello Coello lo menciona en su libro *Breve historia de la computación y sus pioneros*, página 117.

gramática de los lenguajes de programación, y códigos de máquinas estudiados en la teoría de lenguajes en la ciencia de la computación. Usualmente el término autómata se asocia a una máquina que ejecuta tareas automáticas con velocidad y precisión. Estas máquinas son definiciones matemáticas que capturan el significado de secuencia de instrucciones que operan dispositivos. Son máquinas abstractas y simples, utilizadas en reconocer lenguajes regulares, estudiado en el campo de la ciencia de la computación[134].

El lenguaje regular es definido por una gramática, que contiene las reglas de la sintaxis de las instrucciones. Se definen diferentes tipos de gramática, éstas suelen ser clasificadas según el autómata que relaciona el algoritmo que reconoce las instrucciones. Cabe destacar que en la teoría de los lenguajes en la ciencia de la computación, los autómatas corresponden a un tipo de gramática, y su importancia es el algoritmo, como programa que será ejecutado en la computadora para el reconocimiento sintáctico y semántico[135].

> Un autómata es un modelo matemático simple, fundamental y ubicuo en la Informática, útil para explicar el comportamiento de un mecanismo. También se habla de máquina de estados, porque la idea fundamental es describir el estado de una máquina y entender su dinámica mediante la explicación de las posibles transiciones entre estados que puede seguir. Por ejemplo, el comportamiento de un computador digital puede modelarse con un autómata. También el proceso de la ejecución un programa. De hecho, en los principios de la programación, la semántica de los programas era explicada con diagramas de flujo, una notación operacional para explicar un autómata que los ejecutaba. (Cardoso, 2007:1).

Los lenguajes comparten propiedades de forma expresadas en las reglas de la

[134] Libro clásico de la teoría de compiladores. Formal languages and their relation to automata, por Hopcroft, J, Ullman, J, Hill, M, en http://portal.acm.org/citation.cfm?id=1096945. Sábado, 5 de abril 2008. 10:00 horas.

[135] Se entiende que el reconocimiento que se hace en la computadora de un programa, que convertirá las instrucciones en un lenguaje original al lenguaje de la máquina. Este pasa por un primer paso de reconocimiento valido en la escritura de la instrucción (análisis sintáctico) y luego el convertir al código de máquina para su ejecución(análisis semántico)

gramática, suelen ser definidos según definición proporcionada por Noam Chomsky[136], quien propuso una jerarquía de clasificación. Estas definen los lenguajes desde un criterio de formación de las reglas hacia las menos simples. Chomsky utiliza el término de gramática generativa en el sentido que ésta produce oraciones que son parte del lenguaje, que en nuestro caso las hemos denominado instrucciones.

> … llamo gramática generativa a un sistema de reglas de manera explícita y bien-definida asigna descripciones estructurales a las oraciones. Es obvio que cada hablante de una lengua ha llegado a interiorizar y dominar una gramática generativa que expresa su conocimiento de su lengua … Cuando decimos que una gramática genera una oración con cierta descripción estructural, queremos decir simplemente que la gramática asigna esta descripción estructural a la oración. (Chomsky, 1965:10)

La forma como se definen las instrucciones, es representada en una jerarquía, que se inicia en el símbolo distinguido S, y desde ésta, según meta definiciones llega a cada una de las palabras de la instrucción. Chomsky lo explica de la siguiente manera:

(3)

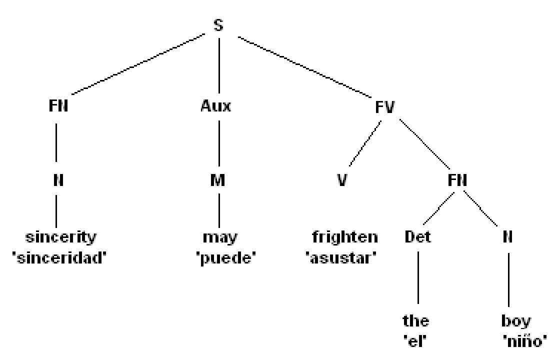

Figura 5. Árbol Sintáctico de una oración[137].

> Una gramática que genera Ahormantes simples como el de (3) puede basarse en un vocabulario de símbolos que incluya tanto formantes (the, boy, etc.) como símbolos categoriales (S, FN, V, etc.).

[136] La jerarquía de Chomsky es una clasificación jerárquica de distintos tipos de gramáticas formales. Esta jerarquía fue presentada en su libro *Aspect of the Theory of Syntax*, en 1965, fue una ampliación de su libro *The Logical Structure of Linguistic Theory* de 1955.

[137] Tomado del libro de Noam Chomsky de 1965.

> Además cabe dividir los formantes en elementos léxicos (sincerity, bob) y elementos gramaticales (Chomsky, 1965:63).

Los lenguajes según la jerarquía de Chomsky son: Lenguajes regulares, que son lenguajes simples, como el conjunto de los números binarios. Los lenguajes libres de contexto son del tipo que usan reglas de los paréntesis[138], incluyen a los lenguajes regulares, son como las expresiones aritméticas. Los lenguajes recursivamente enumerables, incluyen a los lenguajes libres de contexto, son lenguajes que contiene instrucciones de la forma "si, entonces, caso contrario", refiere a que después de la palabra "si" debe continuar una expresión a ser evaluada, tal que, si se cumple lo evaluado hay una expresión a evaluar que le corresponde a la palabra "entonces", y si no cumple la condición hay una segunda expresión que le corresponde a la palabra "caso contrario". En una instrucción de este tipo puede contener otra del mismo tipo, anidando expresiones (unas dentro de otras).

Las reglas definidas por Chomsky son una notación formal que define la gramática de los lenguajes de programación de computadoras, así tenemos que para cada tipo de gramática le corresponde un tipo de autómata que en esencia es una máquina de Turing. Resulta adecuada la definición de reglas de la gramática en la que según su forma definen el tipo de la misma, explicando al respecto Acero, Bustos y Quesada:

> Las reglas… reciben el nombre de reglas libres de contexto o de contexto libre o no contextual. …. En la literatura lingüística se mencionan a veces las reglas sensibles al contexto (o contextuales), es decir reglas que nos dicen como reescribir un <elemento> o <constituyente> en un contexto… Otro tipo de reglas, llamadas a veces de estados finitos son aún más simples (Acero J, Bustos E, Quesada D, 1989:54)

[138] Reglas de los paréntesis, consiste en la sintaxis que tienen los paréntesis en las expresiones aritméticas, en el sentido que si hay un paréntesis izquierdo "(", debe seguirle un paréntesis derecho ")". También debe considerar que si se apertura un paréntesis izquierdo, su correspondiente paréntesis derecho debe estar antes del paréntesis derecho del correspondiente paréntesis izquierdo que fuera abierto.

Los lenguajes son definidos mediante las gramáticas que se expresan mediante la notación de la 4-upla: (Vn, Vt, P, S), donde:

Vn es el vocabulario de los signos auxiliares de la gramática,

Vt es el vocabulario del lenguaje,

P son las reglas de la gramática o producciones[139] y

S es el símbolo distinguido de la gramática.

Los lenguajes son clasificados por la forma que tienen las producciones, siendo estos: Lenguaje regular, libre de contexto y lenguaje sensitivo al contexto

Las producciones de la gramática regular tienen la siguiente forma:

$X \rightarrow aY$

$X \rightarrow a$ Donde $X, Y \in Vn$, $a \in Vn$

Ejemplo de un lenguaje regular: $L = \{a^n b^m c^p / n, m, p > 0\}$

El autómata regular es una máquina de Turing, con la restricción de que sólo puede leer la cinta en un solo sentido (derecha a izquierda o de izquierda a derecha).

[139] Noam Chomsky utiliza el concepto las Producciones; éstas fueron definidas por Emil L. Post, matemático estadounidense, que utilizo un concepto de máquina bastante similar al definido por Turing. Post describe funciones que las relaciona una con otras, llamándolas producciones. Chomsky utiliza éstas para definir el concepto de gramática, mayor referencia en Martin Davis, 1987.

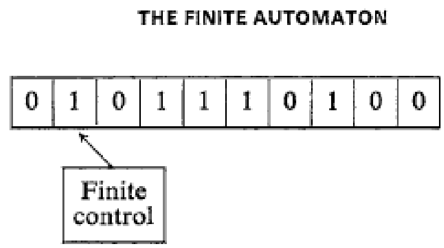

Figura 6. Gráfico obtenido del libro de Hopcroft, J, Ullman, J, Hill.

El lenguaje libre de contexto está definido por la gramática: (Vn, Vt, P, S), en la que las reglas de producción en P tienen la siguiente forma:

$X \rightarrow \alpha$ Donde $X \in Vn$, $\alpha \in (Vn \cup Vt)^*$

α: Cadena de símbolos de Vn o Vt

Ejemplo de un lenguaje libre de contexto: $L = \{a^n b^n / n > 0'\}$

El autómata que reconoce las instrucciones del lenguaje libre de contexto, tiene la restricción que lee la cinta en un sentido (derecha a izquierda o izquierda a derecha), adicionalmente tiene una unidad de memoria tipo pila.

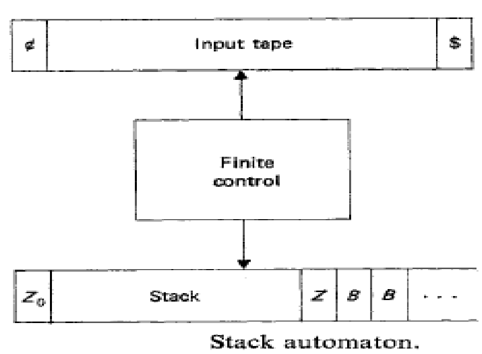

Figura 7. Gráfico obtenido del libro de Hopcroft, J, Ullman, J, Hill.

El lenguaje sensitivo al contexto, es el definido por la gramática: (Vn, Vt, P, S), donde las reglas de la producción tienen la siguiente forma:

$\alpha X\beta \rightarrow \alpha Y\beta$

$X \rightarrow \alpha$ Donde $X,Y \in Vn$, $\alpha, \beta \in (Vn \cup Vt)^*$

Son cadenas de símbolos V_n y V_t

Ejemplo de un lenguaje sensitivo al contexto: $L = \{a^n b^n c^n / n > 0^{/}\}$

El autómata que reconoce las instrucciones del lenguaje del tipo sensitivo al contexto es similar al que reconoce las instrucciones del lenguaje de contexto libre, con la diferencia que no hay restricción en la dirección de la lectura de la cinta.

Los lenguajes del tipo mencionado son conocidos como regulares, que resultan ser la base de la teoría para la construcción de los compiladores[140]. El autómata está directamente relacionado al algoritmo. Los autómatas son derivaciones de la máquina de Turing.

La ciencia de la computación captura como propia lo expresado porque formaliza la teoría de los compiladores, que refiere a los traductores de las instrucciones, que las convierten en mandatos que ejecutan las máquinas.

---

[140] Compilador es el programa de computadora que traduce el programa escrito en lenguaje fuente a un programa escrito en el lenguaje de la computadora, tal como será ejecutado por el computador.

La teoría de los lenguajes de computación encuentra en la semiótica un campo fértil para su profundización y formalización, así tenemos la opinión de Umberto Eco[141], en la que refiere a los signos como medio expresivo.

> En términos académicos no considero la semiótica como una disciplina, ni aun como una división, sino quizás como una escuela, como una red interdisciplinaria, que estudia los seres humanos tanto como ellos producen signos, y no únicamente los verbales. Hay también, una semiótica de las luces de tráfico. La diferencia entre un lenguaje como el inglés y el sistema de luces de tráfico es que el último es más simple que el primero. Entonces, hay una aproximación general a la totalidad de la conducta semiótica, y yo llamo a este estudio la semiótica general. (Umberto Eco, 1993)

Siguiendo la opinión de Umberto Eco, en referencia al concepto semiótico de los lenguajes resulta pertinente la alusión a Peirce.

> Peirce fue redescubierto principalmente en la segunda mitad del siglo, y es su aspecto semiótico el que fascinó a los europeos (a propósito, ese aspecto fue el menos considerado entre los pocos felices que estudiaron a Peirce en Estados Unidos hasta hace poco). Peirce fue estudiado porque el enfoque estructuralista semiótico había privilegiado el modelo lingüístico, y Peirce fue consciente de la enorme variedad de signos que nosotros producimos y usamos. (Umberto Eco, 1993)

Señalamos que en el campo de los lenguajes y sus significados, aún está abierta la investigación, afirmamos que convergen diferentes disciplinas del conocimiento, encontrándonos de acuerdo con lo que afirman Acero, Bustos y Quesada.

> Los problemas terminológicos derivan también, en parte, de la concentración de las diversas tendencias en áreas distintas de la investigación, pese a que coinciden parcialmente. Así, los lógicos y los filósofos del lenguaje han tenido que ocuparse principalmente de los aspectos composicionales del significado; es decir, han tendido a ocuparse sobre todo del problema de cómo los significados de las expresiones se componen de los significados de sus partes. Los lingüistas a su vez, se han concentrado mayoritariamente en el estudio del significado de las unidades (Acero J, Bustos E, Quesada D, 1989:42)

---

[141] Umberto Eco (Alessandria, Piamonte; 5 de enero de 1932) escritor y filósofo italiano, experto en semiótica, Distinguido crítico literario, semiólogo y comunicólogo.

.

Peirce considera la importancia del lenguaje en las expresiones de naturaleza científica, que en nuestro caso, los hemos asociado en cierto sentido a la semántica en los lenguajes formales. En su artículo sobre la *Ética de la Terminología*, hace una crítica a los mundos científico y filosófico, porque están plagados de personas empeñadas en ejercer una magistratura sobre los pensamientos y símbolos, aconseja en resistir en relación con el uso de términos y anotaciones. Al respecto dice:

> … Al mismo tiempo, se necesita un acuerdo general tratándose del uso de términos y anotaciones – no demasiado rígido – con la mayoría de los colegas acerca del mayor número de símbolos, a tal punto que haya un pequeño número de sistemas diferentes de expresión que se deben dominar… Cada símbolo es, en su origen, o una imagen de la idea significada o el recuerdo de algún acontecimiento individual, de persona o cosa, conectada con su significado, o es una metáfora… (Peirce en Sercovich, 1987: 244)

Mencionamos a Peirce, porque nos presenta a la lógica como una forma de semiótica, siguiendo este sentido lo relacionamos con Turing en tanto precisamos que su máquina es un mecanismo lógico: "la lógica, en un sentido general, es solo otro nombre de la semiótica, la doctrina cuasi necesaria o formal de los signos" (Peirce en Sercovich, 1987: 244)

Peirce define el signo como que interviene en el proceso de interpretación. Es también el signo creado desde el representamen que es el signo que representa algo, así nos dice:

> Un signo o representamen es algo que representa algo para alguien en algún aspecto o carácter. Se dirige a alguien, es decir, crea en la mente de esa persona un signo equivalente o, quizás aún, más desarrollado. A este signo creado, yo lo llamo el Interpretante del primer signo… (Peirce en Sercovich, 1987: 244)

Los conceptos de la semiótica resultan pertinentes dado que hemos estado elucidando sobre la máquina de Turing, y su utilización mediante algoritmos que son instrucciones que pertenecen a un lenguaje con determinados significados. El signo que representa algo (representamen), es transformado en otro signo (interpretante). Lo utilizamos en cierta forma a lo que refiere Peirce en los lenguajes de programación de las computadoras, cuando los códigos de un programa fuente se transforman a códigos de la máquina.

El autómata "construye" el árbol sintáctico de la instrucción, en el proceso de traducir la instrucción al código de la máquina. En este sentido se tiene una aplicación de la máquina de Turing, referida a los compiladores como programas que obtienen el programa en código de máquina.

De otro lado, Norbert Wiener, estudioso y fundador de la cibernética y conocedor de los temas propuestos por Turing, en su libro *Dios y Golem S.A.*, publicado en 1963, refiere a que las máquinas son dispositivos que convierten mensajes.

> Una máquina es un dispositivo para convertir mensajes de entrada en mensajes de salida. Un mensaje, desde un punto de vista, es una secuencia de cantidades que representan señales en el mensaje. Tales cantidades pueden ser corrientes o potenciales eléctricos, pero no se reducen a éstos, y en verdad pueden ser de naturaleza muy diferente (Norbert Wiener, 1963: 40-41)

Cabe destacar que para Norbert Wiener el lenguaje tiene un rol importante para manejar aparatos o grupos humanos, relacionando las reflexiones de Leibnitz en su interés por la lingüística: "… aun al ocuparse de las máquinas de calcular, el interés capital de Leibnitz residía primordialmente en la lingüística y en las comunicaciones" (Norbert Wiener, 1958:19)

Añadiendo con respecto al lenguaje: “sin embargo, deseo insistir en que el lenguaje no es un atributo exclusivo de los seres vivientes, pues lo comparten en cierto grado las máquinas construidas por el hombre” (Norbert Wiener, 1958:70).

3.5. Tesis de Turing.

Turing publica en 1950 un artículo titulado *Computing Machinery and Intelligence*, en la revista de filosofía *Mind* de octubre de 1950, refiriendo a la posibilidad de “inteligencia” en las máquinas, dejando en forma clara el significado de inteligencia artificial, como un concepto de simulación mediante la ejecución de un procedimiento, proponiendo un juego de imitación en la que participan tres personas: un hombre, una mujer y un interrogador.

> El interrogador está en una habitación separado de los otros dos (hombre y mujer), y el objetivo es determinar cuál es la mujer y cuál es el hombre. Para este fin el interrogador hace preguntas a los otros dos, la mujer se presenta como una ayuda al interrogador y el hombre procura enredar y confundir al interrogador. Las preguntas del interrogador pueden ser: ¿Diga cuan largo es su pelo? Las respuestas de la mujer pueden ser: Yo soy la mujer, no le crea a él, esto no ayuda mucho porque el hombre puede hacer la misma afirmación, con lo cual se presenta una situación de naturaleza compleja, por un lado se tiene ayuda y por el otro solo confusión y además no sabemos de dónde viene una u otra; para precisar mejor el ambiente físico el interrogador es la computadora y los jugadores están en ambientes separados, además la comunicación entre los jugadores y la computadora es a través de un equipo similar a un télex, es decir no hay ninguna posibilidad física para identificar el sexo, solo será posible mediante las preguntas y respuestas. (Turing, 1950).

El documento conocido como Test de Turing, sostiene que es posible el diálogo, entre la computadora y las personas, tal como lo enuncio en su ejercicio para identificar el sexo de las personas. Expone un dialogo de preguntas y respuestas, que tiene la restricción de limitar las formas coloquiales, reduciéndolas a mantener una jerarquía de preguntas y respuestas.

En 1980, John Searle[142] publica su artículo *Minds, Brains and Programs* en la revista *The Behavioral and Brain Sciences*[143], en la que dice que demuestra la contradicción de la propuesta de Turing, al referirse con un ejemplo conocido como el experimento de la Sala China, que trata acerca de una persona en una habitación, que sólo conoce el idioma inglés y recibe mensajes en chino y debe responderlos en el mismo idioma, además no conoce en lo absoluto el chino; entonces si recibe y envía mensajes, estaría actuando sin entender nada de lo que ocurre. Este argumento sostiene que la inteligencia en la máquina es un proceso que no entiende, entonces se deduce la contradicción. El argumento de Searle no es fuerte, porque supone la inteligencia del tipo cognitiva en la máquina de Turing; pero el concepto de Turing sobre "inteligencia" en la máquina no refiere a entender, más bien a la ejecución mecánica de la instrucción, por tanto no participa el entendimiento. Esta es la tesis que sostiene Turing, sobre procesos que sólo ejecutan.

En el año 1950, Turing manifiesta que dentro de los próximos cincuenta años[144] será posible construir programas y computadoras que tengan mayor capacidad de procesamiento y de almacenamiento, por lo tanto, podrán manejar mayor cantidad de preguntas y situaciones, que permitan la identificación del sexo de los jugadores, tal como lo propuesto en el Test. Asimismo en el documento refuta ideas en el mismo sentido de

---

[142] John Roger Searle, nace el 31 de julio de 1932 en Denver Colorado. Profesor de Filosofía de la universidad de California Berkeley. Obtiene su doctorado en filosofía en 1959 en Oxford. Con diversas publicaciones. Conocido por su tesis de la habitación China para refutar los planteamientos de Turing sobre inteligencia en las máquinas.

[143] Penrose lo menciona en su libro *Las sombras de la mente*, en el primer capítulo, el argumento de John Searle, ataca el argumento del tipo Inteligencia Artificial fuerte, pero concluye que no es total mente concluyente.

[144] En una discusión presentada por la BBC de Londres, el 10 de enero de 1952, entre Richard Braithwaite, Geoffrey Jefferson, Max Newman y Alan Turing, el tema refería a que las máquinas de cálculo automático pueden "pensar". Turing afirma que dentro de 100 años esto será posible, aumentando la fecha al año 2052.

como lo hizo en su documento de 1948, denominándolas objeciones para entender la posibilidad de máquinas “inteligentes”.

Roberto Perazzo[145] menciona un hecho ocurrido en 1991, referente a la exposición de programas de computadora, basados en juegos de imitación, realizado en el Museo de Computación de Boston, Estados Unidos, nos dice que al final del evento se registró que las máquinas lograron engañar a cinco de los diez jueces que actuaron como interrogadores. (Perazzo, 1994: 101).

Si bien los argumentos expuestos por Turing no son contrastables contundentemente, tampoco pueden ser descartados, además según lo expuesto, se presenta el concepto de “inteligencia” para las computadoras, como una simulación en que las máquinas logran confundir a jueces humanos. Si el concepto es de simular o “engañar”, la prueba corresponde a esta situación específica.

La máquina de Turing según la tesis Church-Turing, tiene correspondencia con el cálculo lambda, en cuanto a la recursividad que tiene la ventaja de la simplificación expresiva en la formalización conceptual de las instrucciones.

Ejemplo: Si deseamos obtener la suma de los N primeros números naturales, el programa escrito mediante definición recursiva, debe calcular la suma de los números anteriores, así sucesivamente hasta llegar al número 1, cada paso exige a que se inicie el programa y cuando terminen, regresan sus resultados al proceso anterior, así hasta terminar el cálculo, siendo el programa siguiente:

[145] Roberto Perazzo físico y notable investigador argentino, autor del libro *De Cerebros, Mentes y Máquinas.*

```
Función Sumar (N)
    Si N = 1 entonces
        Suma = 1
    Caso contrario
        Suma = N + Sumar (N – 1)
    Fin si
Fin Función
```

Es posible obtener otro programa que calcula la suma de los N primero números, pero de forma no recursiva, las instrucciones están en forma secuencial, para nuestro ejemplo se necesitan N instrucciones, seria tal largo en instrucciones como números tenga que sumar, las instrucciones serían las siguientes:

```
Suma = 1
Suma = Suma + 2
Suma = Suma + 3
…
Suma = Suma + N
```

La especificación de las instrucciones en forma recursiva fue una condición necesaria a inicios de la computación porque se buscaba hacer programas que ocupen la menor cantidad de espacio debido a la poca capacidad de memoria en las máquinas. Hoy en día esto no es un problema, resulta similar el especificar en forma recursiva o en forma secuencial, en el primer caso la máquina va a trabajar más, debido a que tendrá que preguntar en cada momento de ejecución si el contador interno llega al límite.

En los diversos documentos que tratan sobre la máquina de Turing se suele minimizar la comprobación para una buena ejecución, se entiende de la participación directa del hombre, como si fuera parte de la máquina, como en la máquina-O. Esta característica acompañará a Turing en los diversos trabajos de investigación, idea importante que está en el mismo concepto de autómata[146]. La correspondencia de máquina con el hombre, hace que algunos pensadores lo relacionen con el de inteligencia en el hombre, ignorando que lo que trata Turing es sobre procedimientos automático.

Turing investigó la computabilidad, en el sentido del procedimiento efectivo en relación a su máquina, elucidando detalles mínimos de lo que entendemos por algoritmo. Copeland,[147] nos dice que Turing nunca hizo mención directa a la Tesis de Church-Turing, pero interpreta cierta referencia implícita de Turing en su ensayo: *The Chess*, que fue publicado en 1953 en la colección *Faster Than Thought*, en la sección titulada *Digital Computers Applied to Games*. El párrafo al que se refiere es el siguiente:

> Si uno puede explicar con bastante ambigüedad en el inglés y con la ayuda de símbolos matemáticos, si es requerido, como un cálculo que se debe hacer, entonces siempre es posible programar cualquier computadora digital para hacer ese cálculo, siempre que la capacidad de almacenamiento sea adecuada (Turing en Copeland, 204:567).

Copeland señala lo mencionado por Turing: “no es el tipo de asunto que admite la reducción clara de la prueba” (Turing en Copeland, 204:567-568). Consideramos que la interpretación de Copeland es en cierto sentido entendible al referir un método matemático, suponiendo que es así y lo dice de la siguiente manera: “como un cálculo que está hecho por un obediente empleado en concordancia a un método matemático” (Copeland, 204:568).

---

[146] Una definición de autómata elaborada Ricardo Cardoso puede ser consultada en: http://agamenon.uniandes.edu.co/rcardoso/Cursos/MESw/Material/Automatas/Automatas1.pdf

[147] Jack Copeland (nacido en 1950) es profesor de filosofía en la Universidad de Canterbury (Nueva Zelanda). Recibió su D.Phil y B.Phil en filosofía de la Universidad de Oxford en 1979 por sus investigaciones sobre lógica modal y no-clásica. Es director del Archivo Turing para la Historia de la computación en Canterbury (Nueva Zelanda) desde 1985. Es considerado un experto en los temas de Alan Turing.

El concepto que postula Turing sobre máquinas inteligentes, no refiere a una inteligencia en la forma como se presenta en el hombre, la interpreta como una simulación mediante los algoritmos, que se manifiesta en las máquinas en correspondencia "física" a lo que se ejecuta mediante un programa.

De otro lado, encontramos pertinente subrayar que la máquina de Turing es diferente a la teoría de Turing, tal como lo expresa Piscoya: "esto es, la máquina de Turing es recursiva pero la teoría de la máquina de Turing, que es uno de los productos notables del pensamiento humano, no lo es" (Piscoya, 2009: 273), aclarando que la máquina de Turing es pieza importante en la teoría de Turing, aunque no estamos de acuerdo con Piscoya, porque la trata como equivalente a la función recursiva, mientras para nosotros sostenemos que la máquina se encuentra en el contexto de la definición de lo computable en relación a la naturaleza física y de la complejidad de los procesos mediante dispositivos con estructura lógica, contenida en la teoría de Turing.

La máquina de Turing participa en el significado de la inteligencia artificial, como máquina abstracta que ejecuta procesos siguiendo un orden establecido, sin mayor reflexión. El algoritmo expresa la correspondencia de los estados de la máquina en relación con las instrucciones y dispositivos físicos con los que actúa.

Tomando en cuenta el avance en las ciencias naturales, observamos que aún no se ve una alternativa contundente en la construcción de la máquina-O y tampoco se ha logrado hacer el programa que resuelva exactamente lo propuesto por Turing en 1950 en su famoso test. Aún quedan diversos temas a ser resueltos en el tema de lo que puede ser expresado en forma de programas de las computadoras, pero esto no desamina al hombre. Existen diversos laboratorios en Inteligencia Artificial que están investigando estos temas.

Constatamos el desarrollo de la ciencia y tecnología, materializadas en la computadora, siendo denominada por algunos, como uno de los inventos tecnológicos más importantes del siglo XX. Hoy esta observación no sorprende a muchos, pero en opinión de Turing en 1950 manifestó que se llegará a tener computadoras con mayores capacidades y velocidades, que puedan engañar a las personas como si la máquina estuviera “pensando”, como ocurre con la ejecución de programas de computadora que “juegan” como una persona, como el caso del ajedrez, donde hay programas que ganan partidas de juego a maestros. Éstos son conocidos como programas de estrategias[148].

La computadora tiene múltiples usos, dentro de los cuales podemos agruparlas en cuatro: El primero como almacenamiento y procesamiento de datos, que fue el uso tradicional de las máquinas, el segundo como medio de comunicación, utilizando la interconexión con otras computadoras compartiendo datos, notándose hoy en el Internet[149], el tercer grupo ejecuta tareas de apoyo al trabajo de oficina o empresa, editando archivos en formas diversas, y el cuarto grupo de usos diversos, como son los programas de juegos - archivos multimedia - porque combinan en un solo archivo, sonido, imagen y lógica.

La computadora es una máquina que contiene componentes de naturaleza lógica, la

[148] Se han desarrollado diversos programas en que el jugador construye una civilización que está formada por ciudades y tecnologías y compite contra la computadora. También hay programas en Internet, en donde los jugadores interactúan con otros, bien siendo un buque de guerra o un granjero, según las reglas del juego.

[149] El Internet incorpora nuevos conceptos, para mayor información consultar en http://www.hacienda.go.cr/centro/datos/Articulo/Glosario%20b%C3%A1sico%20ingl%C3%A9s-espa%C3%B1ol%20para%20usuarios%20de%20Internet.doc. Sábado, 28 de junio, hora 20:00 horas.

que resulta materializada en un equipo. Funciona mediante programas que contienen órdenes, es posible referirse a ésta en dos componentes, una que corresponde a su estructura física y la otra a los datos y programas.

Destacamos el hecho de que para que el hombre construya una computadora, primero debió conceptualizarla, había que tener un proceso de abstracción y de formulación. En esto radica la existencia de estas ideas, que se expresaron en la construcción de las primeras computadoras y el sentido de los programas, que permiten la ejecución de las mismas, imitando máquinas.

3.6. Computabilidad y vida artificial.

Turing conocía de programación de las computadoras, también fue usuario de las mismas, así lo menciona Copeland en referencia a una carta que Turing escribe a un colega del *National Physical Laboratory*, a principios de 1951, en la que muestra su alegría por la llegada de la computadora y que ésta le serviría para sus estudios de los diversos patrones que se presentan en la naturaleza como la filotaxis[150] de la serie de Fibonacci[151], como sucesión numérica que está presente en diversas estructuras biológicas.

> Nuestra nueva máquina [la Ferranti Mark I] empieza a llegar el lunes. Estoy esperando con uno de los primeros trabajos para hacer algo sobre 'embriología química'. En particular creo que uno puede dar cuenta de la aparición de los números de Fibonacci en relación con el abeto-conos (Turing, 1951:1).

La "embriología química" estudia el desarrollo de la estructura anatómica en el embrión animal, es el resultado de la difusión de productos químicos que reaccionan entre

[150] Filotaxis es la disposición que presentan las hojas en el tallo. La disposición se presenta en características en cada especie, se sostiene que cada hoja busca tener una mayor exposición al sol.

[151] La serie de números de Fibonacci son aquéllos que los dos primeros números se mantienen fijos y los siguientes son la suma de los dos anteriores: 1, 2, 3, 5, 8,... n, m, n+m …

sí, formando patrones espaciales. En su documento *The Chemical Basis of Morphogenes,* de 1952. Hace una modelización matemática de lo que llegaría a ser conocido por los estudiosos de la morfogénesis en la biología como los patrones de Turing.

Según Damian Strier[152] en su tesis para optar el grado de Doctor en Ciencias Físicas sobre Procesos de Auto-Organización en Sistemas Biológicos, menciona a Turing como pionero en este campo de investigación.

> Turing demostró la plausibilidad de la formación espontanea de un patrón espacial estacionario de morfógenos en un medio inicialmente uniforme. Para ello supuso solamente la existencia de reacciones químicas y de un proceso de transporte difusivo de los morfógenos (Strier, 2002:13)

El modelo matemático propuesto por Turing trata de la morfogénesis de unas manchas, resulto suficiente para establecer un camino de investigación.

> … es claro que además de la información codificada a nivel genético intervienen otro tipo de factores los cuales determinan el patrón de encendido selectivo de genes en las distintas células. Este encendido… puede originarse… presencia de gradientes químicos… Esta posibilidad fue analizada por Alan Turing, encontrando que un sistema de reacción difusión con dos reactivos es, bajo ciertas condiciones, capaz de auto organizarse espontáneamente (Strier, 2002:61)

Turing en su publicación de 1952 formula un cálculo que resuelve ecuaciones que para haber sido un resultado manual, debió dedicar horas, obteniendo de ella un gráfico que son unas manchas, similares a las que tiene un perro de raza dálmata, sugiriendo para continuar en estos temas de cálculo, el uso de la computadora.

> Este proceso es muy conveniente para la computación, y por lo tanto se puede aplicar a dos dimensiones. La figura 2 muestra un patrón, el cual se obtiene en pocas horas por un cálculo manual. (Turing 1952:27)

[152] Damian Strier, argentino, doctor en ciencias físicas, investigador de los procesos auto organizados en los sistemas biológicos, el cual es una orientación que acerca la física a las reacciones químicas.

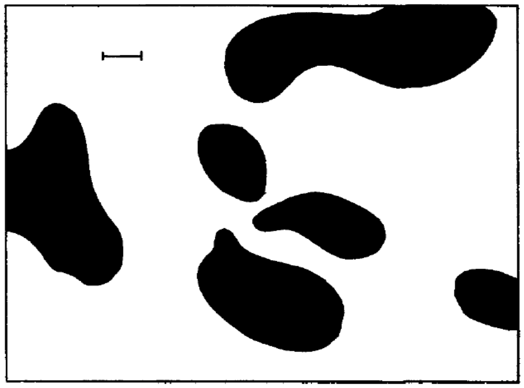

Figure 2. An example of a 'dappled' pattern as resulting from a type (*a*) morphogen system. A marker of unit length is shown. See text, §9, 11.

Figura 8. Grafico obtenido por Turing basado en cálculos de ecuaciones diferénciales[153].

Así mismo el interés de Turing por la genética, queda manifiesto en una carta dirigida a C.H. Waddington[154], el 11 de setiembre de 1952, le indica su interés en las manchas que están presentes en diversos animales, éstas son las que tienen las cebras, tigres, leopardos u otro animal. Copeland menciona que Waddington comentó sobre la publicación de Turing como: "… que la aplicación más clara de la teoría de Turing parecía ser en los moretones que se producen en el deporte, manchas y rayas que se producen cosas de áreas uniforme aparentes." (Copeland, 2004: 509)

El documento de Turing sobre morfogénesis es una modelización matemática sobre reacciones químicas, utiliza para este fin ecuaciones diferenciales[155], que describen sistemas dinámicos. El trabajo de Turing contiene cierto rigor matemático, él así lo menciona aunque reconoce el propósito divulgativo del tema.

[153] Obtenido del libro de Jack Copeland (2004).

[154] Conrad Hal Waddington (1905- 1975) fue un biólogo, paleontólogo y genetista escocés, uno de los fundadores de la biología de sistemas.

[155] Las Ecuaciones Diferenciales son ecuaciones que están presentes las derivadas de funciones de una o más variables, ejemplo: $a_n(x)y^{(n)} + a_{n-1}(x)y^{(n-1)} + \ldots + a_1(x)y' + a_0(x)y = g(x)$, si estas son de una variable se denominan Ecuaciones Diferenciables Ordinarias, caso contrario son Ecuaciones Diferenciables Parciales.

> El propósito de este trabajo es discutir un posible mecanismo por el cual los genes de un cigoto pueden determinar la estructura anatómica del organismo resultante. La teoría no tiene ninguna nueva hipótesis, solo sugiere que ciertas leyes físicas conocidas son suficientes para explicar mucho de los hechos. El entendimiento completo del documento requiere un buen conocimiento de las matemáticas, la biología, y algunos de química elemental. Dado que los lectores no pueden esperar a ser expertos en todos estos temas, una serie de hechos elementales se explican, que se puede encontrar en los libros de texto, pero la omisión haría el documento de lectura difícil. (Turing, 1952:1)

Turing desarrolla el modelo conocido como reacción y difusión, ponderando las reacciones químicas, explicando:

> En este documento se propone prestar atención más bien a los casos donde el aspecto mecánico puede ser ignorado y el aspecto químico es la más significativa. Estos casos prometen mayor interés, por la acción característica de los propios genes es presumiblemente química. (Turing, 1952:2)

Supone que las velocidades de reacción obedecen a la “ley de acción de masas” y está en proporción a la concentración de las sustancias que participan de la reacción, presenta un ejemplo de la velocidad en la que se formará el cloruro de plata y precipitará del nitrato de plata y de sodio por acción del cloruro.

> $Ag^{+} + Cl^{-} \rightarrow Ag\,Cl$. Será proporcional al producto de concentración de iones de plata Ag^{+} y los iones de Cloro CL^{-}. Esto es señalado en la ecuación $Ag\,NO_3 + Na\,Cl \rightarrow Ag\,Cl + Na\,NO_3$ (Turing, 1952:5)

Turing explica al final de su documento que lo tratado tiene consideraciones de límite y que la principal desventaja es que sólo obtiene resultados para casos particulares. Aclara que puede ser superado con la utilización de la computadora, asimismo explica que los temas considerados se ven pequeños ante la gran variedad de casos que se presentan en los complicados fenómenos biológicos. Finaliza indicando que es posible modelar matemáticamente muchos casos, y que su trabajo se orienta en la comprensión de las formas biológicas.

> Teniendo esto en combinación con las matemáticas utilizadas en este trabajo son relativamente elementales, difícilmente se podría esperar encontrar que muchos fenómenos biológicos observados estarían cubiertos. Se cree, sin embargo, que los imaginarios sistemas biológicos que han sido tratados, y los principios que se han discutido, deben ser de alguna ayuda en la interpretación real de las formas biológicas. (Turing, 1952:43)

El concepto de computabilidad utilizado por Turing está relacionado a los temas de la "vida" artificial, en referencia al algoritmo genético, que es el procedimiento que se emplea para simular la evolución natural para producir generaciones sucesivas de entidades definidas. Turing presenta la idea de un algoritmo genético[156] en su publicación *Intelligent Machinery,* de 1951, en la que denomina un procedimiento de búsqueda genética o evolutiva.

> Hay la búsqueda genética o evolutiva, por la cual una combinación de genes es buscado con un criterio de valor de supervivencia. El notable éxito de esta confirmación de búsqueda es en cierta medida la idea de que en la actividad intelectual consiste principalmente de diversos tipos de búsqueda (Turing, 1951:22)

La investigación de estos temas están en el mismo sentido al realizado por Turing, fue de interés de prestigiosos matemáticos, en cuanto a mecanismos del tipo auto reproductivos, como se muestra en los trabajos de von Newmann, referidos por Copeland, al describir una carta que von Neumann dirige a Norbert Wiener, el 29 de noviembre de 1946, sobre los mecanismos de auto reproducción: "Yo pensaba mucho sobre mecanismos auto-reproductivos. Puedo formular el problema rigurosamente alrededor del estilo en el que Turing lo hizo para sus mecanismos" (Copeland, 2004:515).

---

[156] Algoritmo Genético es el término introducido en la Ciencia de la Computación por John Holland en 1975 en la Universidad de Michigan.

Copeland explica que Turing había formulado las ideas sobre los procedimientos autos reproductivos, expresando que la teorización realizada por von Neumann acerca de la auto-reproducción, fue fuertemente influenciada por la definición de la máquina de Turing universal, así lo dice en una carta de Herman Goldstine[157], amigo y conocedor de los trabajos que realizaba von Newman: “von Neumann tuvo un profundo interés por los autómatas. En particular, siempre tuvo un profundo interés por el trabajo de Turing”, luego Copeland añade lo mencionado por Goldstine sobre el concepto de autómata: “Turing previó un excepcional e inesperado resultado. …en esencia, lo que el mostró es que en cualquier autómata particular puede ser descrito por un conjunto finito de instrucciones” (Goldstine en Copeland 2004: 515).

Turing en su ensayo *The Chess,* de 1953, trata sobre las computadoras digitales aplicadas a los juegos, establece criterios de naturaleza formal, en la que argumenta de la posibilidad de construir un programa donde la computadora actúe como un jugador en una partida de ajedrez. Empieza su ensayo con la pregunta ¿Podría uno hacer que una máquina juegue ajedrez?, respondiendo que es posible plantear varios significados en relación a la misma pregunta. Dice que una computadora electrónica programada adecuadamente se constituye en sí misma en una máquina, en referencia a la máquina universal de Turing. De otro lado, como posibilidad hacia el futuro, propone una máquina que construida adecuadamente permitirá jugar ajedrez.

En el ensayo, Turing trata sobre el mérito del programador que logre codificar el programa que incorpore un mecanismo de “aprendizaje”, en el sentido de que el programa

[157] Herman Heine Goldstine (13 de septiembre de 1913 – 16 de junio de 2004), matemático, informático y administrador, fue uno de los principales desarrolladores de ENIAC, el primer computador electrónico digital de propósito general.

deberá tener reglas que haga posible el "aprendizaje" de situaciones que se presentan en el juego de ajedrez. Así lo expresa:

> Si este producto obtuvo resultados que son bastante nuevos y de este modo interesante para el programador, ¿Quién podría tener el crédito? Compare esto con la situación de que el Ministro de Defensa ha dado órdenes para investigar y contrarrestar el arco y la flecha. ¿Debería el inventor del escudo tener el crédito o debería ser para el Ministro de la Defensa? (Turing, 1953:7)

Antes de fallecer Turing, en 1954, presenta el documento *Solvable and Unsolvable Problem,* que resulta ser una explicación sencilla del *Entscheidungsproblem,* resuelto en su documento *On computable numbers, with application to the Entscheidungsproblem,* de 1936, con la diferencia metodológica, de que en el documento de 1936 desarrolla el concepto de la máquina abstracta, ahora en 1954, revisa el caso mediante un juego de rompecabezas, que trata de una cuadricula de 16 espacios, que contiene 15 fichas y una vacía, de forma que permite el desplazamiento de las fichas vecinas hacia el lugar vacío.

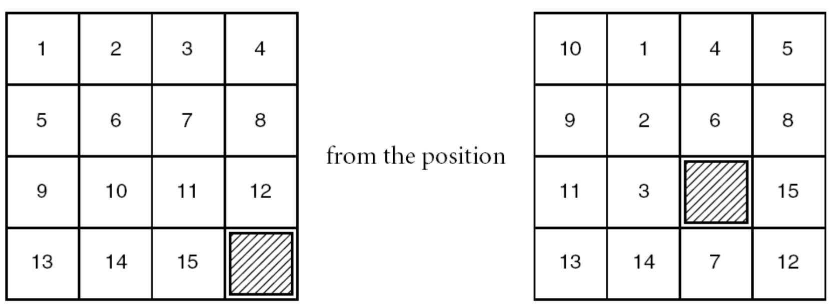

Figura 9. Puzzle de 16 casilleros, figura obtenida de Turing, 1954.

El artículo fue publicado en Science News[158] y trata sobre problemas que admiten una solución algorítmica, Turing trabaja un ejemplo de un problema que no es resoluble por cualquier método sistemático, a la pregunta: ¿Un puzzle para todos los casos tiene solución? La respuesta es que no se puede encontrar un método que responda la pregunta

[158] Science News, revista de divulgación científica de la época (1954).

de forma afirmativa o negativa. Turing prueba que sólo es posible obtener la solución en los casos que se obtienen desde una situación inicial en el tablero del puzzle que dé luego de varias modificaciones queda "desordenada" (en relación a la posición inicial). Reflexiona sobre el ejemplo del tablero de 16 casilleros (4 x4) y 15 fichas, calculando las posibles posiciones de las fichas, resultando 2`092,278`988,000, y las posibilidades se reducen si se considera en el tablero un estado inicial y a partir de éste producimos los posibles estados finales del tablero (Turing, 1954:3).

En el documento Turing analiza el caso del puzzle con infinitos casilleros, y concluye que no sería posible encontrar una solución, incluso utilizando una búsqueda exhaustiva[159]; para los casos de tableros finitos, es posible un procedimiento de búsqueda sin descartar la utilización de la intuición, según Copeland, las reglas formales no excluyen el uso de la intuición en el método utilizado por Turing: "El argumento de resolubles y problemas sin solución ilustra porqué la necesidad de que la intuición no siempre puede ser eliminado en favor de las reglas formales" (Copeland, 2004:580).

Turing relaciona el concepto de función computable con el de procedimiento efectivo, aclarando la similitud de ambos, en el sentido que establece un método que encuentre una explicación entre lo que se puede y no se puede decidir, remontándose a su documento de 1936.

> … se puede reducir a la definición de "función computable" o "procedimiento sistemático". Una definición de cualquiera de estos puede definir todos los demás. Desde 1935 una serie de definiciones se han dado, explicando en detalle el significado de uno u otro de estos términos, y estos han demostrado ser equivalentes entre sí y por lo tanto equivalente a la declaración anterior. (Turing: 1954:8).

[159] Búsqueda exhaustiva, es el tipo de búsqueda que revisa o evalúa todos los posibles casos.

Define el concepto de procedimiento sistemático como aquel que se ejecuta para resolver un juego del tipo puzzle, precisando que tendrá una solución, si el resultado corresponde a la posición inicial de las fichas, solo así tiene sentido el procedimiento sistemático.

> Antes de que podamos examinar esta cuestión adecuadamente debemos ser bastante claros que significa para nosotros "procedimiento sistemático" para decidir una cuestión. Pero ahora esto no tiene por qué darnos cualquier dificultad particular. Un "procedimiento sistemático" fue una de las frases que hemos mencionado como equivalente a la idea de puzzle, ya sea porque se podría reducir una de la otra. Si ahora estamos claros sobre lo que es un puzzle, entonces debemos ser igualmente claros sobre "procedimiento sistemático". De hecho, el procedimiento sistemático es un puzzle en el que nunca hay más de una posible movida en cualquiera de las posiciones que se plantean y en la que algunos significados se adjuntan con el resultado de final. (Turing, 1954:9)

En el documento Turing explica que por limitaciones de espacio no pudo tratar de forma más extensa temas como:

> (1) No es posible resolver el problema de decisión sobre la sustitución de procesos aplicados solo a las filas fichas blancas y negras. (2) Hay ciertos puzles particulares para los que no existe un procedimiento de decisión, las reglas son fijas y el único elemento variable es la posición inicial. (3) No existe un procedimiento para decidir si un determinado conjunto de axiomas no conduce a una contradicción o no. (4) El "problema de la palabra en semi-grupos con la cancelación" no tiene solución. (5) Recientemente se ha anunciado de Rusia de que el "problema de la palabra en grupos" no tiene solución… (6) Existe un conjunto de 102 matrices de orden 4, con coeficientes enteros examinados que no existe un método de decisión para determinar la conveniencia de otra matriz dada es o no expresable como un producto de matrices a partir de lo dado. (Turing, 1954:13)

Consideramos que Turing sintetiza al conjunto de los temas tratados, en la búsqueda en aspectos complejos, relacionados a lo computable, esta se manifiesta explícitamente en la carta de 1951, dirigida al biólogo John Young[160], en la que le menciona sus intenciones sobre la investigación en morfogénesis, y su interés en las redes

[160] John Zachary Young (Bristol 18 de marzo de 1907 – Oxford 4 de julio de 1997) Zoólogo y neurofisiólogo inglés.

neuronales. La carta de dos páginas explica casi en la mitad de ella, temas relativos al cerebro y su capacidad de almacenar datos, es comparada con las formas que se dan en las computadoras, indica que estos estudios son más difíciles de alcanzar. Precisa que está trabajando temas más sencillos que se orientan hacia la comprensión del funcionamiento del cerebro humano.

> Me temo que estoy muy lejos del escenario, donde me siento inclinado a empezar a hacer alguna pregunta anatómica [del cerebro]. De acuerdo con mis ideas de cómo establecer al respecto que no ocurrirá hasta una etapa tardía, cuando tengo una teoría bastante clara acerca de cómo se hacen las cosas. En la actualidad no estoy trabajando en el problema en absoluto, pero en mi teoría matemática de la embriología… dar explicaciones satisfactorias sobre i) gastrulación[161]. ii) la poligonalidad de las flores y estructuras simétricas, por ejemplo, estrellas de mar. iii) la disposición de las hojas, en especial a las que están involucradas en la forma. La serie de los números de Fibonacci (0, 1, 1, 2, 3, 5, 8, 13 ,...). iv) los patrones de color de los animales, por ejemplo, rayas, manchas y motas. v) Los patrones en las estructuras casi esféricas examinándolas como en algunos radiolarios[162], pero esto es más difícil y dudoso. (Turing, 1951: 2)

Turing trabajó una teoría alrededor de la definición abstracta de máquina. De esta forma asoció el concepto de procedimiento efectivo con las que permitan operar aquello que es decidible, en orientación a lo propuesto en el *Entscheidungsproblem,* específicamente avanzó en el concepto de computabilidad en temas diversos, siendo la principal orientación e interrogante para entender el funcionamiento del cerebro.

La teoría de Turing tiene como constante la utilización de los algoritmos como procedimiento efectivo, estos se relacionan a los programas de las computadoras que simulan inteligencia mediante procedimientos mecánicos, que operan dispositivos que tienen componentes que cumplen reglas de naturaleza lógica.

[161] Gastrulación es el proceso en el que se forma las capas germinales del embrión que originan todos los tejidos el futuro bebé.
[162] Radiolario es un grupo de organismos microscópicos, unicelulares (protozoos) marinos.

“Máquinas” de Turing. Estas máquinas son seres humanos que calculan. Y lo que él dice también podría expresarse en forma de juegos[163].

Wittgenstein

CAPÍTULO IV

COMPUTABILIDAD Y MÁQUINA DE TURING

En el presente capítulo evaluamos el concepto de computabilidad expresada en la tesis de Church-Turing, comparando las ideas contenidas en la propuesta de Church con las enunciadas por Turing, las que ya fueron expuestas en los capítulos anteriores.

Defendemos la tesis de que el concepto de computabilidad en Church es de diferente significado al de la teoría de Turing, dado que en este último es de mayor profundidad en relación al significado de algoritmo.

En los inicios de 1930, el concepto de computabilidad se asoció al de función recursiva y al de procedimiento efectivo tal como lo enuncia la tesis de Church-Turing, ya desde esa fecha se dieron diferencias de opinión con respecto a las ideas que rodean la mencionada tesis, como ocurrió entre Wittgenstein y Gödel (con respecto a interpretaciones de los teoremas de incompletitud) o entre Gödel y Church (con respecto al formalismo de algoritmo), así también entre Gödel y Turing (con respecto al concepto

[163] Wittgenstein en §1096, en el primer volumen de *Observaciones a la Filosofía de la Psicología*.

de inteligencia en las máquinas).

Incorporamos en la argumentación los trabajos de von Neumann[164] y los de Norbert Wiener[165], que fueron publicados en la década de los 50 del siglo pasado, sus afirmaciones parecían "futuristas", tal vez por esta razón no se tomaron en cuenta en la constitución inicial de la teoría de la computación, manteniendo distancia e incorporada en cierta forma en las ingenierías. Hoy constatamos que los resultados de las investigaciones de Von Neumann sobre la arquitectura de las computadoras y de Norbert Wiener sobre Cibernética, están presentes en temas como robótica, inteligencia artificial, órganos artificiales, entre otros. Temas relacionados al concepto de lo computable.

El presente capítulo resalta las diferencias entre la tesis de Church y la teoría de Turing en relación a que el concepto de computabilidad en la máquina de Turing es más amplio al sostenido en la tesis de Church. La teoría de Turing se sostiene sobre la definición de algoritmo, expresada formalmente en la máquina abstracta de Turing, para este fin utilizamos lo enunciado por ilustres investigadores que califican como filósofos, matemáticos y lógicos, que gozan de prestigio mundial por sus resultados académicos

---

[164] John von Neumann zu Margitta, (28 de diciembre de 1903 - 8 de febrero de 1957) matemático húngaro-estadounidense, de ascendencia judía. Recibió su doctorado en matemáticas de la Universidad de Budapest a los 23 años. Padre de la teoría de juegos, publicó *Theory of games and economic behavior* junto a Oskar Morgenstern, en 1944. Pionero de la computadora y de la aplicación de la teoría de operadores a la cuántica. La computadora EDVAC, desarrollada por Von Neumann, Eckert y Mauchly. Otra de sus inquietudes fue la capacidad de las máquinas de autorreplicarse.

[165] Norbert Wiener (26 de noviembre de 1894, Columbia, Missouri - 18 de marzo de 1964, Estocolmo, Suecia) matemático estadounidense, fundador de la Cibernética. Acuñó el término en su libro Cibernética o el control y comunicación en animales y máquinas, publicado en 1948. En setiembre de 1906, a los once años, ingresó en la Universidad Tufts para estudiar matemáticas. Se licenció en 1909 y entró a Harvard a estudiar zoología, en 1910 se trasladó a la Universidad Cornell para empezar estudios en filosofía. Volvió a Harvard para continuar sus estudios de filosofía. Obtuvo el doctorado en 1912 con una tesis sobre lógica matemática. De Harvard pasó a Cambridge, Inglaterra, donde estudió con Bertrand Russell y G. H. Hardy. En 1914 estudió en Göttingen, Alemania con David Hilbert y Edmund Landau. Entre 1915 y 1916 enseñó filosofía en Harvard y trabajó para General Electric y la Encyclopedia Americana en Maryland. Profesor de matemáticas en el MIT. Es uno de los precursores de la teoría de la comunicación.

como muestran su publicaciones, así tenemos: Martin Davis[166] , Stephen Kleene, Wilfried Sieg[167], Robert Soare[168], Stewart Shapiro[169] y Dina Goldin[170].

4.1. Tesis de Church - Turing.

La tesis Church-Turing define la computabilidad de una función computable, resultado importante en la ciencia de la computación debido a que ayudo a formular una disciplina del conocimiento humano: La ciencia de la computación[171].

El éxito más evidente de la tesis de Church-Turing se manifiesta en la formulación de la teórica de los compiladores, que trata sobre los lenguajes para hacer programas que funcionan en las computadoras, en estrecha relación con los conceptos de algoritmo y de recursividad.

---

[166] Martin Davis, (nacido en 1928 en Nueva York) matemático estadounidense conocido por su trabajo relacionado con el décimo problema de Hilbert. Obtuvo su PhD en la Universidad de Princeton en 1950 y su tutor fue Alonzo Church. Es profesor emérito de la Universidad de Nueva York. Es coinventor del algoritmo de Davis-Putnam y de los algoritmos DPLL. También es coautor, junto con Ron Sigal y Elaine J. Weyuker de *Computability, Complexity, and Languages, Second Edition: Fundamentals of Theoretical Computer Science*

[167] Wilfried Sieg, profesor de filosofía en Universidad Carnegie Mellon, Pittsburgh desde 1985, matemático, físico y lógico. Ph.D. en Stanford University, en Filosofía, Lógica Matemática, MS, matemáticas y lógica, en Westfälische Wilhelms-Universität, Münster, BS, Matemática y Física en Freie Universität, Berlín. Ha ocupado el cargo de jefe del departamento de filosofía de la Universidad Carnegie Mellon, ha realizado diversas publicaciones en libros y ensayos sobre temas diversos de historia y filosofía de la matemática, análisis de la computabilidad, teoría y búsqueda automática.

[168] Robert I. Soare, matemático, obteniendo los grados de A.B. en la Universidad de Princeton en 1963, Ph.D. en Matemáticas en la Universidad de Cornell en 1967, es profesor de matemáticas de la Universidad de Chicago, tiene diversas publicaciones: Teoría de la recursividad y los cortes de Dedekind, Métodos fundamentales para la construcción de grados recursivamente enumerable, La computabilidad y la recursividad, historia y el concepto de computabilidad.

[169] Stewart Shapiro, profesor de Filosofía en la Universidad estatal de Ohio y profesor visitante de la Universidad de Saint Andrew en Escocia. Estudio matemáticas y filosofía en la Universidad de Case Westerm Reserve, obtuvo su MA y Ph.D. en matemáticas en la Universidad de Nueva York en Buffalo, condecorado como O'Donnell Proffesor of Philosophy, tiene diversas publicaciones de libros y ensayos.

[170] Dina Goldin, Ph D. en Computer Science en la Universidad de Brown, profesora en la universidad de Brown, consultora e investigadora, ha escrito diversos ensayos respecto a la teoría de la computabilidad.

[171] El término *Computer Science* fue acuñado por George Elmer Forsythe, matemático especializado en análisis numérico que fundó uno de los primeros departamentos de Ciencia de la Computación en Estados Unidos.

La tesis se enuncia de la siguiente forma: Una función es efectivamente computable si sólo si es Turing computable. Esta incluye dos conceptos, el primero sobre la función efectivamente computable que refiere a las funciones recursivas que son un cálculo efectivo, enunciado por Church en el cálculo lambda; el segundo sobre el procedimiento efectivo, definición dada por Turing en el sentido de ejecutar exitosamente una secuencia de instrucciones en una máquina de Turing.

La tesis Church-Turing trata sobre el algoritmo para el cálculo de una función matemática, y es imposible una demostración matemática, dado que el concepto es de naturaleza intuitiva.

Tanto Church como Turing, trabajaron sus investigaciones en relación con los resultados del segundo teorema de Gödel (teorema de la incompletitud). Así ambos demuestran en forma diferente la naturaleza de realizar el cálculo, y la posibilidad e imposibilidad de calcular, tomando como base los números enteros, específicamente Church prueba la existencia de cálculos efectivos mientras que Turing prueba la indecibilidad de ciertos cálculos.

El matemático Kleene, alumno y amigo de Church, en el año 1952, publica en su libro *Introduction to Metamathematic*, la tesis de Church-Turing, el libro con un poco más de 564 páginas, es una defensa del concepto de función computable en relación a la función recursiva y que es posible de ser computada mediante una máquina de Turing: "La evidencia de que el análisis es completo, es decir, que para cualquier función que sea evidentemente calculable puede hallarse una máquina de Turing que la compute". (Kleene, 1952:322)

Asimismo, Kleene propone la tesis Church-Turing en el teorema XXX, la equivalencia de la tesis de Church con lo enunciado por Turing.

> La tesis de Turing de que toda función que sea naturalmente considerada como computable, es computable en el sentido por él especificado, esto es, computable por una máquina de Turing, es equivalente a la tesis de Church por virtud del teorema XXX. (Kleene, 1952:340).

El Teorema XXX, trata de las funciones computables, que resultan ser las funciones recursivas parciales, funciones computables, funciones 1/1 computables. Formulando el teorema como la unión de los teoremas XXVIII y XXIX, que dicen: "toda función recursiva parcial ℓ es 1/1 computable y también toda función ℓ recursiva parcial en ℓ funciones Ψ, completamente definidas es 1/1 computable" (Kleene, 1952:328) y que "toda función parcial computable ℓ es recursiva parcial y toda función parcial ℓ computable a partir de ℓ funciones Ψ completamente definidas es recursiva parcial en Ψ" (Kleene, 1952: 337).

Kleene estudió permanentemente la computabilidad de las funciones matemáticas, incorporando el concepto abstracto de máquina, que es una definición que sería parte de la teoría de autómatas, que trata de los algoritmos que trabajan secuencias de caracteres expresadas en lo que se denomina forma normal de Kleene: Cadenas de caracteres que tienen representaciones de formas recursivas. Así mismo, en la definición de Turing se encuentra el concepto de algoritmo como secuencia de instrucciones, siendo interpretado como equivalente a las funciones recursivas, ganando aceptación en la comunidad científica en computación, interpretándose la tesis de Church-Turing en la definición que relaciona el concepto de algoritmo con el significado de computabilidad.

En ciertos campos de la ciencia de la computación se ha ratificado la "validez" de la tesis, como se muestra en: La teoría de los lenguajes de computación, en la definición de las estructuras de datos y en diversos procesos para adicionar, eliminar o buscar datos. El concepto de recursividad-algoritmo ha contribuido en la construcción de los lenguajes de programación de computadoras, tales como el LIPS, que es un lenguaje que opera estructuras de listas de datos mediante funciones del cálculo lambda, y también el lenguaje de programación PROLOG, que utiliza funciones recursivas de proposiciones lógicas de primer orden. Ambos lenguajes de programación fueron muy populares en la década de 1970 a 1980, pero hoy son desplazados por lenguajes de programación de naturaleza más intuitiva para los programadores, como son los lenguajes de orientación visual, basada en dibujar los formularios de pantallas.

En el campo de la lógica, se menciona el concepto de computabilidad en el mismo sentido a lo expresado por Kleene, así tenemos en opinión de Mosterín y Torretti:

> Una función asigna valores a los elementos de su dominio. Algunas funciones parecen ser incomputables, como la función que asigna a cada ser humano la fecha de su muerte… otras funciones parecen ser computables, como las funciones aritméticas de adición, sustracción, multiplicación y división … Una función es computable si existe un algoritmo que nos indica cómo computarla de un modo efectivo. La computabilidad es la propiedad de ser computable. La noción de función computable ha sido exitosamente precisada en la lógica y la teoría de la recursión mediante nociones tales como la de la función recursiva o máquina de Turing (Mosterín y Torretti, 2002, 100-101)

El concepto de computabilidad suele ser definido mediante los conceptos de recursividad y de máquina de Turing, así también lo expresa Ferrater Mora:

> Se llama decidible a un cálculo C cuando puede forjarse un método o un procedimiento mecánico mediante el cual sea posible decidir – en una serie de operaciones finita – si una fórmula bien

> formada de C es o no un teorema de C… si existe en un cálculo C o una teoría formalizada T un procedimiento o método de decisión es llamado problema de decisión. Si se encuentra tal procedimiento o método, el cálculo o la teoría formalizada reciben el nombre de decidibles; si no, el de indecidible… Las anteriores definiciones no tienen carácter formal. Para una definición formal suficiente del término ‘decidible’ aplicando a una teoría formalizada T usaremos la formulación de A. Tarski (*Undecidable Theories*, 1953). Una teoría es llamada decidible si el conjunto de todas sus funciones válidas es recursivo; de lo contrario, es llamada indecidible. Siguiendo al mismo autor diremos que una teoría formalizada T puede ser 1) decidible, 2) Indecidible 3) esencialmente indecidible (Ferrater Mora, 2004, 786)

Lo indicado por Ferrater Mora, corresponde a la explicación de la palabra decisión, entendemos que se encuentra en el mismo sentido dado por Hilbert al plantear la unificación de las matemáticas, y resuelto negativamente mediante el teorema de Gödel.

Hoy la definición de computabilidad tiene significados que no pueden desligarse de la computadora y su funcionamiento. El concepto de computable está incorporado en nuestro lenguaje cotidiano. Algunos científicos de la ciencia de la computación cuestionan el término de computabilidad en su única equivalencia con las funciones recursivas, como Robert Soare, quien sostiene: “Específicamente nosotros recomendamos: El término recursivo ya no lleva el significado adicional de computable o decidible” (Robert Soare, 1996)

Es importante considerar que el concepto de computabilidad, expresada en la tesis Church-Turing, se construye entre los años 1930 a 1960, y se considera que en este último año nace la disciplina de la ciencia de la computación (en Estados Unidos). La computabilidad en un contexto histórico, se presentó en la búsqueda de la solución del problema *Entscheidungsproblem*, en medio de la crisis de los fundamentos de las matemáticas, expresadas en las tres corrientes matemáticas, lográndose la fabricación de la computadora y manifestándose en lo que el hombre puede calcular.

4.2. Etimología de cálculo y computable.

Para elucidar el significado de la palabra computable, nos referirnos a su etimología, dado que toda palabra de alguna manera mantiene un núcleo fuerte de sus significados iniciales, la comparación con el significado original nos ayuda a la comprensión del significado actual. La palabra computación, computabilidad y sus derivados, lo aprendimos desde textos escritos en inglés, incorporándose recientemente al español. Así en el diccionario de la Real Academia Española, encontramos la definición de la palabra computable: “Adjetivo. Que se puede computar”, consultada la definición de la palabra computar en el mismo diccionario, señala:

> Se deriva del latín computāre. (1). Verbo transitivo. Contar o calcular por números algo, principalmente los años, tiempos y edades. (2). Verbo transitivo. Tomar en cuenta, ya sea en general, ya de manera determinada. Usado también como pronominal. Se computan los años de servicio en otros cuerpos. Los partidos ganados se computan con dos puntos (Real Academia Española, 2009).

La Real Academia define la palabra computar como la acción de contar o calcular usando números enteros positivos. Así mismo, al consultar la palabra computador en el mismo diccionario, indica la siguiente definición:

> (1). Adjetivo. Que computa (calcula). Úsese también como sustantivo. (2). calculador (aparato que obtiene el resultado de cálculos matemáticos). (3). computadora electrónica. (4). calculadora (aparato que obtiene el resultado de cálculos matemáticos). (5). computadora electrónica. Máquina electrónica, analógica o digital, dotada de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas matemáticos y lógicos mediante la utilización automática de programas informáticos. (Real Academia Española, 2009).

El término computar se asocia al significado de calcular, esto se expresa en la cuarta definición (4), por lo tanto, podríamos suponer que la calculadora sería un equivalente a computadora. En la quinta definición (5) sobre máquina electrónica, refiere al concepto de tratamiento de la información, como una extensión del significado de computadora que resulta ser una definición más amplia de las calculadoras.

En el castellano, específicamente en España, se usa la palabra ordenador para designar a la computadora, que resultó ser copia del vocablo *ordinateur* usado en el lenguaje francés. Esto ocurrió debido a la influencia de la empresa IBM[172], dado que esta empresa en España y en otros países de Europa, había logrado vender con mucho éxito equipos que eran tabuladores de tarjetas (que contenían perforaciones que representaban datos), que resultaron de gran ayuda en los procesos de la contabilidad, las máquinas ordenaban y clasificaban las tarjetas, de ahí la palabra ordenador en su sentido más amplio. IBM no usó el término computadora para nombrar sus equipos, tanto así, que en la guerra de Corea (1951), el gobierno de los Estados Unidos le encarga a IBM la construcción de una computadora con mayor potencia de cálculo, nombrándola *Defense Calculator*, y le asigna el código 701, luego basada en esta máquina, producirá en serie su modelo 650 que fueron nombradas *Data Processing Systems*, fue recién en 1981[173] en que lanza al mercado su famosa computadora con el nombre de *Personal Computer (PC)*[174], aceptando el

[172] La empresa IBM empieza con Herman Hollerith, joven inmigrante alemán que trabajaba para el censo estadounidense, desarrolló una máquina perforadora que agilizaba el trámite censal. La presentó al concurso público del momento, lo ganó y montó su propia empresa “Tabulating Machine”. Tras varias fusiones con otras empresas toma el nombre de Computing-Recording Company, luego Thomas J. Watson se puso al frente y en 1924 le cambió el nombre, llamándola Internacional Bussines Machines. IBM se dedicaba a la fabricación de todo tipo de máquinaria: balanzas industriales, cronómetros, cortadores de carne o queso.

[173] El IBM PC compatible, fue lanzado en agosto de 1981. El modelo original fue llamado "IBM 5150". La frase "computadora personal" era de uso corriente antes de 1981, y fue usada por primera vez en 1972 para denominar al Xerox PARC's Alto. Sin embargo, debido al éxito del IBM PC, lo que había sido un término genérico llegó a significar específicamente una computadora compatible con las especificaciones de IBM.

[174] El término "computadora personal" apareció en la revista New Scientist en 1964, en una serie de artículos llamados «El mundo en 1984». En un artículo titulado *The Banishment of Paper Work*, Arthur L. Samuel, del Centro de Investigación Watson de IBM escribió: «Hasta que no sea viable obtener una educación en casa, a través de nuestra propia computadora personal, la naturaleza humana no habrá cambiado».

nombre de computadora para referir al equipo electrónico y todos sus componentes (monitor, teclado, discos y memorias etc.)

Resaltamos, que las primeras máquinas electrónicas fueron construidas para apoyar en los cálculos científicos, y dado el éxito en este campo se inició una carrera para mejorar su potencia de cálculo, cada vez que se iba a terminar la construcción de una nueva computadora se estaba pensando en la siguiente. Ésta es una característica en la fabricación de computadoras manteniéndose hasta el día de hoy.

En el diccionario etimológico de Joan Corominas[175] encontramos la definición de la palabra *disputar*, indicando que derivó posteriormente el verbo *putare* y que en el siglo XVII derivo a la palabra calcular, en este sentido, Corominas afirma: "Tomado del latín disputare 'examinar' o discutir (una cuestión)' 'discutir, disertar' (derivado de putare 'limpiar' 'podar (una planta)'. 'contar, calcular'). Siglo XVII latín putativus 'que se calcula'" (Corominas, 1990).

Resaltamos la palabra en latín *putare* en su significación equivalente a las palabras de contar y/o calcular. En el diccionario etimológico de la lengua castellana de Pedro Felipe Monlau[176], indica sobre el prefijo *com*:

> Con, co, com, cor, cum. De la proposición latina cum, que en lo antiguo se escribió com… En

[175] Joan Coromines Vigneaux (Barcelona, 1905 — Pineda de Mar, Barcelona, 1997) fue un filólogo español autor del *Diccionario crítico etimológico castellano e hispánico* y del *Diccionari Etimològic i Complementari de la Llengua Catalana*. Se lo considera uno de los grandes romanistas y lexicógrafos del siglo XX.

[176] Pedro Felipe Monlau (Barcelona, 1808-Madrid, 1871) Médico y escritor español. Fue catedrático de la Universidad de Madrid y director de la Escuela de Diplomacia. Escribió numerosas obras de medicina legal, higiene y psicología, entre las que destacan *Elementos de higiene pública* (1847), *Higiene del matrimonio* (1853) e *Higiene industrial* (1856).

> castellano con se conmuta en co cuando le sigue vocal ó h, v. gr. En co-etaneo… en com cuando el simple principia por b ó p, v. gr. En com-binar… en cor cuando le sigue r, v. gr. Cor-regidor… y toma la forma cum en cumplir en sus compuestos y derivados. Con expresa la relación por la cual dos o más personas, dos o más cosas están juntas, relación muy sencilla en sí, pero que se hace muy complexa a causa de las ideas accesorias que en muchos casos se le agrega. Con expresa compañía, reunión, cooperación, agregación, ensambladura, y el prefijo colectivo, amplificativo e intensivo, por excelencia… (Monlau, 1856)

En el latín la palabra *computare* está formada por el prefijo *com* y el verbo *putare*. El prefijo *com* utilizado en el sentido de juntar y de amplificación del verbo *putare* que fue mencionado como significado de cálculo. La palabra *computare* adquiere el sentido de referirse a calcular en un sentido mayor al de cálculo, dado a la forma amplificativa e intensiva del prefijo. De otro lado la palabra cálculo proviene de la palabra en latín *calculus* que significa piedra y es natural que sostengamos que las piedras servían como ayuda para contar, agrupar o separar cantidades.

Encontramos natural lo expresado en cuanto a la cantidad de piedras y la dificultad de manipular una enorme cantidad de piedras, indicando sobre los límites del cálculo, mientras que en el uso de la palabra cómputo se amplifica el significado. Así mismo, el concepto de calcular lo encontramos referido a la utilización de objetos auxiliares para realizar operaciones, haciendo extensiva nuestra capacidad cognitiva, amplificándola como cuando ocurre al utilizar las piedras.

4.3. Filosofía de la Computación

Reflexionamos sobre la filosofía en la ciencia de la computación, en el sentido de precisar algunos conceptos que nos permitan deslindar interpretaciones que consideramos erradas, aclarando que aún se encuentra en debate el significado de lo computable, en contra de aquellos que consideran que el tema está cerrado, como que el concepto solo fuera de naturaleza matemática y que corresponde únicamente a la función recursiva, ignorando el hecho de que la ciencia de la computación si bien tiene relación con la ciencias formales, resulta que es diferente en cuanto a su campo de acción. La computación no es una escuela de las matemáticas, es una disciplina independiente.

En cuanto a los temas filosóficos en la computación, consideramos lo expresado por William Rapaport[177]: "Sorprendentemente la filosofía de la ciencia de la computación está lejos de ser bien desarrollada", esta expresión explica sobre el estado de la cosa y siguiendo este sentido propone la implementación de un curso que fue dictado en la primavera del 2004, en la universidad de New York en Buffalo, dirigido simultáneamente a los estudiantes del departamento de ciencia de la computación e ingeniería con los del departamento de filosofía, ambos correspondientes al quinto año de estudio. Rapaport precisa: "La filosofía de la ciencia de la computación no es la filosofía de la inteligencia artificial (AI); por supuesto, esta incluye la filosofía de la AI, pero se extiende más allá de su ámbito de ocupación" (Rapaport, 2005)

En la actualidad existen pocos cursos que tratan sobre la filósofa de la ciencia de la computación, sus contenidos suelen dedicarse al estudio de la inteligencia artificial, pero hay temas diversos que requieren ser analizados y estudiados. Rapaport propone un contenido base de un curso, en el que incluye temas y libros, abordando preguntas tales como: ¿Qué es ciencia de la computación? ¿Es la ciencia de la computación ciencia o

[177] William J. Rapaport, profesor de la Universidad de Búfalo, investigador de la ciencia de la computación, inteligencia artificial, lingüística computacional, ciencia cognitiva, lógica y matemáticas, graduado de filosofía en la Universidad de Indiana.

ingeniería? ¿Qué es un algoritmo?, por otro lado Raymond Turner[178] y Amnon Eden[179] en el documento *The Philosophy of Computer Science* publicado en el diccionario filosófico on line de Stanford confirman lo expresado, mencionando:

> La filosofía de la ciencia de la computación concierne a cuestiones filosóficas que surgen de la reflexión sobre la naturaleza y práctica de la disciplina académica de ciencia de la computación. Pero ¿Qué es esto? Ciertamente no solo es programación. Después de todo, mucha gente que escribe programas no es científica de la computación. (Turner y Eden, 2008)

Turner y Eden comentan sobre la naturaleza dual de los programas, por un lado la naturaleza del texto en relación a las instrucciones del lenguaje, y de otro lado el de la naturaleza del proceso mecánico en relación a la utilización de un programa en una organización u oficina.

Estos aspectos confirman el problema filosófico sobre la identidad de un programa, expresando que hay la necesidad de precisar que es un programa: Puede entenderse que es el texto fuente en el que está escrito o es el archivo que se ejecuta en una computadora en un determinado momento.

En cuanto a la semántica de un programa, suele ser referido al resultado o bien la forma de cómo realiza la ejecución, en el sentido que si dos programas hacen lo mismo pero tienen formas diferentes de ejecución. Obligándolos a precisar sobre la semántica del código de las instrucciones.

---

[178] Raymond Turner, profesor de la universidad de Essex, doctor en matemáticas de la universidad de Londres, y doctor en filosofía en la misma universidad, con diversas publicaciones sobre matemáticas, filosofía de la lógica, y filosofía de la ciencia de la computación.

[179] Amnon Eden, miembro del staff de la Scholl of Computer Science & Electronic Engineering, con diversas publicaciones sobre programación de computadoras, diseño y modelamiento de datos, inteligencia artificial, y filosofía de la ciencia de la computación.

Turner y Eden muestran como ejemplo, el cálculo de factorial de un número natural, presentando dos programas, el primero con funciones recursivas y el segundo utiliza bucles. Sosteniendo que son semánticas diferentes a pesar que realizan el mismo cálculo, dado que el código de máquina es diferente uno del otro.

El primer ejemplo definido mediante funciones recursivas, así tenemos:

```
Funcion Factorial (N)
        Si N = 1 entonces
                Factorial = 1
        Caso contrario
                Factorial = N * Factorial (N – 1)
        Fin si
Fin Funcion
```

Notamos que una vez que se llama a la función Factorial (n), esta llamará a la misma función Factorial (n-1), así sucesivamente hasta llegar a la función Factorial (1), luego resuelve el valor con un mecanismo de cálculo de arriba hacia abajo, para volver con los valores obtenidos y completar el cálculo en cada caso de abajo hacia arriba.

El segundo programa ejemplo para el cálculo de factorial es mediante bucles, así tendríamos la siguiente codificación:

```
Funcion Factorial (N)
        K = 1
        F = 1
        Hacer mientras K <= N
                F = F * K
                K = K + 1
        Fin Hacer
        Factorial  = F
Fin Funcion
```

Este programa ejecuta el cálculo de forma diferente al anterior, multiplica directamente todos los números implicados según lo establece el bucle.

Cuando referimos a la implementación de un programa, indicamos al código objeto que deberá contener el computador para que ejecute lo expresado. Este código se obtiene mediante programas llamados compilador[180]. Los compiladores en su arquitectura tienen dos componentes: el primero se denomina analizador sintáctico y el segundo el analizador semántico, ambos analizan el programa si está bien escrito y que todas las instrucciones cumplan con las reglas sintácticas del lenguaje del programa fuente, para finalmente generar el código objeto. El código objeto está en relación al analizador semántico. Los programas pueden tener el mismo significado en cuanto calculan el mismo valor, pero desde el punto de vista del código objeto, utilizan diferentes significados ya que sus códigos objetos son diferentes, expresados, en la ejecución. El significado está en correspondencia al código obtenido como consecuencia de la traducción realizada, de un código del programa fuente hacia el código de máquina.

Siguiendo con la presentación de interrogantes epistemológicas en la ciencia de la computación, Turner y Edén, precisan con respecto a la máquina de Turing, indicando que hay dos controversias: Una histórica y otra empírica, que se centran en dos posibles interpretaciones:

> I) Las máquinas de Turing pueden hacer cualquier cosa que pueda ser descrito como "regla de oro" o "puramente mecánico". II) Lo que puede ser calculado por una máquina (trabajando con datos

[180] Un compilador es el programa que traduce el texto codificado en un lenguaje de programación (el que elabora un programador) a un programa en código de la máquina. El segundo programa se dice que está en lenguaje de la máquina. Para mayor información ver http://compilers.iecc.com/crenshaw/

> finitos de acuerdo a un programa finito de instrucciones) es computable en la máquina de Turing. (Turner y Eden, 2008)

Aclaran que la interpretación (I) captura la noción de un método efectivo mecánico en la lógica y en las matemáticas, mientras en la interpretación (II) el método efectivo corresponde a una máquina física. En este caso se trata de la computadora. Esto precisa el concepto de computabilidad en dos orientaciones, una en relación al método efectivo en la lógica y las matemáticas y el otro a las máquinas mecánicas eléctricas.

Consideramos adecuada la interpretación de Eden, cuando formula la existencia de tres orientaciones en el pensamiento de la ciencia de la computación, así indica:

> El paradigma racionalista, que fue común entre teóricos científicos de la computación, define la computación como una rama de las matemáticas, trata de los programas a la par de los objetos matemáticos, y tiene por objeto un conocimiento a priori a cerca de sus "correcciones" por medios de razonamiento deductivo. El paradigma tecnocrático, promulgada principalmente por ingenieros de software, definen la ciencia de la computación, como una ciencia disciplina de la ingeniería, trata a los programas como simple datos y búsquedas probables, y un conocimiento a posteriori sobre su fiabilidad empírica usando pruebas particulares. El paradigma científico, prevalece en las ramas de la inteligencia artificial, define la ciencia de la computación como una ciencia natural (empírica), toma a los programas como entidades a la par de los procesos mentales, y busca a cerca de estos conocimiento a priori y posteriori, mediante la combinación de la deducción formal y experimentación científica (Eden Amon, 2007:1)

Eden utiliza el término paradigma en el sentido formulado por Thomas Kuhn[181]. Paradigma [182] como forma en las que las comunidades científicas entienden y/o perciben y/o interpretan las teorías científicas:

[181] Thomas Samuel Kuhn (Cincinnati, 18 de julio de 1922 - 17 de junio de 1996) filosofo estadounidense. Ph.D en física por la Universidad Harvard en 1949 y tuvo a su cargo un curso Historia de la Ciencia en Harvard de 1948 a 1956. Profesor en la Universidad de California, Berkeley hasta 1964, en la Universidad de Princeton hasta 1979 y en el MIT hasta 1991. En su obra *La estructura de las revoluciones científicas* (1962) introduce el concepto de paradigma.

[182] Jesús Padilla Gálvez, en su libro Tratado Metateórico en la tesis científica refiere a que Materman le descubre 21 modos distintos en que Kuhn emplea el término paradigma. Esto se encuentra en el artículo *On Nature of Paradigmas*, 1970.

> Sin embargo, hasta cierto punto, la práctica de la astronomía, de la física, de la química o de la biología, no evoca, normalmente, las controversias sobre fundamentos que en la actualidad, parecen a menudo endémicas, por ejemplo, entre los psicólogos o los sociólogos. Al tratar de descubrir el origen de esta diferencia, llegue a reconocer el papel desempeñado en la investigación científica por lo que, desde entonces llamo 'paradigmas'. Considero a estos como realización científicas universalmente reconocidas, que durante cierto tiempo, proporcionan modelos de problemas y soluciones a una comunidad científica. En cuanto ocupo su lugar esta pieza de mi rompecabezas, surgió rápidamente un bosquejo de este ensayo" (Kuhn, 1970:13-14)

Tomamos en cuenta que Kuhn ante la cantidad de críticas recibidas por el uso no muy claro del término paradigma, prefirió cambiar de nombre a la de Matriz Disciplinar, descrito así en su artículo: Segundas Reflexiones Acerca de los Paradigmas.

> Habrá menos confusión si en lugar de ella se utiliza la expresión matriz disciplinar. Disciplinar, porque es la posesión común de los que practican una disciplina profesional, matriz, porque está compuesta por elementos ordenados de varias clases… Ni siquiera intentare dar una lista exhaustiva, sino identificare… tres de ellos… generalizaciones simbólicas, en particular, son aquellas expresiones desarrolladas sin cuestión previa alguna por parte de grupo que pueden ser fácilmente vaciadas en una forma lógica… modelos, acerca de los que no diré nada más en este artículo, aquello que provee al grupo de analogías relevantes o, cuando se los mantiene con profundidad, de una ontología. …..los ejemplares son soluciones concretas a problemas, aceptadas por el grupo como paradigmáticas en un sentido bastante general" (Kuhn en Suppe, 1979:513)

Preferimos denominar como orientación en vez de paradigma, en el sentido dado por Eden, es así como refiere Peter Wegner[183], quien dice: Que durante la década de 1950 hay una preponderancia de la orientación científica, con dominio a una concepción empírica basada en juicios a posteriori; en la década de 1960 hay predominancia en la orientación racionalista, con el dominio de las teorías matemáticas basados en los juicios de naturaleza a priori, y en la década de 1970 el predominio de la orientación tecnocrática.

---

[183] Peter Wegner, nacido en 1932, científico de la computación, ha hecho importantes contribuciones en la programación orientada a objetos en la década de 1980, ha escrito sobre la C*omputación Interactiva* como un nuevo enfoque de la computación. En 1999, fue galardonado con la Cruz de Honor de Austria para la Ciencia y el Arte. Actualmente es profesor emérito de la Universidad de Brown.

Estas definiciones dadas por Wegner fueron publicadas en la década de 1970.

Eden y de Wegner, nos presenta la interrogación, si lo que denominamos como ciencia de la computación es ciencia o una ingeniería. No está en el alcance de la presente investigación el contestar estos temas, pero si el presentarlos para que nos ayuden a explicar el concepto de computabilidad, dado que contiene una carga metodológica, ontológica y por tanto filosófica.

Si la orientación es racionalista, el método a utilizar en grandes rasgos es el de las matemáticas, la ontología del caso considera los programas y los datos como expresiones matemáticas, y el tratamiento epistemológico corresponderá a notaciones matemáticas para referir a los procedimientos y datos.

Si la orientación es tecnológica, el método a utilizar es el de las ingenierías, la ontología del caso considera los programas y los datos como códigos e instrucciones que permiten el control de equipos físicos, mecánicos y/o eléctricos, y el tratamiento epistemológico corresponderá a la tecnología.

Si la orientación es científica, el método a utilizar es el de las ciencias, éstas corresponderán a las ciencias fácticas como ocurre en el caso de la Inteligencia Artificial, la ontología del caso considera los programas y los datos como representaciones teóricas en relación al mundo.

Eden en su artículo *Three Paradigms of Computer Science*, postula que siguiendo cada una de las orientaciones enunciadas, se tiene una postura epistemológica y por tanto una visión ontológica, implicando significados que se diferencian unos de otros. Como ejemplo en los programas, que pueden ser concebidos como una función matemática, o como una secuencia de instrucciones que opera un dispositivo.

Consideramos que el significado que aceptamos como concepto de computación depende de lo que entendemos por objetos y las interpretaciones expresadas en ideas, así resulta pertinente referirnos a dos pensadores Frege y Pierce para elucidar lo que entendemos sobre los significados.

Frege sobre el concepto de significado, en su obra de 1879, *Conceptografía,* crea una notación lógica y trasluce una crítica al psicologismo, al afirmar que la lógica no tiene que ver con los procesos mentales. Avanzó en su doctrina semántica que sería la base de su logicismo, llegando a publicar en 1891 su texto *Función y Concepto* que nos dice sobre expresiones que son iguales en significado aunque no se expresan en el mismo sentido. Aclarando sobre la confusión que se produce entre los términos, forma y contenido o entre signo y cosa designada: "… la diferencia de la designación no puede bastar ella sola para fundamentar una diferencia en las cosas designadas" (Frege en Valdés, 1998: 55)

Siguiendo con el texto, Frege aclara sobre la función, en la que nos lleva al concepto de argumento: "Sin embargo, es justamente la notación consistente en escribir <x>, que indica de manera indeterminada, la que nos lleva a la concepción correcta. Se llama a x el argumento de la función" (Frege en Valdés, 1998:57).

Muestra la distinción entre función como entidad no saturada, porque puede tomar diferentes valores, siendo el objeto una entidad saturada porque toma un valor, así indica:

"… pues a la función, por si sola, hay que llamarla incompleta, necesitada de compleción o insaturada. Y de este modo se diferencian de modo fundamental las funciones de los números" (Frege en Valdés, 1998: 58)

Frege explica que si bien el número 2^4, es el mismo que se obtiene de 4x4, indica que tienen la misma referencia (refieren al número 16) pero no tienen el mismo pensamiento, nada impide el escribir $2^4 = 4x4$. En su documento de 1892 *Sobre Sentido y Referencia,* nos presenta la relación entre el signo, sentido y referencia, así específica:

> La conexión regular entre signo, su sentido, y su referencia, es de tal género, que al signo le corresponde un sentido determinado y a este, a su vez, una referencia determinada, mientras que a una referencia (a un objeto) no le pertenece solo un signo (Frege en Valdés, 1998:86)

Resaltamos lo expresado por Frege, porque entendemos que el signo es un componente importante para capturar el significado: "Un nombre propio (palabra, signo, combinación de signos, expresión) expresa su sentido, se refiere a, o designa, su referencia. Con un signo expresamos su sentido y designamos su referencia" (Frege en Valdés, 1998:90)

Frege relaciona el pensamiento con el sentido, indicando que la referencia es diferente al pensamiento, la referencia tiene correspondencia al valor de verdad de una oración. Según lo expuesto por Frege y en el objetivo de elucidar los argumentos sobre la equivalencia de la tesis de Church y la teoría de Turing, específicamente entre las definiciones de cálculo efectivo y procedimiento efectivo, expresados en la tesis de Church-Turing, en la que ambas definiciones refieren a lo mismo, a la función computable, y son definiciones que tienen distintos sentidos (son diferentes pensamientos). Por lo tanto, siguiendo esta reflexión, no se justifica la presente investigación sobre el análisis de las diferencias, dado que significan lo mismo. Este tipo de análisis la consideramos reduccionista porque ignora el signo como un componente necesario en el significado.

Frege en su artículo *Sobre Sentido y Referencia* establece como característica en la transmisión de un pensamiento, que la idea viaja acompañada con otros pensamientos vinculados, por tanto hay más pensamientos que oraciones, lo que hace de la comunicación un proceso muy rico.

> Parece que casi siempre, al emitir un pensamiento principal, asociamos con él, pensamientos concomitantes que, aunque no son expresados, el oyente los vincula también con nuestras palabras en virtud de leyes psicológicas. Y puesto que tales pensamientos concomitantes aparecen asociados por si mismos a nuestras palabras, casi como el propio pensamiento principal, también nosotros queremos expresar tal pensamiento concomitante. Con esto se vuelve más rico el sentido de la oración, y puede muy bien suceder que tengamos más pensamientos simples que oraciones (Frege en Valdés, 1998:107)

Resaltamos de que el pensamiento "viaja" en compañía de otros pensamientos. Estos pensamientos concomitantes, pueden estar en cierto grado fuera del sentido del pensamiento principal, al respecto Frege añade: "en otros puede ser dudoso si el pensamiento concomitante pertenece al sentido de la oración o solo lo acompaña" (Frege en Valdés, 1998: 107)

Las funciones tienen relación a los predicados, así como expresamos de la computadora como un objeto, en la idea de ser una máquina de Turing, así Frege en su artículo de 1892, *Sobre Concepto y Objeto*, profundiza las nociones de concepto y objeto y su relación con las nociones de sentido y referente, aclarando que el concepto no es predicable refiriéndose a una singularidad: "… un concepto es la referencia de un predicado, un objeto es lo que jamás puede ser referencia total de un predicado, si bien puede ser la referencia de un sujeto" (Frege en Valdés, 1998:110).

Sostenemos que las reflexiones de Peirce complementan en el significado de signo, en este sentido estamos de acuerdo con María Rivas[184] quien sostiene que existe relación entre los conceptos dados por ambos pensadores (Frege y Pierce), en la noción fregeana de sentido y la noción peirceana de interpretante.

Para Rivas el representamen es el signo que refiere a un objeto, el interpretante es el que media entre el signo y el objeto (referencia).

> … el sentido o el interpretante - tiene un carácter mediador entre los otros dos - el signo y el objeto. El signo o representamen funciona como un significante que hace referencia a un objeto… Los análisis del signo de Frege y Peirce pretenden ser un modelo general y formal de lo que es, en principio, ser un signo, de lo que serían las condiciones que se requieren para que algo sea un signo (Rivas, 2002:4)

Así Rivas establece en su investigación un análisis paralelo entre lo que refieren al signo Frege y Peirce:

> Así pues, según Frege, el sentido de un signo que es un nombre propio contiene el modo de darse el referente; según Peirce, el objeto determina al signo a que lo represente de una cierta manera, y esa forma de representar el signo a su objeto es el interpretante. Es decir, el sentido y el interpretante están mediando entre el signo y el objeto, de tal forma que el objeto tiene una cierta capacidad determinado de cómo el signo va a referir a él (Rivas, 2002:5)

Rivas enfatiza acerca de la existencia de las semejanzas de la noción de signo de Frege y de Peirce: "Uno de los puntos más interesantes de las propuestas de Peirce y Frege

[184] María Uxia Rivas Monroy, profesora del departamento de Lógica y Filosofía Moral de la Universidad de Santiago de Compostela, Doctorado en Alemania, especialista en filosofía del lenguaje, historia de la filosofía analítica, pragmatismo semiótica, autora de varios trabajos sobre C.S. Peirce, Frege, y Putnam.

es la semejanza que, desde el punto de vista epistemológico, presentan sus teorías acerca del signo, a pesar de la diferencia terminológica existente entre ellos" (Rivas, 2002: 14).

Rivas en su artículo de 1996 sobre Frege y Peirce: *En Torno al Signo y su Fundamento,* escribe sobre el rasgo más característico de abordar el estudio de los signos a través de un esquema tríadico, indicándonos:

> La noción fregeana de modo de darse y la peirceana... son idénticas ni con el sentido ni con el objeto inmediato, respectivamente. Ambas nociones procede del objeto, y en este sentido pueden considerarse como propiedades o cualidades. En definitiva como primeros. … se transforma en categorías mediadoras… se transforman en terceros siendo respectivamente sentido e interpretante se transforma en categorías mediadoras entre el signo y el objeto, esto es, se transforman en terceros, siendo respectivamente sentido e interpretante. (Rivas, 1996: 14)

Rivas establece la importancia del signo, en relación a su correspondencia a la realidad, por esta razón encontramos adecuado el estudio de las expresiones consideradas equivalentes, existiendo en ellas una riqueza epistemológica que nos ayuda a comprender las esencias de lo que transmiten a través de los signos:

> … no resulta extraña y fuera de lugar la comparación entre el sentido fregeano y el interpretante peirceano… Esta la pretensión de este estudio… estas nociones las que permitirían explicar porque los signos nos permiten comprender y conocer la realidad (Rivas, 1996: 14).

En la ciencia de la computación, específicamente en los lenguajes de programación de computadoras, encontramos tres elementos que se relacionan entre sí, el programa formado por una secuencia de instrucciones, que están escritas siguiendo un formato que cumplen con las reglas de un lenguaje de programación; las operaciones básicas (código) que constituyen la forma el cómo realizan determinadas operaciones en la computadora; y el instante de la ejecución del programa en cuanto a la obtención del resultado. Estos tres elementos: programa, lenguaje y ejecución, se relacionan de forma tal que podemos decir en un sentido amplio, que tienen tres campos de interpretación: sintaxis, semántica y pragmática.

La clasificación de sintaxis (formas y reglas de escritura), semántica (formas de interpretación expresada en códigos de máquina) y pragmática (uso específico), ayudan a interpretar por el que un programa escrito en un lenguaje de programación, llega a ser un proceso que se ejecuta en un determinado instante. Estos tres elementos trabajan sobre interpretaciones que son códigos. Encontramos en ellos una fuerte relación con lo tratado en los signos.

Sostenemos que es una interpretación errada el considerar solo la semántica de un algoritmo, en cuanto a su resultado, ignorando los códigos de la computadora, dado que consideramos que existen significados en los códigos de ejecución, el error se obtiene de considerar al primero como a una semántica y el segundo a una sintaxis. Esta forma de interpretación reduce lo computable a la formulación del problema, ignorando que en la codificación existe una semántica, como ocurre en un sistema axiomático.

4.4. Computabilidad.

Sostenemos que el concepto de computabilidad es más amplio al expresado mediante la función recursiva, entendemos que la utilización de este tipo de función matemática en el inicio de la ciencia de la computación, inspiró y posibilitó la construcción de una teoría de la computación, fue a través de las funciones recursivas donde la teoría alcanza su mejor expresión. Se interpretó como equivalentes y coextensivas la función efectiva y el procedimiento efectivo, en relación a la función computable.

El procedimiento efectivo es la mejor expresión que elucida el algoritmo, como procedimiento automático, en cierto sentido relacionado a la axiomatización y deducción de teoremas de una teoría formalizada.

La tesis de Church-Turing, define como equivalentes los conceptos de función efectiva y el de procedimiento efectivo, resultando ser una definición básica, abriendo un campo de estudio en la denominada teoría de la recursividad, tratado en las matemáticas, específicamente en las metamatemáticas.

Sostenemos que la computabilidad se dedica al estudio de la realización de los cálculos numéricos, considerando aspectos relativos a la potencia y formas para formalizarla y aumentarla, en la medida que utilicemos dispositivos físicos, tal como se expresa en sentido figurativo, cuando hablamos de usar piedras para indicar operaciones de suma, resta u otro en relación a operar lógicamente dispositivos de naturaleza mecánica eléctrica logrando la materialización de las tecnologías de la información.

Jack Copeland en su artículo publicado en el 2004, en el libro editado por Luciano Floridi, *Philosophy of Computing and Information*, expresa sobre la importancia de la máquina de Turing con respecto al concepto de programa almacenado, así nos dice: "El principio básico de la computadora moderna, es la idea de controlar las operaciones de la máquina por medio de un programa de instrucciones de códigos almacenados en la memoria de la computadora que fue pensado por Alan Turing en 1935" (Copeland, 2004:3)

Al hablar de la computabilidad referimos al concepto de algoritmo como procedimiento que obtiene respuesta a un problema, mediante un proceso que tiene especificado los últimos detalles, las instrucciones son expresadas mediante un texto finito, y no es posible la actividad creadora en el momento de ejecución.

Si bien el concepto de procedimiento efectivo es matemático, resulta que los matemáticos cuando defienden un procedimiento, este no es necesariamente riguroso en cuanto a la precisión, se requiere de explicación, y si se habla de reglas estas no indican el orden en que deben ser ejecutadas. La máquina de Turing define un proceso, al que llamamos procedimiento efectivo, que contiene instrucciones elementales y precisas, de forma que resulta ser un programa, tal como hoy son los que se ejecutan en las computadoras, siendo en la máquina de Turing un código elemental.

La computabilidad en relación a las funciones computables, se utiliza en los números naturales, enteros y racionales, en contraste a los números reales, dado que en matemáticas, este conjunto de números es no numerable, a los que hemos referido como números inconmensurables, como lo indico Cantor[185].

En el concepto de algoritmo se encuentran el de computabilidad y otros, así lo manifiesta Hans Hermes[186]: "Del concepto de algoritmo podemos extraer toda una serie de importantes conceptos ulteriores. Algunos de esos conceptos son los de computabilidad, enumerabilidad y generalidad" (Hermes, 1984, 32)

---

[185] Conjunto numerable o contable, cuando sus elementos pueden ponerse en correspondencia a los números naturales. Cantor fue el primero que uso esta definición, tal como lo muestra en su artículo en 1874 *Über reine Eignschaft des inbegriffes aller reellen algebraischen* Zahlen Journal de Crelle 77, p258-262.

[186] Hans Hermes, nace el 12 de febrero de 1912 en Neunkichen Alemania y fallece el 10 de noviembre del 2003, matemático, físico, químico, biólogo y filósofo, profeso en Freburg y Múnich, estudioso de Frege, profesor de Mosterín en la universidad de Münster.

Para las funciones computables, suele definirse mediante un concepto constructivo, en el sentido que se dice que es computable si existe una máquina de Turing que la ejecuta. Es en este sentido que el concepto es constructivo, hay que presentar una máquina de Turing que ejecuta lo expresado en la función, específicamente con argumentos de números reales, así lo expresa Hans Hermes: "A pesar de la interpretación clásica del cuantificador existencial, la computabilidad de funciones reales se demostrara, en general, desde luego, constructivamente, suministrando un procedimiento de computación de forma explícita" (Hans Hermes, 1984:35).

Decimos que un conjunto es enumerable si los elementos se ponen en correspondencia 1 - 1 a los números naturales, es decir pueden ser ordenados a través de una relación que define una secuencia de los elementos del conjunto, así mismo decimos que un conjunto es numerable cuando es finito o es enumerable: "Un conjunto es numerable si es finito o biyectable con N. Todo conjunto enumerable es numerable, pero no a la inversa" (Mosterín, y Torretti, 2002: 191).

Mosterín y Torretti muestran como ejemplo que todas las funciones numéricas computables son numerables pero no son enumerables, para este fin definen una secuencia numerable de todas las funciones computables: $f_1(m)$, $f_2(m)$, $f_3(m)$,… $f_i(m)$. Se define una función computable $f^*(m)$, que tendría la siguiente definición: $f^*(m) = 1$ si $f_n(m) = 0$, y $f^*(m) = 0$ si $f_n(m) \neq 0$. Por lo tanto $f^*(m) \neq f_n(m)$, $\neq 0$. $f^*(m)$ es computable pero no podría estar en la secuencia de las funciones computables, entonces no es numerable.

Definimos decidibilidad en un conjunto si existe un procedimiento efectivo que determina si un elemento pertenece o no al conjunto: “Un conjunto es decidible si hay un algoritmo para decidir que objetos le pertenecen y cuáles no” (Mosterín y Torretti, 2002: 147)

La generabilidad la entendemos como la característica principal, aplicación o efectividad de todos o cada miembro de una categoría, en forma tal que sus elementos corresponden a lo especificado en un conjunto, así Hans Hermes nos lo indica a pie de nota: “En la literatura conjuntos generables se refieren con frecuencia a conjuntos generados” (Hans Hermes, 1984, 42).

Un ejemplo que ilustra la generabilidad, proporcionado por Hans Hermes, que trata sobre “El mínimo conjunto de números reales que contienen los números 9 y 12 y está clausurado con respecto a la sustracción y a la multiplicación por $\sqrt{2}$ puede obtenerse con el siguiente algoritmo: (a) 9 (b) 12 (c) x – y (d) x $\sqrt{2}$ ” (Hans Hermes, 1984, 28).

Luego nos muestra dos números generados en el algoritmo mencionado, así nos dice: “Los ejemplos (1) y (2)… muestran la generabilidad del módulo generado por 3 y $3\sqrt{2}$ ” (Hans Hermes, 1984,42). Resulta adecuado el ejemplo, dado que nos muestra una definición del tipo constructiva de formación de números, esta se define mediante un proceso (algoritmo), 3 se obtiene del resultado de 12 – 9 aplicando las reglas (a), (b) y (c), y el resultado $3\sqrt{2}$ se obtiene aplicando la regla (d).

Consideramos adecuado el concepto de algoritmo dado que está contenido en los conceptos de computabilidad, enumerabilidad, decidibilidad y generabilidad, tal como lo hemos mencionado, con el añadido que la computabilidad no está referida únicamente al conjunto de los números naturales, más bien transciende estos límites, conteniendo aspectos referidos a operar dispositivos físicos, así mismo no se pueden ignorar lo que se entiende como lo no computable.

Explicamos en el capítulo anterior que Turing no precisó sobre el mecanismo del funcionamiento de los componentes de su máquina, así mismo mencionamos de que en su documento de 1936, refirió a su máquina como una definición matemática, y en su documento de 1938, le incorporó un dispositivo al que denominó oráculo, que resulta ser una máquina con más potencia, que tiene la capacidad de saber si un problema es o no computable. En este tema la computabilidad incorpora aspectos que no se explican en las funciones recursivas generales, así mismo deja espacio a la comprensión de la complejidad de la computabilidad, que incluye aspectos mencionados como complejidad de tiempo y de espacio en el uso de las computadoras.

4.5. Complejidad computacional.

Cuando Turing define la máquina abstracta en 1936, ésta tiene una memoria en forma de una cinta, que no tiene fin en ambos sentidos, es de longitud infinita hacia la izquierda y a la derecha, por lo que se entiende que la máquina tiene suficiente memoria, aunque esta no se utilice plenamente. También observamos que cuando las máquinas ejecutan una secuencia de instrucciones disponen de todo el tiempo, no hay restricción al respecto, no es parte de la definición de la máquina, solo interesa si la máquina termina o no la ejecución del proceso.

En los textos de la ciencia de la computación que tratan sobre la complejidad muestran dos orientaciones de investigación, una relacionada a la cantidad de tiempo exigido en la computación de un programa, en cuánto y cómo se comporta el tiempo de la ejecución, por otro lado el estudio del tamaño de un programa en cuanto a la cantidad de código requerido para resolver determinado problema a computar.

En ambos casos (complejidad de tiempo y de espacio) fueron "ignorados" por Turing, dado que consideramos que era consciente de estas dificultades, había que dejar de lado para lograr la precisión en la definición alcanzada, expresada en la abstracción de la máquina expresada en la memoria de la cinta y del tiempo sin límites.

En el caso, de la complejidad en el tiempo, se investiga la duración de la ejecución de un programa de computadora, y como se afecta, en la medida que se trata de valores que aumentan cuando el argumento también aumenta. Dado que una función recursiva es una definición de sí misma, el tiempo requerido para calcular un valor es el tiempo para el cálculo del anterior argumento más un tiempo adicional. El comportamiento del tiempo para el cálculo mediante las funciones recursivas se comporta como una función polinómica[187], por esta razón se les denomina problemas de tipo P, siempre que se ejecutan en una máquina determinista[188] de Turing, y se dice que son problemas del tipo NP, en los casos que se ejecutan en una máquina de Turing no determinístico[189].

En cuanto a la complejidad de espacio, se considera el funcionamiento de un

---

[187] Polinomios expresiones algebraicas de la forma $P(x)= a^n x^n + a^{n-1} x^{n-1} + \ldots + a^1 x + a^o$.

[188] Una máquina de Turing determinístico, es aquella en la que la función de transformación de los estados, está precisamente definida, es decir cuando la máquina de Turing está en un estado y lee un símbolo de la cinta, pasa a otro estado y se desplaza en la cinta.

[189] Una máquina de Turing no determinístico es aquella en la que la función de transformación de los estados se da el caso que se puede pasar a un estado u otro para una situación determinada, tiene más de una alternativa.

programa en la computadora en relación a los recursos de memoria, esta forma de plantear el problema estudia el comportamiento de la duración de ejecución, se dice que corresponde a una complejidad del tipo L, que está incluida en los problemas tipo P, lo mismo con la complejidad de espacio no determinístico, del tipo NL.

Otra forma de plantear el caso de la complejidad computacional, es estudiando la variable del tamaño del programa de computadora, desligándose en cierto sentido del concepto de la recursividad, incorporando en la investigación el tamaño de memoria requerida para el funcionamiento de un programa, en esta orientación encontramos los estudios de Kolmogorov[190].

Mosterín refiere a la complejidad de Kolmogorov, en el sentido de que la define en relación al tamaño de lo que se está describiendo, incluyendo en la definición la regularidad hasta su ausencia, de tal forma que el resultando es más simple en una descripción de una secuencia de caracteres cuando hay regularidad, ocurriendo lo contrario cuando no hay regularidad en lo que se describe.

> En 1965 Andrei Kolmogorov introdujo de un modo preciso la noción de complejidad - llamada actualmente en su honor complejidad de Kolmogorov o K - como una medida de la información individual o aleatoriedad de las secuencias (Mosterín y Torretti, 2002:97).

Gregori Chaitin[191], afirma que Turing descubre la no computabilidad y refiere a que Hilbert insistió en la existencia de un procedimiento mecánico que decidiese si la

[190] Andrei Kolmogorov, (25 de abril de 1903 en Moscú, 20 de octubre de 1987), matemático ruso. Obtuvo su doctorado en la universidad estatal de Moscú en 1929, hizo importantes progresos en topología, estadística, fundador de la teoría de la complejidad algorítmica.
[191] Gregori J.Chaitin, nace en 1947 en Nueva York. De padres argentinos. Estudia matemática en la universidad de Buenos Aires. Trabaja en el centro de investigación J. Watson de IBM en York town, Heights New York. Trabaja en la teoría algorítmica. Ha publicado diversos artículos y libros sobre temas de la ciencia de la computación.

demostración se ajusta a reglas especificadas, Hilbert nunca aclaró que significó el procedimiento mecánico (Chaitin 2003). En este sentido se expresa la complejidad, en la medida de precisar el procedimiento, que contenga aspectos aun no estudiados a profundidad, como el de la aleatoriedad y la no computabilidad.

En los años 1950, Von Neumann resalta la importancia de la complejidad temporal de los cálculos, demandando en la búsqueda de la mejora de la potencia de las computadoras en relación a disminuir el tiempo de ejecución de cálculos matemáticos, agregando al tema de la computabilidad los componentes físicos.

Chaitin indica sobre al problema de complejidad de tamaño de un programa, no le importa cuán rápido se efectúe un programa. Explica que la complejidad se encuentra en relación a la cantidad de información que está contenida en las instrucciones y datos que el programa efectuará.

> Es una teoría del largo del programa, del tamaño del programa, no del tiempo de su corrida; no importa si es un programa rápido o no. Se mide por la complejidad de H de una tarea de cómputo por la cantidad de información que es necesario tener en el programa. La mínima cantidad de información que se requiere (Chaitin, 2002:56).

Dentro del concepto de complejidad de la computación encontramos el caso de la aleatoriedad. Ésta se manifiesta en los diversos programas como el de la generación de números aleatorios, utilizados en diversos procesos automáticos, y sucede que en su mayoría son pseudogeneradores, porque estos programas empiezan con un valor que suele llamarse “semilla” y a partir de ésta se obtienen los siguientes números. Se define un intervalo de números, cuando se repite el valor de la semilla. Una solución al caso es implementada mediante un componente físico de generación del número aleatorio que es un circuito basado en el ruido que produce un diodo (este mecanismo tiene la naturaleza de ser aleatorio).

La complejidad computacional requiere ser formalizada, así para Chaitin refiere a que Turing y Church definen cada uno un lenguaje de programación: Turing utiliza un lenguaje de máquina, mientras que Church utiliza un lenguaje de alto nivel comparado al lenguaje LISP en relación al cálculo lambda de las funciones recursivas: “El de Turing no es un lenguaje de alto nivel, como el LISP; se trata más bien de lenguaje de máquina. El código está formado por unos y ceros que se le suministra al procesador central de un ordenador” (Chaitin, 2003:32).

Chaitin compara el cálculo lambda al lenguaje de programación LISP, indicando que le corresponde a un lenguaje de programación de computadoras de alto nivel: “Por ejemplo el cálculo lambda de Church es un idioma de programación… Tiene funciones recursivas, y a mi juicio es muy parecido a LISP” (Chaitin, 2002:72).

En la forma de plantear la complejidad de la computación mediante la definición de la máquina de Turing con Oráculo, se han conjeturado dos opciones: Una que dice que es posible construir una computadora con el dispositivo Oráculo con una nueva física, como lo sostiene Roger Penrose, y la otra de que la máquina de Turing contiene la capacidad de cómputo suficiente, expresada por Stephen Hawking[192], indicando que si ya simulamos en una computadora el funcionamiento del cerebro de una lombriz no hay

---

[192] Stephen William Hawking, nace en Oxford el 8 de enero de 1942. Físico miembro de la Real Sociedad de Londres. Titular de la cátedra Lucasiana de Matemáticas en la universidad de Cambridge. Ya jubilado en el 2009. Se doctora en física en 1966 en Oxford, en 1979 es nombrado catedrático lucasiano. Investigador, ideó nuevas técnicas matemáticas en estudios de la relatividad, los agujeros negros, descubriendo que estos emiten radiación. También sustenta que después del Big bang, se crean objetos supermasivos. Tiene varias publicaciones y libros.

ninguna diferencia para lograr simular el funcionamiento de un cerebro humano. (Penrose, Shomony, Cartwright y Hawking, 1999:96).

Encontramos en la teoría de Turing los estudios que busca el entendimiento del funcionamiento del cerebro humano, considerando que en la medida que comprendamos sobre su funcionamiento, se podrá formular algoritmos que permitan simular diversos aspectos del cerebro humano.

4.6. Paso a Paso.

En la búsqueda de lo que es computable, se efectuaron diversas investigaciones, entre estas, se encuentra la investigación de Church, lo afirma Sieg Wilfried, de esta forma:

> La Tesis Church, por ejemplo, expresa en su forma original el cálculo efectivo de funciones de número-teórico son exactamente las funciones cuyo valor son computables mediante el cálculo de ecuaciones de Gödel, es decir, funciones recursivas generales (Sieg Wilfried, 2004:4)

La investigación efectuada por Turing, en 1936, tuvo la misma referencia a la investigación realizada por Church, en cuanto a la ejecución de una función computable, pero en forma distinta, ya que al primero refiere al procedimiento efectivo que ejecuta un cálculo, que sigue instrucciones, recogiendo el sentido de una ejecución de tipo paso a paso.

Wittgenstein reconoce el sentido del concepto de máquina de Turing, en las instrucciones que constituyen los procedimientos para ser ejecutados en la máquina, que pueden ser seguidos por un hombre, por lo tanto la máquina actúa como si fuera el hombre. Sieg considera su importancia epistemológica: “El mensaje breve de Wittgenstein sobre las máquinas de Turing ‘estas máquinas son humanos que calculan’ captura la característica del análisis de Turing de la calculabilidad que la hace de esta epistemológicamente relevante” (Sieg Wilfried, 2004:7).

Turing verso su investigación sobre los números computables con aplicación al *Entscheidungsproblem*, siendo equivalente al resultado de la investigación de Church, tal como lo indicó en una nota al final de su trabajo, tanto así, que retraso su publicación en consideración a la publicación de Church, pero Gödel pensó en forma distinta, expresando que el procedimiento mecánico definido por Turing precisa la definición de un sistema formal, en relación al concepto de algoritmo. Al respecto, Sieg Wilfried indica:

> Gödel subrayó la importancia del análisis de Turing, repetida y enfáticamente. Afirmó en 1964, que sólo el trabajo de Turing proporciona "una precisa, sin duda y adecuada definición del concepto general de sistema formal". Ya que un sistema formal para Gödel, es sólo un procedimiento mecánico para producción de teoremas, la adecuación de esta definición recae perfectamente sobre el correcto análisis de Turing de los procedimientos mecánicos (Sieg Wilfried, 2004:37)

De otro lado, Church consideró que el cálculo efectivo mediante funciones recursivas expresan el concepto de calculabilidad, y por ser así la recursividad es un método que en su expresión general no lo supera ningún otro, lo resalta Sieg de la siguiente forma:

> Para dar un análisis más profundo Church señaló, en la sección 7 de su documento dos métodos para caracterizar el cálculo efectivo de la función del número-teórico que se sugieren. El primero de estos métodos usa el concepto de "algoritmo", y el segundo emplea la noción de "calculabilidad en una lógica". Él sostiene que ningún método da lugar a una definición más general que la recursividad. (Sieg Wilfried, 2004:45)

Para Church el concepto de cálculo efectivo en relación a las funciones recursivas, son evidentes como funciones calculables y que éstas son un procedimiento del tipo paso a paso. Es en este sentido en que Church considera que el cálculo lambda expresa adecuadamente la noción intuitiva de algoritmo. Sieg nos dice que Church tiene razones para justificar su tesis, la primera es la observación casi empírica en la que todas las funciones computables pueden ser mostradas como recursivas y que son equivalentes al argumento de paso a paso. (Sieg Wilfried, 2004).

El concepto de paso a paso, expresado por Church y el de Turing, fue considerada por una parte de la comunidad científica, como complementaria y por otros como coextensivas, razón por la cual se entiende la tesis Church-Turing: La recursividad de las funciones computables es equivalente al de procedimiento efectivo, pero es en la máquina de Turing donde se expresa de forma más adecuada el concepto de paso a paso, en la medida de que se sigue en forma mecánica la ejecución de operaciones elementales, mientras que en las funciones recursivas, se requiere del entendimiento o de una semántica del concepto de función. Sieg enuncia sobre la recursividad como una profundización de las operaciones mecánicas, pero es Turing quien logra el concepto del paso a paso:

> Al examinar el análisis y recursividad de Turing, encontraremos la clave para responder a la pregunta que he planteado en la diferencia entre las propuestas de Church y la de Turing. Muy brevemente: Turing profundizo el argumento de paso a paso de Church, en operaciones mecánicas subyacentes a los pasos elementales, mediante la adecuada formulación de las limitaciones que garantizan su recursividad (Sieg Wilfried, 2004:60-61)

Church consideraba que las funciones recursivas eran la mejor forma de obtener un cálculo efectivo de funciones, específicamente el cálculo lambda, y lo que hizo Turing fue el precisar los pasos elementales en el concepto paso a paso. Church consideró que el cálculo efectivo se formula mediante la función recursiva: “Church propuso en una reunión de la American Mathematical Society en abril de 1935, que la noción de una manera efectiva de calcular la función de enteros positivos debía ser identificado con la de una función recursiva.". (Sieg Wilfried, 1992:1).

El concepto del paso a paso se expresa adecuadamente en el procedimiento efectivo tal como ocurre en la secuencia de instrucciones que se ejecutan en una máquina de Turing, en clara diferencia a lo propuesto en el de cálculo lambda, dado que es menos expresiva en relación al de paso a paso.

En la ejecución de un programa de computadora, se manifiesta el paso a paso, dado que dependen de la lógica de la arquitectura del equipo, en la que todos los dispositivos de la máquina funcionan de manera discreta, en la que ejecuta una instrucción por vez, incluso en las computadoras que tienen varios microprocesadores[193], estos mantienen la forma de ejecución discreta, compartiendo los procesadores según sea el caso[194].

4.7. Máquina de lápiz y papel.

Turing cuando define el concepto de máquina, refiere a un procedimiento que será ejecutado por una persona, siguiendo instrucciones, y utilizando un lápiz y un papel. Esta idea está presente en el método utilizado por Turing y que es central en la sustentación de la presente tesis, porque proporciona el alcance del concepto utilizado por Turing. El procedimiento efectivo refiere a un proceso mecánico, con pasos básicos, en la medida que

[193] El microprocesador es el circuito integrado de la computadora que ejecuta el programa, este ha ido evolucionando en el tiempo, adquiriendo constantes mejoras, Core Duo es un microprocesador con dos núcleos, a la fecha se están trabajando procesadores de múltiples núcleos (permiten procesos paralelos), ya se anunció el microprocesador de 8 núcleos.

[194] Tenemos entendido que los multiprocesadores son utilizados parcialmente, los programas de computadora aprovechan las cualidades de los procesos paralelos en el computador, el mejor uso corresponde a diversos procesos como en los graficadores de pantalla.

utiliza dispositivos de una máquina.

El concepto de máquina y las ideas relacionadas modelaron una concepción en las matemáticas, contribuyendo en el campo matemático denominado finitismo, al respecto Sieg acota:

> ¿Qué es exactamente finitismo? por contraste sigue abierto. La primera cuestión no fue obtenida a lo largo de línea de la Gödelización por generalización de la recursión, pero por un enfoque muy diferente debido a Alan Turing, y en cierta medida Emil Post. Se centraron en los procesos simbólicos subyacentes de los cálculos numéricos en lugar de los propios cálculos. Esto dio lugar a los fundamentos de la enseñanza teórica, a través de la máquina universal de Turing, también debido a la práctica de la informática. En consecuencia, surgieron las bases de lo que ese tiempo eran cuestiones oscuras casi filosóficas. (Sieg Wilfried, 2005:138)

El resultado obtenido mediante funciones efectivamente calculables es equivalente al que se obtiene en la máquina de Turing, pero la sintaxis utilizada es diferente, porque en la ejecución de las funciones operan números enteros, mientras en la máquina de Turing, la ejecución es sobre símbolos que semánticamente expresan números naturales, es en este sentido en que la tesis Church es igual a lo propuesto en la máquina de Turing.

> La tesis de Turing, sugiere la identificación de funciones efectivamente calculables con funciones cuyos valores pueden ser calculadas por un dispositivo de computación idealizada, una máquina de Turing. Como los dos conceptos matemáticos son probadamente equivalentes, las tesis son "equivalentes", y están agrupadas en la denominación tesis de Church-Turing. (Sieg Wilfried, 1997:1)

La computadora es en esencia una máquina de Turing universal, las funciones calculables son resolubles en la computadora. Lo expresado es una forma de la tesis de Church-Turing. Pero en el caso de la máquina de Turing el algoritmo es un procedimiento que tiene la particularidad de ser una secuencia precisa de pasos, en la que Turing la expresa como comparación con una máquina de lápiz y papel.

El planteamiento de la similitud entre la máquina de Turing y la máquina de lápiz y de papel con la presencia humana que ejecuta las instrucciones, resultando una relación entre el hombre que ejecuta un procedimiento con el de algoritmo, así nos dice Sieg: "Turing…, subrayó en su documento de 1953. Precisando que el concepto (recursividad, Turing computabilidad) va a la captura de procesos mecánicos que pueden ser llevados a cabo por los seres humanos" (Sieg Wilfried, 1997:7).

Establecemos que Turing maneja un concepto de computabilidad en correspondencia a la ejecución mediante lápiz y papel y que una persona siga instrucciones. En palabras de Soare:

> En 1935 Turing y todos los demás utilizaron el término "computadora" para una idealización del cálculo de un humano con material extra como el lápiz y papel, calculadora o un escritorio, siendo un significado muy diferente a la utilización de la palabra al día de hoy (Soare Robert, 1996: 9)

De otro lado, para operar funciones mediante el cálculo lambda, se requiere del conocimiento de la sintaxis y la semántica de las funciones recursivas, específicamente cuando asociamos representaciones numéricas en relación a cálculos sofisticados, resultando de un nivel mayor a las formas especificadas en la máquina de Turing.

4.8 Recursividad e inducción matemática.

La función recursiva está relacionada a la inducción matemática, utilizada por Peano en la axiomatización de los números naturales, específicamente en la definición de un número como el número anterior más uno.

Church asocia la recursividad con el cálculo lambda, definiendo un tipo de proceso mediante la composición de funciones. Soare considera que la teoría de la computabilidad trata de los procesos efectivos como que con recurrentes, en relación a expresiones recursivas, afirma que el concepto de la recursividad reduce la computabilidad, así tenemos:

> Acerca de la preservación de la diferencias intencionales entre los conceptos de "computabilidad" y "recursividad" específicamente se recomienda que: El término "recurrente" ya no debe llevar el significado adicional de "computable" o función "decidible" definidas utilizando máquinas de Turing máquina de registro, o sus variantes debe ser llamado "computables" en lugar de "recursiva", hay que distinguir la diferencia intencionales entre la tesis de Church y la tesis de Turing, y utilizar este último en particular en cuestiones relativas a la mecánica, el nombre del tema debería ser "teoría de la computabilidad" o simplemente computabilidad en vez de "teoría de la función recursiva" (Soare Robert, 1996,1)

La computabilidad tiene un significado mayor al de función computable referida a funciones recursivas en los números enteros. La computabilidad en el procedimiento constituido por instrucciones elementales, permite el cálculo de relaciones, que como concepto matemático contiene a las funciones, que a su vez más amplio que el de funciones recursivas.

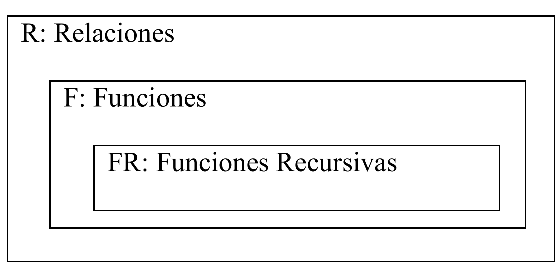

Figura 10. Jerarquía en las funciones recursivas, funciones y relaciones.

La máquina de Turing opera signos, que no son números, carece de toda significación, ejecuta instrucciones en un orden establecido, y los resultados no le dicen nada, lo único que determina es que ha ejecutado, manifestado en el hecho de detenerse, mientras que en el cálculo de las funciones recursivas, se requiere de interpretación y jerarquización para la valuación de la función. Razón por la cual sostenemos que la teoría de la computabilidad es de diferente significado al de la teoría de la recursividad, Soare propone desligar ambos conceptos, así nos dice: “Algunos lógicos han señalado que es curioso que el tema de la teoría computabilidad (en adelante llamada "la materia") sea llamada "teoría de la función recursiva" o "teoría de la recursividad" en lugar de lo que sería más natural "teoría de la computabilidad"” (Soare Robert, 1996:2)

Church y Kleene consideraron que el cálculo lambda expresa adecuadamente las funciones recursivas generales, con relación al cálculo efectivo. Pero Gödel tenía dudas sobre la veracidad del concepto, llegando a calificar lo afirmado por Church como “totalmente insatisfactoria”, así lo menciona Robert Soare sobre la conversación sostenida por Church y Gödel (hay que considerar que ambos frecuentaban la misma facultad de matemáticas de la universidad de Princeton), así nos dice Soare:

> En 1930 Church había estado estudiando una clase de funciones de cálculo efectivo llamadas funciones lambda-definible. El alumno de Church, Kleene demostró en 1933 que un gran número de clases de funciones teóricas eran lambda-definible. Con la fuerza de esta evidencia, Church propuso a Gödel alrededor de marzo de 1934 que la noción de "efectivamente calculable" se identifica con "lambda-definible" una sugerencia que Gödel, rechazó como "totalmente insatisfactoria" (Soare Robert, 1996:7)

Tanto Church como Kleene consideraron que la recursividad expresa el concepto de computabilidad, ampliándolo al manejo de símbolos como la Forma Normal[195] de Kleene, que es una formulación de cadenas de caracteres, que tienen una notación recursiva.

Robert Soare sostiene que en ningún momento Church demuestra que el procedimiento efectivo definido por Turing son funciones recursivas, encontrándose una limitación conceptual, a la consideración de que la recursividad de funciones son expresables desde funciones recursivas iníciales, tal como se muestra en el contra ejemplo de la función recursiva de Ackermann (mencionada en el segundo capítulo), que trata de una función recursiva que no es posible ser expresada mediante funciones recursivas iniciales. Soare menciona que lo dicho por Church y Kleene son una definición, por esto, Gödel mostro sus dudas.

> Si los pasos básicos son etapas recursivas, entonces es fácil seguir por el teorema de la Forma Normal demostrado por Kleene, que probó que todo proceso es recursivo, comunicándolo a Gödel antes de noviembre de 1935. La debilidad fatal en el argumento de Church fue el supuesto básico de que los pasos atómicos son etapas recursivas, algo que no justificó. (Soare Robert, 1996:8)

Consideramos que al haberse planteado la tesis de Church-Turing e interpretado que define la computabilidad, reduce el significado, desvaneciendo las diferencias que tienen la recursividad y el procedimiento efectivo, haciéndolas complementarias (en el mejor sentido). Robert Soare, manifiesta en este sentido sobre la tesis de Church-Turing: “En 1952 Kleene se refiere a "la tesis de Church" y "tesis de Turing" lo que es aún más curioso es que la frase "tesis de Church" también llegó a denotar la "tesis de Turing" y quizás bien expresado para otros, pero borrando toda intencional distinción” (Soare Robert, 1996:14)

---

[195] Forma Normal de Kleene refiere a que una cadena de caracteres pueden ser expresadas con operaciones recursivas así tenemos a+a+a+… a =a*

Destacamos que el concepto de la máquina de Turing en relación a las computadoras era un concepto temprano, en 1936 no había computadoras. La definición de Turing se acepta en correspondencia a la función recursiva y función computable. Así nos dice Robert Soare, sobre la aceptación de la recursividad acerca de la máquina de Turing.

> Funciones recursivas establecen una tradición matemática y había más un llamado a una audiencia general de matemática. Church estaba dispuesto a establecer. La lógica matemática como un campo de las matemáticas y para evitar las inconsistencias descubiertas por Kleene y Rosser en general en el sistema lambda de Church... el formalismo de Turing habría tenido poca aceptación de un público general, porque en fecha tan tardía como 1946 el término "computadora" significaba un humano que calcula con lápiz y papel (Soare Robert, 1996:33)

Lo expresado está en relación al contexto en el que se desarrolla la discusión en la comunidad científica, sobre el cálculo efectivo mediante la función recursiva según el cálculo lambda, y es así como finalmente se conoce en la mayoría de los textos de computación, estableciéndose la teoría de la recursividad, en referencia al de función computable.

Con fines de acentuar lo expresado, Church y Kleene utilizan el término de recursividad, que resultó entendible por la comunidad de matemáticos, anteladamente al de cálculo lambda, aunque siempre estuvo presente en forma implícita.

> En 1935 y 1936 Church y Kleene cambian su formalismo y la terminología de funciones lambda-definible (en general) por recursivas porque esta última era más aparente a la audiencia y era más una tradición establecida y el público reconocía esta terminología (Soare Robert, 1996:35)

Según Robert Soare, Church utiliza el término recursivo como adverbio[196] en relación al término computable en relación a la teoría de la función recursiva (definida por Kleene). Ambos (Church y Kleene) estaban convencidos de que expresaban en mejor sentido el concepto de computabilidad, pero ocurrió, que en ningún momento Gödel ni Turing utilizaron el concepto de computable en alusión a las funciones recursivas.

> Church introdujo el uso de "recursivo" como un adverbio que significa "computable" por ejemplo, "Recursivamente enumerables" y más tarde Kleene introdujo el término teoría de la función "recursiva"… Gödel y Turing nunca utilizaron el término "recursivo" en el sentido computable y explícitamente rechazaron tales sugerencias (Soare Robert, 2007:2)

Gödel adiciona en 1963, una nota suplementaria al final de su trabajo sobre sentencia formalmente indecidibles que fuera escrita en 1931, en la que indica que el trabajo de Turing de 1936, expresa un sistema formal en forma precisa y adecuada en relación a sus teoremas VI y XI en la que hay sentencias aritméticas indecidibles y que no pueden probar su consistencia en el sistema. Para Gödel un sistema formal está relacionado al concepto de máquina de Turing.

> Como consecuencia de avances posteriores, en particular del hecho de que gracias a la obra de A.M. Turing ahora disponemos de una definición precisa e indudablemente adecuada de la noción general de sistema formal, ahora es posible dar una versión completamente general de los teoremas VI y XI. Es decir, se puede probar rigurosamente que en cada sistema formal consistente que contenga una cierta porción de teoría finitaria de números hay sentencia aritméticas indecidibles y que, además, la consistencia de cualquiera de esos sistemas no puede ser probada en el sistema mismo. (Gödel, en Mosterín 2006:87)

En 1964 Gödel proporciona a Martin Davis diversas notas suplementarias para la reimpresión de algunos de sus trabajos. En opinión de Jesús Mosterín, había una importante posdata en la que Gödel propone identificar la noción de sistema formal con la

[196] Adverbio, es una parte invariable de la oración que califica o determina la significación del verbo, del adjetivo y veces de otro adverbio. Los adverbios sirven para indicar circunstancias del verbo.

de máquina de Turing: "… lo que equivale a identificar teoría formalizada con conjunto recursivamente enumerable- es decir, generable por una máquina de Turing" (Gödel en Mosterín, 2006:166)

Gödel dice que la obra de Turing, permite una definición precisa, adecuada e incuestionable de concepto general de sistema formal, en relación al procedimiento mecánico.

> La obra de Turing proporciona un análisis del concepto de <<procesamiento mecánico>> (<<algoritmo>>, <<procedimiento computacional>> o <<procedimiento combinatorio finito>>). Se ha probado que este concepto es equivalente al de <<Máquina de Turing>>. Puede definirse un sistema formal simplemente como un procedimiento mecánico para producir filas de signos, llamadas formulas deducibles. (Gödel, en Mosterín 2006:197)

De otro lado, Gödel discrepa con Turing con respecto a ideas sobre la mente, específicamente en lo referente a la inteligencia en las máquinas, que son expresados por Turing con respecto al de procedimiento mecánico, así lo hace saber: "Téngase en cuenta que los resultados mencionados en esta posdata no establecen límites de la capacidad de la razón humana, sino más bien de la posibilidades del puro formalismo en matemáticas" (Gödel, en Mosterín 2006:197).

La diferencias entre Gödel y Turing, radica en la idea de Turing en equiparar en cierto sentido la mente humana con una máquina de Turing, Gödel refiere que se estaría limitando a la mente humana, así lo expresa Hao Wang[197] en su libro *A Logical Journey*

[197] Hao Wang, nacido el 20 de Mayo de 1921 en Jinan China y fallece el 13 de mayo de 1995 en EEUU, Chino americano, lógico, filósofo y matemático. Obtiene su maestría en matemáticas en la universidad de Tsinghua en 1945, luego viaja a EEUU, se doctora en lógica en la universidad de Harvard en 1948. En 1972 se une al primer grupo de chinos americanos que vistan China. Estudioso de la demostración de teoremas lógicos utilizando computadoras. En 1983 recibe el premio Milestone para automatizado de demostraciones de teoremas otorgado por International Join Conference on Artificial Intelligence.

from Gödel to Philosophy: “Con respecto a la cuestión central de superioridad de la mente sobre las computadoras, Gödel nota el error en la filosofía de Turing” (Hao Wang; 1974:202).

Gödel creía en la importancia de la definición formal, como procedimiento, en la que podía ser una definición del tipo heurístico. En enero de 1936, expresó algunas dudas sobre si el "procedimiento finito" podría ser analizado en todo o si sólo debía servir "como un principio heurístico" (Soare Robert, 2007:7)

Consideramos que se han producido diversas interpretaciones de la tesis de Church, desde la original basada en el cálculo lambda hasta la recursividad en relación a la máquina de Turing, así lo dice Soare: “Tesis de Church (en su primera versión) (1934) Una función es efectivamente calculable sólo si es lambda-definible” (Soare Robert, 2007:9), luego Church la modifica a “Tesis de Church (1936) Una función de enteros positivos es efectivamente calculable si y sólo si es recursiva” (Soare Robert, 2007:10), expresada luego en relaciona la máquina de Turing como: “La tesis de Turing (1936) Una función es intuitivamente computable (efectivamente calculable) si y sólo si es computable por una máquina de Turing, es decir, una máquina automática (una máquina), tal como se definen en Turing (1936)” (Soare Robert, 2007:12).

Si tomamos en cuenta el concepto intuitivo de algoritmo en relación al procedimiento efectivo, entendemos que lo computable está en relación a un algoritmo, como procedimiento, que significa un programa de computadora, que en su ejecución no es sólo recursivo. Soare precisa que el significado de función recursiva ya tiene nombre y resulta ser el de inducción matemática.

> Cuando se utiliza un término como "recursivo" que también significa "computable" o "algoritmo" como Kleene lo hizo, entonces uno nunca está seguro de si en un caso particular significa "calculable" o "inductivo" y nuestro lenguaje tiende a convertirse indistinto. Retomando "recursivo" a su significado original de "inductivo" ha hecho su uso mucho más claro. No necesitamos otra palabra que signifique "computable". Ya tenemos una (Soare Robert, 2007:48)

Para la evaluación de un determinado valor en una función según la inducción matemática, se requiere calcular la función para el valor anterior, de tal manera que resulta ser una computación de arriba hacia abajo para luego regresar a completar los cálculos, siendo lo mencionado una condición necesaria en la ejecución de un cálculo mediante un proceso recursivo.

4.9. Interacción y comunicación.

Existen diversas características que describen el funcionamiento de los programas en las computadoras, que producen interpretaciones no convencionales con respecto a la tesis de Church-Turing, entre uno de estos se refiere a la condición necesaria de interacción entre los diversos procesos que se dan en la máquina y otro es el de la comunicación entre las máquinas, inclusive aquella que refiere a la necesaria interacción entre el hombre y la máquina. En el caso de las computadoras como equipos, desde el inicio ha estado presente el estudio del lenguaje, como forma de comunicación y de interacción entre los diferentes componentes internos o externos de la máquina, así tenemos por ejemplo la opinión de Norbert Wiener, que expresa que este tema preocupó a la comunidad científica desde épocas anteriores a Turing, en referencia a Leibnitz.

Cabe resaltar que el lenguaje no es un atributo exclusivo de los seres vivientes, así mencionamos de máquinas que requieren signos como formas de comunicación, y precisamente la necesidad del hombre de operar estas máquinas, se requiere trasmitir órdenes, en el sentido expresado por Norbert Wiener.

Conocemos los trabajos de investigación de Norbert Wiener, en el estudio de diversos mecanismos biológicos y naturales en lo que denominó Cibernética, resaltó el tema del lenguaje como forma de comunicación. Para Wiener es posible la comunicación de hombre a máquina, incluso de máquina a máquina: "Generalmente, al pensar en las comunicaciones, suponemos que se efectúan de persona a persona. Sin embargo, es posible que un hombre hable a una máquina, o está a un ser humano, o un aparato a otro" (Norbert Wiener, 1958:71)

Norbert Wiener con el mismo criterio de Turing, define la posibilidad de que las computadoras (que ya estaban presentes en 1958) podían jugar ajedrez con un hombre, encontrándose de acuerdo con la tesis de Turing expuesta en 1950 sobre la posibilidad de que las máquinas imiten el juego del hombre. Wiener expresa su percepción del futuro incluyendo opiniones de von Neumann y Claude Shannon, indicándonos:

> Hace algún tiempo, sugerí un modo de utilizar una de las modernas máquinas de calcular para jugar una partida de ajedrez bastante pasable… Es fácil construir una que juegue pobremente, de acuerdo con las reglas. Es una tarea sin esperanza construir una que lo haga perfectamente, pues requeriría demasiadas combinaciones. El profesor Johann von Neumann... ha comentado esta dificultad. Sin embargo, no es fácil ni imposible construir una que jugara lo mejor posible para un número limitado de jugadas… Coincido con Shannon en que una máquina de esa clase jugaría ajedrez como un aficionado muy bueno y hasta posiblemente como un maestro (Norbert Wiener, 1958:164-165)

Dina Goldin de la universidad de Connecticut y Peter Wegner de la universidad de Brown han escrito diversos artículos[198] referidos a la tesis de Church-Turing, formulando

[198] Citamos algunos de los artículos publicados en 2003, 2005 y 2007.

observaciones en el sentido de que la tesis no explica adecuadamente los procesos interactivos que hoy se dan en las redes de computadoras que incluyen el tema de comunicación. Así también las características de funcionamiento de los programas con respecto a las pantallas, porque en éstas se presentan pantallazos que exigen una respuesta y según, la máquina va mostrando el avance en nuevos pantallazos, esto se realiza dinámicamente en forma secuencial, denominándolos interacción.

El argumento que tienen Goldin y Wegner es una crítica a la tesis de Church-Turing en el sentido que el concepto de computabilidad en correspondencia a la función recursiva, ignora la comunicación y la interacción, y además no son bien identificadas mediante funciones, así nos dicen: "En efecto la tesis fuerte de Church-Turing es incorrecta, el comportamiento de algoritmos basadas en funciones no captura todas las formas de computación" (Goldin Dina y Wegner Peter, 2005: 2)

Ambos presentan un ejemplo que trata de un vehículo que debe ir de un lugar hacia otro, y manifiestan que si éste fuera representado únicamente mediante una función, el vehículo se dirigiría directamente al lugar de destino, sin tomar en cuenta las diferentes circunstancias que ocurren en el camino, como por ejemplo: Los semáforos, el tráfico o las personas que están en el camino, concluyendo que no podría llegar al lugar de destino, porque se quedaría en el camino producto de alguna circunstancia, consecuentemente no podría ser una función la que exprese adecuadamente esta situación, más bien, debe ser un proceso interactivo que va evaluando en cada momento las circunstancias que se presentan en el camino, así nos dicen:

> En todo caso el problema es computable por un mecanismo de control, como en un carro robótico, que continuamente recibe como entrada el video del camino y actúa en los frenos de las ruedas…

> La computación realizada por el carro automático y sistema operativo es interactivo, donde la entrada y salida ocurren durante el cálculo, no antes o después de ella. (Goldin Dina y Wegner Peter, 2005: 3)

En un artículo de Eugene Eberbach, Dina Goldin y de Peter Wegner sobre: Las ideas de Turing y modelos de computación, definen modelos computacionales que dicen que son más potentes al de la máquina de Turing, a pesar que reconocen de las capacidades de los recursos (espacio y tiempo) en la máquina de Turing: "El modelo de máquina de Turing puede ser extendido por eliminar a priori los límites de sus recursos, que puedan resultar en: una configuración inicial infinita, una arquitectura infinita, tiempo infinito, un alfabeto infinito" (Eberbach Eugene, Goldin Dina y Wegner Peter, 2004: 17)

Proponen modelos computacionales más potentes al de la máquina de Turing, consideramos que deberían considerarse modelos más expresivos, que reflejan la combinación de diversos componentes, en relación a una representación más directa a las máquinas tal como hoy las tenemos. El planteamiento de Eberbach, Goldin y Wegner se expresa de la siguiente forma:

> Esperamos que la computación en súper-Turing (modelos) se haga un paradigma central de la ciencia de la computación. Sin embargo, no podemos afirmar que los súper-modelos de Turing, como se describe en este capítulo, son definitivos y completos. Más probable es que será reemplazado en el futuro por los modelos que son aún mejor y más completo para la solución de problemas, en la interminable búsqueda para una mejor descripción de la realidad (Eberbach Eugene, Goldin Dina y Wegner Peter, 2004: 34)

Sabemos que en la teoría de Turing se define una máquina que tiene un dispositivo al que se denomina Oráculo (máquina-O). Alrededor de la interpretación de la máquina-O se ha producido un debate filosófico, que aún no está terminada, en cuanto al alcance y posibles significados del dispositivo, incluso sirve como argumento para sustentar la computabilidad cuántica[199].

Consideramos que Turing incorporo la comunicación en la máquina abstracta, en la medida que está constituida por dispositivos que requieren comunicarse entre sí, haciéndose más evidente cuando se trata del oráculo. En tal razón, coincidimos con Robert Soare sobre la interpretación de la máquina-O, en relación a la red de computadoras en Internet.

> … procesos en línea o interactivos de computación son un proceso que interactúa con su medio ambiente, por ejemplo, un computador comunicándose con una base de datos externa tal como en la World Wide Web... este parece que la o-máquina de Turing es un buen modelo teórico para analizar un proceso interactivo porque suele haber un algoritmo o procedimiento fijado en el centro, que por la tesis de Turing podemos identificar con una a-máquina de Turing, y hay un mecanismo para el proceso de comunicarse con su entorno, que cuando codificados en números enteros puede ser considerado como un tipo oráculo de Turing (Soare Robert, 2009:41)

Turing define la máquina-A y luego describe la máquina-C, en el sentido de ser una máquina como el hombre que opera con lápiz y papel, también refiere a la máquina-O, así lo recuerda Eberbach, Goldin y Wegner, al respecto:

> Máquinas automáticas (a-máquinas) no eran el único modelo presentado por Turing en su ensayo de 1936. En el mismo trabajo, Turing propuso las maquinas-elección (c-máquinas) como un modelo alternativo de cómputo. Considerando que una a-máquina opera en la forma de una caja cerrada como si en "piloto automático" (de ahí su nombre), c-máquinas interactúan con un operador como un usuario humano durante el cálculo (Eberbach Eugene, Goldin Dina y Wegner Peter, 2004: 5)

Se suele interpretar que las máquinas-O están relacionadas a las máquinas-C, porque el oráculo sería equivalente al hombre, en el sentido que es el operador de la máquina, y resuelve lo que la máquina puede calcular.

[199] Computabilidad cuántica, es el tipo de computabilidad basada en máquinas que se sustentan en la física cuántica, de forma tal que permiten la construcción de una computadora que podría computar lo que no puede hacer una máquina de Turing.

> Eventualmente a-máquinas son adoptadas como el modelo estándar de la computación y para las c-maquinas algunos creen que las máquinas con oráculo introducida por Turing justo pocos años después (38) provee la formalización de las c-máquinas haciendo a estas innecesarias. (Eberbach Eugene, Goldin Dina y Wegner Peter, 2004: 6)

El concepto de oráculo lo entendemos como que es implementado en la parte de los programas, en los procesos de consistencia que se ejecutan con el propósito de verificar la información a fin de que corresponda a las reglas que ejecutará el programa, garantizando que se llegará a un resultado.

En la medida que en las computadoras se presentan operaciones de forma interactiva, los mecanismos de consistencia de los datos se hacen más potentes y necesarios, tomándose en cuenta lo enunciado por Turing sobre la máquina abstracta y el problema de la parada. Para evitar que esto ocurra los datos deben ser consistentes antes de que se inicie la ejecución. La verificación asegura de que los datos correspondan a las reglas del programa garantizando que se ejecutará correctamente.

4.10. Modelo y Máquina de Turing.

Manejamos el concepto de modelo en los dos sentidos que fueron expuestos en el primer capítulo, resumiéndola como una teoría. Sostenemos en un primer sentido que la máquina de Turing es un modelo de la computadora y en otro sentido que la teoría basada en la máquina de Turing, permite conceptuar diversos modelos de autómatas.

Consideramos lo representado como modelo de la representación, como ocurre al considerar modelo al objeto que sirve de muestra. La máquina de Turing es un modelo elemental de cualquier dispositivo mecánico eléctrico, al ser una máquina de lápiz y papel, define las características de lo que se entiende por computable en relación al formalismo establecido en los dispositivos de la máquina abstracta y su relación lógica expresada mediante el procedimiento efectivo.

Consideramos en otro sentido, la representación como modelo de lo representado, como teoría que deriva modelos explicativos. La máquina de Turing siendo la teoría y los autómatas son modelos derivados de la combinación de máquinas de Turing.

Resulta que si pretendemos explicar un algoritmo para la búsqueda de datos en archivos distribuidos en diversas computadoras (base de datos distribuida) resultará difícil expresarlo hacerlo en la notación de una máquina de Turing. Hoy nos valemos de otros instrumentos para expresar nuestros procedimientos efectivos. El concepto de computabilidad está en la base de lo que expresa el procedimiento efectivo, este no solo es la secuencia de instrucciones, también incorpora los dispositivos que constituyen la máquina y la necesaria comunicación entre estos.

Si consideramos que un modelo es una teoría y la computadora es el objeto real e incluimos en ésta los diferentes aspectos que se dan en la tecnología de la información, como: Las comunicaciones y los dispositivos que interactúan mediante procesos, entonces el concepto de computabilidad en la teoría de Turing contiene las características de la comunicación siendo el modelo elemental de la computadora.

En lo que se refiere a la correspondencia de la máquina de Turing, como una máquina de lápiz y papel, que ejecuta un procedimiento efectivo en el sentido de lo que puede hacer un hombre siguiendo instrucciones sin entendimiento que el necesario para ejecutar, los programas y las computadoras se adecuan a este tipo de realización de la abstracción mencionada. El modelo como formulación aplicada a una realidad, es donde se ejecutará el programa, así en su utilización en las computadoras en situaciones específicas, determinará la realización del modelo.

Consideramos que la máquina de Turing es un modelo básico de la computadora y expresa la computabilidad, referida a la computadora. De otro lado la teorización de la máquina determina diversos modelos de autómatas.

4.11. La Computadora.

Las computadoras son herramientas de apoyo a diversas actividades, como son: Calcular, almacenar, editar documentos, y medio de comunicación entre otros, haciéndose necesarias, como lo demuestran los presupuestos millonarios para la adquisición de tecnología de procesamiento de información.

Para la comunicación entre las computadoras se utilizan algoritmos, basado en la estructura de capas, como exigen los protocolos de comunicación[200]. El concepto de protocolo no es un algorítmo, dado que es un acuerdo que establece reglas, pero en cada nivel del mismo corresponde hablar de algoritmos, como ocurre en los temas académicos de reconocimiento de mensajes, detección de errores, recuperación de errores, entre otros.

La ciencia de la computación se inicia conociendo sus límites, en cuanto a lo que es computable y no es computable. El *Entscheidungsproblem* propuesto por Hilbert y el

[200] Protocolo de comunicación, es un conjunto de reglas usadas para la comunicación de las computadoras mediante una red. El protocolo son las reglas que dominan la sintaxis, semántica y sincronización de la comunicación, permitiendo la comunicación, sincronización y transferencia de datos. Suelen ser implementados mediante dispositivos físicos y programas de computadora.

teorema de la incompletitud de Gödel, están incorporadas a las conclusiones proporcionadas por Church y Turing, en referencia a la función computable, en este sentido, la ciencia de la computación a diferencia de otras ciencias, empezó con un límite en el horizonte, en consideración a lo mencionado, se avanzó en el campo de la lógica, en las teorías de la demostración, la teoría de la información, hasta plantearse los temas de la complejidad computacional. Así Wilfried Sieg dice:

> Entscheidungsproblem de Hilbert, el problema de la decisión en la lógica de primer orden, fue una cuestión que requiere una precisa caracterización de los "métodos efectivos", véase la tesis de la Church-Turing. Aunque parcial, se encontraron respuestas positivas durante la década de 1920, Church y Turing demostraron en 1936 que el problema general es indecidible. El resultado y las técnicas que intervienen en su prueba (por no hablar de conceptos muy matemáticos) inspirado en la investigación de la complejidad de la teoría de la recursión de conjuntos que llevó en primer lugar a la clasificación de la aritmética, híper-aritmética, jerarquías analíticas, y que más tarde de las clases de complejidad computacional. (Sieg Wilfried, 1997:4)

La ciencia de la computación tiene varios años de haberse constituido, podemos decir que son 80 años, desde que se definió la máquina de Turing y que se formulara el cálculo lambda, que luego formaría la teoría de la computabilidad como equivalente a la teoría de las funciones recursivas, siendo la tesis de Church-Turing el concepto base. En los últimos 20 años el concepto de computabilidad es revisado en diversos sentidos a la luz de las mayores capacidades de las computadoras, el avance tecnológico en el campo de las comunicaciones y almacenamiento de los datos, y en desarrollo de la lógica.

En palabras de Robert Soare: “El período moderno de la teoría de computabilidad se puede dividir en tres períodos. 1. Era de definibilidad Lambda 1931-1935. 2. Era de la teoría de la Recursión 1935-1995. 3. Era de la Computabilidad 1996-al presente” (Soare Robert, 2007:1)

Sostenemos que la máquina de Turing expresa adecuadamente el concepto de computabilidad, porque toma en cuenta: i) procedimiento efectivo, en el sentido de ser una definición que permite la ejecución secuencial de instrucciones, como hemos denominado paso a paso. ii) máquina de lápiz y papel como concepto de lo automático de un procedimiento que realiza un hombre, iii) recursividad y recurrencia concepto matemático de funciones y relaciones computables expresada en procesos de forma recurrente. iv) interacción y comunicación como elementos necesarios de una máquina y también entre máquinas. Consecuentemente, esta concepción permitió la fabricación de una computadora, así Martin Davis le otorga un papel importe a la definición de máquina de Turing.

> Alan Turing descubre lo universal (o de uso general) de la computadora digital como una abstracción matemática... este trabajo abstracto ayudó a Turing y John von Neumann hacia la concepción moderna de la computadora electrónica (Martin Davis, 2006:125)

Consideramos que las computadoras electrónicas son equipos construidos según la arquitectura definida por von Neumann, constituida por: Una unidad de control, unidad de memoria, unidad aritmética lógica y un dispositivo de entrada y otro de salida de datos.

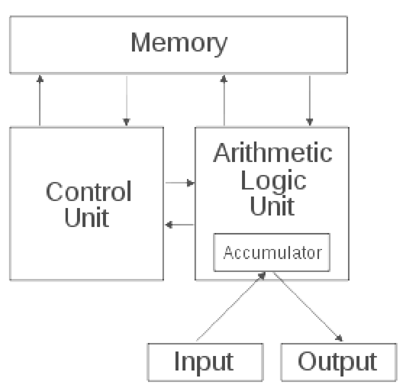

Figura 10. Arquitectura de John von Neumann

La arquitectura de la computadora de von Neumann fue formulada sobre la base del trabajo de Turing, como lo que afirma Martin Davis.

> Prácticamente todos los ordenadores de hoy de más de US$ 10 millones de las súper computadoras hasta los chips de teléfonos celulares y los furbies, tienen una cosa en común: todos ellos son "máquinas de Von Neumann", variaciones sobre la arquitectura básica de John von Neumann, construidas sobre la base del trabajo de Alan Turing, establecido en 1940 (Martin Davis, 2006:126)

Cuando uno revisa la arquitectura de las computadoras según el modelo de von Neumann, tomamos en cuenta, en primer lugar que refiere a componentes físicos y en segundo lugar a la definición los datos y programas como elementos externos. Esta separación es significativa dado que conceptúa a las computadoras en dos áreas tecnológicas (la ingeniería de las máquinas y la ingeniería de los programas). Al considerar el campo denominado como hardware (relacionado a lo físico) y el de software (relacionado a lo lógico), separó en dos grandes problemas involucrados en el tema de lo computable: El primero en cuanto a la máquina y el segundo a los datos y programas.

Los datos en las computadoras se organizan de diferentes formas y se definen mediante estructuras que se relacionan a reglas del tipo lógico. Hoy tenemos estructuras de cierta complejidad, así como diversos dispositivos físicos con mayor capacidad de almacenamiento. Destacamos de la máquina de Turing, que los datos se encuentran expresados en símbolos que se almacenan en la unidad de memoria, mientras que lo denominado programa (por contener las instrucciones de la máquina), son parte de la definición de la máquina. Hoy encontramos computadoras que emulan a otras computadoras, tal como en la máquina universal de Turing (máquina que imita a otra máquina).

Consideramos que los datos corresponden a la parte variable, a lo que no es físico y los programas son en cierto sentido una parte fija, pero en la teoría de Turing se considera que los programas pueden ser variables como aquellos que “aprenden” (en el campo de la Inteligencia Artificial), es decir, programas que cambian su estructura lógica de la ejecución. En la máquina de Turing están presentes los componentes mencionados: físico, datos y programa, así Martin Davis precisa:

> ...las tres categorías, la máquina era un objeto físico… hardware. El programa fue el plan para hacer una computación... Los datos eran el ingreso numérico. La máquina universal de Turing demostró que la distinción de estas tres categorías es una ilusión... Por último, la máquina universal en sus acciones paso-a-paso ve que... el código de máquina como datos para ser trabajado. Esta fluidez... es fundamental para la práctica contemporánea del computador (Martin Davis, 2006:128)

En el propósito de calcular funciones existe en debate sobre la potencia para determinar si una función es computable, dado que en la máquina de Turing el problema se plantea como el proceso que si no se detiene entonces no es computable (en la máquina-A), pero en la máquina-O, tiene capacidad de determinar si una función es computable. Es alrededor del dispositivo Oráculo como computadora en la que se manifiesta temas como el de hipercomputación, así tenemos en palabras de Martin Davis: “A pesar de todo lo anterior, sería tonto afirmar que no habrá ningún futuro dispositivo que será capaz de calcular lo no computable. De hecho, la necesidad actual del movimiento de la "hipercomputación" llama nuestra atención” (Martin Davis, 2006:130)

En nuestra opinión el concepto de hipercomputación es una extensión del concepto de computabilidad en la teoría de Turing, referidos a los procesos de consistencia necesarios y suficientes que garantizan la ejecución computacional, dado que automáticamente se cumplan con las reglas lógicas. En correspondencia con el teorema de incompletitud de Gödel, en el sentido, que una vez construido un programa, este corresponde a una problemática de un universo especifico, pero dada la incompletitud de lo formalizado, siempre será necesario el mantenimiento del programa que consiste en añadir nuevas reglas de situaciones que no estaban incluidas. En casos prácticos mejorando las reglas de consistencia, consecuencia del funcionamiento del programa. Asegurando que el mecanismo del tipo oráculo en potencia sea "completa".

Para la definición de la arquitectura de una computadora, von Neumann profundiza el sentido y forma del código de la máquina en relación a la ejecución de los procesos, esta refiere a la imitación de otra máquina (máquina-U), en el sentido que diferentes máquinas al ejecutar una secuencia de código, todas obtendrían el mismo resultado. El código en las computadoras, es la característica significativa en el concepto de la máquina de Turing, tal como lo dice von Neumann: "En contraste con los códigos completos, existe otra categoría de códigos mejor designados como códigos cortos. Estos están basados en la siguiente idea." (von Neumann, 1958:71)

El concepto de código corto es el código de Turing, como parte de la sintaxis de las instrucciones que constituyen el procedimiento efectivo. El código es el que determina el comportamiento de la máquina, en cuanto que permite la ejecución de lo que está establecido en las instrucciones del programa, es en este sentido que von Neumann precisa el concepto de imitación de una máquina.

> Un código según el esquema de Turing supone que hacen que una máquina se comporte como si fuera una máquina específica (que supone que imita a este último) debe hacer las siguientes cosas. Debe contener en términos de que la máquina entienda y (deliberadamente obedecer), las

> instrucciones (partes más profunda del código) que hará que la máquina examine todos los pedidos que recibe y determina si esta orden tiene la estructura adecuada a una orden de la segunda máquina (von Neumann, 1958:72)

Consideramos que la tesis de Church-Turing pone en la misma referencia al cálculo de una función recursiva y el concepto del código corto (expresado en el procedimiento efectivo) en la perspectiva de la arquitectura de una máquina que ejecuta instrucciones. La recursividad no refleja completamente el sentido de un código máquina, es siguiendo esta reflexión donde encontramos adecuada la opinión de von Neumann resaltando la teoría de Turing, como sigue:

> El resultado importante de Turing es esta manera en la que la primera máquina puede imitar el comportamiento de cualquier otra máquina. La estructura de orden que es lo que causa a seguir que puede hacer completamente diferente a una característica de la primera máquina que está verdaderamente involucrada. Así, la estructura de orden a la que se refiere en realidad, puede hacer frente a las órdenes en un carácter mucho más complejo que son característicos de la primera máquina: cada una de estas órdenes de la máquina secundaria puede implicar la realización de varias operaciones por la máquina mencionada en primer lugar… La razón, para llamar a un código secundario un código corto es, por cierto, histórico: los códigos cortos se han desarrollado como una ayuda a la codificación, es decir, son el resultado de la voluntad de ser capaz de código más breve para un equipo que posee su sistema de orden natural que permitiría, tratándola como si se tratara de una máquina diferente con un sistema más conveniente (von Neumann, 1958:73)

Para von Neumann, metafóricamente, la computadora se compara al cerebro humano, pensaba que en el estudio del sistema nervioso se puede encontrar el conocimiento de los formalismos de la comunicación y del cálculo, esto lo encontramos en el mismo propósito de Turing en la investigación en los últimos años de su vida.

En el caso de von Neumann describe al final de su libro *The Computer and the Brain* que el lenguaje del cerebro no es el lenguaje de las matemáticas, indica de que hay un lenguaje en relación al código que es más sencillo al utilizado para hablar, dado que el procedimiento efectivo en la máquina de Turing está constituido por códigos básicos necesarios y suficientes para operar los diversos dispositivos, basado en la definición abstracta de autómata. En referencia a esto señala Von Neumann lo siguiente:

> … debe tenerse en cuenta que el lenguaje aquí involucrado puede corresponder a un tipo de código en el sentido descrito anteriormente en lugar de un código completo: cuando hablamos de las matemáticas, podemos estar hablando una lengua secundaria, basada en la lengua primaria verdaderamente utilizados por el sistema nervioso central (von Neumann, 1958:82)

Consideramos que hemos profundizado en la diferenciación de conceptos que contribuyen al esclarecimiento del significado de la computabilidad, el ignorarlos reducen la interpretación en diversos temas que se dan en la ciencia de la computación. En la diferenciación contribuimos en el sentido expuesto sobre la filosofía de la computación (en la sección 4.3 del presente capítulo).

CONCLUSIONES

1. La tesis de Church-Turing sostiene la computabilidad de una función matemática, en la que equipara el cálculo lambda en referencia a las funciones recursivas, con el procedimiento efectivo definido en la máquina de Turing, siendo estas definiciones de naturaleza matemática.
2. El concepto de computabilidad tiene un significado más amplio que el de solo función computable, ya que contiene condiciones necesarias y suficientes que son explicadas mediante conceptos que necesariamente no son funciones recursivas.
3. Existe diferencias conceptuales entre el procedimiento efectivo en la teoría de Turing y el cálculo lambda en la tesis de Church. Si los separamos, encontramos en la definición de la máquina de Turing, el concepto de computabilidad en un sentido más amplio y riguroso.
4. El concepto de computabilidad, es un procedimiento que se ejecuta paso a paso, que está constituido por instrucciones del tipo imperativo, que no requieren interpretación, pues cada instrucción contiene la suficiente información para hacer lo que se manda hacer.
5. El concepto de computabilidad se asocia al procedimiento que puede ser ejecutado por un hombre, utilizando lápiz y papel, siguiendo las instrucciones, sin requerir mayor interpretación.
6. La capacidad de computar está en relación con las operaciones lógicas que podemos ejecutar mediante códigos o signos, en la medida que asociamos el funcionamiento de dispositivos lógicos de naturaleza mecánica y eléctrica. La máquina que computa

tiene la capacidad de imitar a otras máquinas, formulada en la máquina universal de Turing.

7. Las funciones recursivas son una condición necesaria en el concepto de computabilidad, en el sentido que refleja a un lenguaje formal de procedimientos mecánicos del tipo bucle o del tipo recursivo. Siendo estos procedimientos diferentes en su implementación y ejecución.
8. El concepto de computabilidad no se agota en la recursividad, su elucidación abre diversos campos de investigación como: complejidad del cálculo, aumento de la potencia de cálculo, interacción de las computadoras en relación con la máquina de Turing con Oráculo, y aprendizaje en las computadoras.
9. La interacción es un componente necesario en la definición de la computabilidad, porque expresa la relación del funcionamiento de los diversos dispositivos del equipo que hace la computación.
10. La comunicación es un componente necesario en la definición de la computabilidad, porque se requiere transmitir instrucciones y resultados.
11. La máquina de Turing como modelo derivado de una teoría (según las ciencias formales), supone una teoría de la computabilidad. Esta debe incorporar los diversos aspectos que se tratan en la teoría de Turing.
12. La filosofía de la computación se encuentra en proceso de maduración y desarrollo, ésta se nutre de la filosofía tradicional y del desarrollo de la ciencia y tecnología de la computación, siendo el concepto de computabilidad el más importante a elucidar, tomando en cuenta la ontología, el método y su orientación, considerando a las matemáticas no como un todo, sino más bien como parte de la ciencia de la computación.

BIBLIOGRAFIA

Acero J, Bustos E, Quesada D. (1989). *Introducción a la Filosofía del Lenguaje.* España, Madrid: Ediciones Cátedra S.A

Aczel Amir. (2005). *El Último Teorema de Fermat.* (Domínguez Roberto Trad.). México DF, México: Fondo de Cultura Económica.

Aho Alfred y Ullman Jeffrey. (1972). *The Theory of Parsing Translation and Compiling*. Englewood Cliffs NJ, USA: Prentice Hall Inc.

Alchourron Carlos, Méndez José y Orayen Raúl. (1995). *Lógica.* Madrid, España: Editorial Trotta.

Álvarez Carlos, Martínez Rafael, Ramírez Santiago y Torres Carlos. (1993). *David Hilbert: Fundamentos de las Matemáticas.* (Segura Luis Trad.). México DF, México: Servicios Editoriales de la Facultad de Ciencias, UNAM.

Aris Rutherford, Davis Ted y Stuewer Roger. (1995). *Resortes de la Creatividad Científica. Ensayos sobre fundadores de la ciencia moderna.* (Ámela Juan Trad.). México DF, México: Fondo de Cultura Económica.

Aristóteles. (1994). *Tratados de lógica (Órganon).* (1a ed.). (Candel Miguel Trad.). Madrid, España: Editorial Gredos.

Aspray William. (1985). *The Princeton Mathematics Community in the 1930.* www.princeton.edu/~mudd/finding_aids/mathoral/pmc05.htm (2009, 12 de abril)

__________ (1993). *John von Neumann y los orígenes de la computadora moderna.* (Alterman Elena Trad.). Barcelona, España: Editorial Gedisa.

Azcarate María de Ponte. (2005). *Realismo y entidades abstractas. Los problemas del conocimiento en matemáticas.* Tesis de Grado de Doctor. Universidad de la Laguna de España, Facultad de Humanidades y Ciencias Sociales.

Bacon Francis. (1984). *Novum Organum.* (Litrán Cristóbal trad.). Madrid, España: Editorial Sarpe.

Babini José. (1967). *Historia de las Ideas Modernas en Matemáticas.* (1a ed.). Washington DC, USA: The Pan American Union Washington D.C.

Barrow D.W. (1976). *Técnicas Recursivas de Programación.* (Forno Ricardo Trad.). Buenos Aires, Argentina: Editorial Kapehisz.

Barrow John. (1999). *Imposibilidad. Los límites de la ciencia y la ciencia de los límites.* (1a ed.). (Carlos de la Reta Trad.). Barcelona, España: Editorial Gedisa.

Bell E.T. (1999). *Historia de las Matemáticas.* (2a ed.). (R. Ortiz Trad.). México DF, México: Fondo de Cultura Económica.

Berlinski, David. (2007). *Ascenso infinito.* (1a ed.). (Rubén Díaz Trad.). Buenos Aires, Argentina: Editorial Debate.

Bertrand Russell. (1920). *Introducction to Mathematical Philosophy*. Londres, Inglaterra: London George Allen & Unwin LTD y New York, USA Ed: The Macmillan Co.

___________ (1981). *Lógica y Conocimiento*. (Muguerza Javier Trad.). Madrid, España: Taurus Ediciones S.A.

Bertrand Russel y Whitehead Alfred. (1920). *Principia Mathematica, Introducction to Mathematical Philosophy*. Londres, Inglaterra: London George Allen & Unwin LTD y New York, USA: The Macmillan Co.

Blanche, Robert. (1965). *La Axiomática.* (1a ed.). (Pulido Ana Trad.). México DF, México: Editorial Fondo de Cultura Económica.

Boolos G, Burgess y Jefrrey R. (1988). *Computability and Logic.* (2a ed.). New York, USA: Cambridge University Press.

Brandford, P. and Wollowski, M. (1994). *A formalization of the Turing Test.* SIGAR

Bulletin, V 6, pp 4-10. cs.indiana.edu/pub/techreports/tr99.pdf. (2008, 16 de junio).

Brunschvicg Leon (1945). *Las Etapas de la Filosofía Matemática*. (1a ed.) (Cora Rato trad.). Buenos Aires, Argentina: Editorial Lautaro.

Cardoso Ricardo. (2007). *Autómatas* agamenon.uniandes.edu.co/~rcardorso/avisos/mes~/Material/Automatas1.pdf. (2009, 12 de abril).

Castellano Peñuela Juan. (1987). *Una formulación de las relaciones entre distintas representaciones del conocimiento*. Tesis de grado de Doctor, Universidad Politécnica de Madrid, Facultad de Informática, Madrid España.

Calvo Vélez, David. (2006). *Modelos teóricos y representación del conocimiento.* Tesis de Grado de Doctor. Universidad Complutense de Madrid, Facultad de Filosofía. Madrid, España.

Cardona Suarez Carlos. (2003). *Wittgenstein & Gödel. Debate acerca del sentido y la interpretación de las proposiciones matemáticas*. Tesis de grado de Doctor, Universidad Nacional de Colombia, Facultad de Ciencias Humanas. Bogota, Colombia.

Carnap R. (1958). *Introduction to symbolic logic and its applications.* (1a ed.). (Meyer W y Wilkinson J. trad.). New York, USA: Dover Publications INC.

Carter Matt. (2007). *Mind and computers, an introduction to the philosophy of artificial intelligence.* (1a ed.). Edinburgh, Scotland: Edinburgh University Press.

Coello Coello, C. (2004). *Breve historia de la computación y sus pioneros.* México DF, México: Fondo de Cultura Económica.

Cood E.F. (1990). *The Relational Model for Database Management*. Boston, MA, USA: Addison Wesley Publishing Company.

Copeland Jack. (2004). *The Essential Turing Seminal Writings in Computing, Logic,* *Philosophy, Artificial Intelligence, and Artificial Life plus the Secrets of Enigma.* Oxford New York, USA: Clarendon Press-Oxford.

__________ (2004). *The Church-Turing Thesis*. NeuroQuantology, Issue 2, PP 101-115.

Copeland Jack y Proudfoot, D. (1999). *Alan Turing's forgotten Ideas in Computer Science* Scientific American Inc, April, pp 99-103 http://www.cs.virginia.edu/~robins/Alan_Turing's_Forgotten_Ideas.pdf. (2008, 1 de mayo).

Corominas Joan. (1990). *Breve Diccionario Crítico Etimológico Castellano e Hispánico*. Madrid, España: Editorial Gredos.

Courant Richars y Robbins Herbert. (2006). *¿Qué son las matemáticas?*. (Manrrique Martin Trad.). México DF, México: Fondo de Cultura Económica.

Crade Tim. (2008). *La mente mecánica, introducción filosófica a mentes, máquinas y representación mental*. (Almela Juan Trad.). México DF, México: Fondo de Cultura Económica.

Crosson Frederick J. (1970). *Inteligencia Humana e Inteligencia Artificial.* (Pérez José Luis Trad.). México DF, México: Fondo Editorial de Cultura.

Cuena José. (1985*). Lógica Informática*. (1a ed.). Madrid, España: Alianza Editorial.

Chaitin Gregori (2002). *Información y Azar.* Boletín de la Asociación Matemática Venezolana, Volumen IX, Número 1, PP 55-81.

__________ (2003). *Ordenadores, Paradojas y Fundamentos de las Matemáticas*, www.cs.auckland.ac.nz/~chaitin/investigacion-y-ciencia.pdf. (2009, 12 de abril).

Chomsky Noam (1965). *Aspectos de la teoría de la sintaxis*. (Otero C.P. Trad.). España, Madrid. Editorial: Aguilar.

__________ 1997. *Entrevista a Noam Chomsky*, www.psicothema.com/pdf/128.pdf (2008, 20 de Octubre).

Church Alonzo. (1936). *An Unsolvable Problem of Elementary Number Theory*, www.jstor.org/stable/2371045. (2008, 25 de agosto).

__________ (1956). *Intoduction to Mathematical Logic.* Princeton New Jersey, USA: Princeton University Press.

Da Costa N.C.A. (2000). *Lógica Inductiva y Probabilidad*. (1a ed.). Lima, Perú: Fondo de Cultura Económica.

Dahl O.J. y Dijkstra E.W. y Hoare C.A.R. (1972). *Structured Programming*. New York, USA: Academic Press Inc.

Davis Martin. (1987). *Influencias de la Lógica Matemática en las Ciencias de la Computación*. (García Facundo Trad.). www.econ.uba.ar/www/departamentos/humanidades/plan97/logica/Legris/apuntes/ Davis.pdf. (2009, 15 de julio).

_________ (1995). *American Logic in the 1920s.* The Bulletin of Symbolic Logic, Volume I, Number 3, Sept, PP 273-278.

_________ (2006). *The Church–Turing thesis consensus and opposition*. citeseer.ist.psu.edu/viewdoc/sumary?doi=10.1.1.91.7093. (2009, 12 de enero).

_________ (2006). *What is Turing Reducibility?* Notices of the AMS, Volume 53, Number 10, November, PP 1218-1219.

_________ (2008). *Hilbert's Tenth Problem is Unsolvable.* Naughtybag.files.wordpress.com/2008/03/martindavis.pdf. (2009, 13 de mayo).

Davis Martin y Hilary Putnam. (1960). *A computing procedure for quantification theory*. Journal of the ACM, Volume 7, Number 3, July PP 201-215.

Denning Peter (2003). *Great principle of Computing.* Communications of the ACM, Volume 46, Number 11, November, PP 15-20.

_________ (2005). *Is computer Science Science?* Communications of the ACM, Volume

48, Number 4, April, PP 27-31.

_________ (2007). *Computing is a Natural Science.* Communications of the ACM, Volume 50, Number 7, July, PP 13-18.

Díaz Estévez Emilio. (1975). *El Teorema de Gödel*. (1a ed.). Pamplona, España: Ediciones Universidad de Navarra S.A.

Díez José y Moulines Ulisis. (1997). *Fundamentos de la Filosofía de la Ciencia*. Barcelona, España: Editorial Ariel S.A.

Dorothy Elizabeth. (1982). *Cryptograpfhy and Data Security*. Massachusetts, USA: Addison Wesley Publishing Company Inc.

Duncan Ronald y Weston Miranda. (1996). *La Enciclopedia de la Ignorancia.* (Helier Roberto Trad.). México DF, México: Fondo de Cultura Económica.

Eco Umberto. (1993). *Entrevista sobre semiótica y pragmatismo*. http://www.observacionesfilosoficas.net/umbertoeco.html. (2008, 20 de octubre).

Eden Amnon (2007). *Three Paradigms of Computer Science.* Philosophy of Computer Science, Volume 17, Number 2, July. PP 135–167.

Euclides (1991). *Elementos.* (María Luisa Puerta Castaños, Trad.) Madrid, España: Editorial Gredos.

Euclides (2010). *Los Elementos de Euclides*. www.euclides.org. (2010, 26 de octubre).

Fernández Inmaculada. (1992). *Inconmensurabilidad y Racionalidad Científica.* Tesis para obtener el grado de Doctor en la Facultad de Filosofía de la Universidad Complutense de Madrid. Madrid, España.

Ferrater Mora. (2004). *Diccionario de Filosofía.* Barcelona, España: Editorial Ariel S.A.

Flores Alfonso, Holguín Magdalena y Meléndez Raúl. (2003). *Del espejo a las herramientas. Ensayos sobre el pensamiento de Wittgenstein.* Bogotá, Colombia:

Siglo del Hombre Editores - Pontificia Universidad Javeriana – Universidad Nacional de Colombia.

Floridi, Luciano. (1999). *Philosophy and Computing: An introduction*. New York, USA: Routledge of the Taylor & Francis Group

_________ (2004). *Philosophy of Computing and Information*. London, Kingdom: Blackwell Publishing Ltd.

_________ (2007). *Semantic Conceptions of Information.* plato.stanford.edu/archives/spr2007/entries/information-semantic. (2010, 20 de enero)

Frege Gottolb (1972). *Conceptografía. Los Fundamentos de la Aritmética.* (Padilla Hugo Trad.). México DF, México: Editorial Instituto de Investigaciones Filosófica.

Gallego, Enrique. (2001). *Técnicas de demostración e indecibilidad e inseparabilidad en teorías formales.* Tesis para obtener el grado de Doctor en la Facultad de Filosofía de la Universidad Complutense de Madrid. Madrid, España.

García Zarate Oscar (2003). *Introducción a la lógica.* Lima, Perú: Fondo Editorial de San Marcos.

_________ (2007). *Lógica.* Lima, Perú: CEPREDIM.

_________ (2009). *Elementos de Lógica.* Lima, Perú: Editorial e Imprenta Wari S.A.C.

Goldin Dina, Wegner Peter. (2003). *Computation Beyond Turing Machines.* Communications of the ACM, Volume 46, Number 4, April, PP 100-102.

_________ (2005). *The Church-Turing Thesis: Breaking the Myth.* LNCS 3526, Springer PP 152-168.

_________ (2007). *The Interactive Nature of Computing: Refuting the Strong Church-Turing Thesis.* www.cs.brown.edu/people/pw/strong-cct.pdf. (2009, 14 de abril).

Goldin Dina, Wegner Peter y Eberbach Engene (2004). *Turing's ideas and Models of* *computation*. www.cse.uconn.edu/~dqg/papers/turing04.pdf. (2009, 12 de abril).

Guerra Hernandz, A. (1999). *Introducción al cálculo lambda*, http://www.uv.mx/aguerra/teaching/fp-06/clase09.pdf. (2008, 17 de junio).

Guillermo Martínez y Piñeiro Gustavo. (2009). *Gödel para todos*. Buenos Aires, Argentina: Emece Editores S.A.

Hasen Brinch (1977). *The Architecture of Concurrent Programs*. New Jersey, USA: Prentice Hall Inc.

Haugeland, John. (1999). *La Inteligencia Artificial*. (Tullí Irene Trad.). Buenos Aires, Argentina: Siglo XXI.

Hawking Stephen. (2010). *Dios creó los Números*. (Ariz Ubaldo Trad.). Barcelona, España: Editorial Crítica.

Hermes Hans. (1984). *Introducción a la teoría de la computabilidad Algoritmos y Máquinas*. (Garrido Manuel y Martin Aránzazu trad.). Madrid, España. Editorial Tecnos.

Hilbert David. (1900). *Mathematical Problems Lecture delivered before the International Congress of mathematical at Paris 1900*. aleph0.clarchku.edu/˜djoyce/hilbert/problems.html. (2008, 18 de marzo).

Hoare C.A.R. (1969). *An Axiomatic Basic for Computer Programming.* Communications of the ACM, Volume 12, Number 10, Octuber. Pp. 576-583.

_________ (1973) *Hints on programming language design. Memo AIM-224 stan-cs-73-403. Stanford University, Computer Science Departament.*

_________ (1987) *Law of Programming.* Communications of the ACM, Volume 30, Number 8, Octuber. Pp. 672-686.

Hoare C.A.R. y Jones C.B. (1989). *Essays in Computing Science.* Cambridge, Great Britain: Prentice Hall International.

Hofstadter Duglas. (2007). *Gödel, Escher, Back: Un Eterno y Gragil Bucle.* (Usabiaga Mario y López Alejandro Trad.). Barcelona, España:Tusquets Editores S.A.

Hopcroft John y Ullman Jeffrey (1969). *Formal Languages and Their Relation to Automata.* Massachusetts, USA: Addison Wesley Publishing Company.

Huber Kennedy (2002). *Twelve Articles in Giuseppe Peano.* San Francisco, USA: Peremptory Publications.

_________ (1963). *The Mathematica Philosophy of Giuseppe Peano.* Philosophy of Science, Volume 30, Number 3, July. PP 262-266.

Kleene S.C. (1935). *A theory of positive integers in formal logic. American* Journal of Mathematics, Volume 57, Number 12, January. PP. 153-173.

_________ (1952). *Introducción a la Metamatemáticas.* (1a ed.) (Garrido Manuel trad.) Madrid España: Tecnos S.A.

Knuth Donald Ervin. (1970). *Von Neumann's first computer program.* Computing Surveys Volume 2 Number 4, December pp 247-260.

_________ (1974). *Computes programming as a art.* Communication of the ACM Volume 17 Number 12, pp 667-673.

_________ (1981). *The Art of Computer Programming.* (2a ed.), Massachusetts, USA: Addison Wesley Serie.

Kolman Bernard y Baby Robert. (1986). *Estructuras de Matemáticas Discretas para la Computación*. (Herrera Arturo Trad.). Naucalpan de Juárez, México: Prentice Hall.

Kolmogorov A.N. y Fomin S.V. (1975). *Elementos de la Teoría de Funciones y del Análisis Funcional*. (Vega Carlos Trad.). Moscú, URSS: Editorial MIR.

Kuhn, T. (1970). *La estructura de las revoluciones científicas*. (2a ed.). (Contin A. trad.). Madrid, España: Fondo de Cultura Económica.

Lavine Shaughan. (2005). *Comprendiendo el Infinito.* (Torres Esteban Trad.). México DF, México: Fondo de Cultura Económica.

Lakatos Imre. (1983). *La metodología de los programas de investigación científica.* (Juan Carlos Zapatero Trad.). Madrid, España: Alianza Editorial.

_________ (1986). *Prueba y refutaciones. La lógica del descubrimiento matemático.* (Worrall John y Zahar Elie Trad.). Madrid, España: Alianza Editorial.

_________ (1987). *Matemáticas, ciencia y epistemología*. (Rives Diego Trad.). Madrid, España: Alianza Editorial.

Lewin Renato. (2003). *Logical Modal.* www.labmat.puc.cl/cursos/archivos/2003/1/MAT1405/1056116600/111.pdf. (2009, 15 de diciembre).

Locke J. (1689). *Ensayo sobre el entendimiento humano*. (1a ed.). (O'Gorman E. Trad.). México DF, México: Fondo de Cultura Económica.

Manin Yu I. (1979). *Lo Demostrable e Indemostrable*. (Kotenko E. Trad.). Moscú, URSS: Editorial MIR.

_________ (2010). *A Course in Mathematical Logic for Mathematicians*. (2 ed.). New York, USA: Springer.

Manzano María. (1989). *Teoría de Modelos.* Madrid, España: Alianza Editorial.

_________ (1999). *Vida, Obra y algunos milagros de Alonso Church.* Ágora, Volumen 18, Número 1: PP 107-132.

_________ (2005). *¿Qué es esa cosa llamada lógica?.* En Representación y Logicidad. be.geocities.com/fapespcleime/resources/conferences/simposiorepresentacionylogicidad2005.pdf (2008, 15 de mayo)

_________ (2006). *Sobre el Razonamiento Formal. E*n 50 Años de la Inteligencia Artificial. (Caballero Antonio, Miguel Sergio Eds.). Albacete, España: Universidad de Castilla-La Mancha, Departamento de Sistemas Informáticos.

Mc Carthy J. (1956). *The inversion of function defined by Turing machines.* www-formal.stanford.edu/jmc/inversion.pdf. (2007, 23 de diciembre).

_________ (1959). *Programs with common sense.* www-formal.stanford.edu/jmc/mcc59.pdf. (2007, 23 de diciembre).

_________ (1962). *A basis for a mathematical theory of computation.* www.dspace.imt.edu/bitstream/handle/1721-1/6099/AIM-031.pdf. (2008, 15 de diciembre).

_________ (1977). *Epistemological problems of artificial intelligence.* www.formal.stanford.edu/jmc/epistemological.pdf. (2007, 23 de diciembre).

_________ (1996). *Towards a mathematical science of computation.* www.formal.stanford.edu/jmc/towards.pdf. (2007, 23 de diciembre).

Monlau Pedro Felipe. (1856). *Diccionario Etimológico de lengua Castellana*. Madrid, España: Imprenta y Estereotipia de M. Rivadeneyra.

Mosterín Jesús. (1999). *Límites del conocimiento y de la acción*. Números. Revista de didáctica de las matemáticas, Volumen 40, diciembre, PP 45-53.

_________ (2000). *Los Lógicos* (3a ed.). Madrid, España: Editorial Esparza Calpe S.A.

_________ (2003). *Conceptos y Teorías en la Ciencia* (1a ed.). Madrid, España: Alianza Editorial S.A.

_________ (2006). *Kurt Gödel Obras Completas* (1a ed.). Madrid, España: Alianza Editorial S.A.

_________ (2010). *Naturaleza, Vida y Cultura.* (1a ed.). Lima, Perú: Fondo Editorial de la Universidad Inca Garcilaso de la Vega.

Mosterín Jesús, Torretti Roberto. (2002). *Diccionario de lógica y filosofía de la ciencia.* (1a ed.). Madrid, España: Alianza Editorial S.A.

Nagel Ernest, Newman James. (1959). *La Prueba de Gödel.* (Viran Ramón Trad.) México

DF, México: Centro de Estudios Filosóficos UNAM.

Padilla Gálvez Jesús. (2000). *Tratado Metateórico de las Teorías Científicas*. La Mancha, España: Universidad de Castilla – La Mancha.

Palau Gladys. (2002). *Introducción Filosófica a las Lógicas no Clásicas*. Barcelona, España: Gedisa S.A.

_________ (2006). *Lógica natural e Inteligencia artificial. E*n 50 Años de la Inteligencia Artificial. (Caballero Antonio, Miguel Sergio Eds.). Albacete, España: Universidad de Castilla-La Mancha, Departamento de Sistemas Informáticos.

Palau Gladys y colaboradores. (2004). *Lógicas Condicionales y Razonamiento de sentido común*. Barcelona, España: Gedisa S.A.

Peirce C.S. (1878). *Deducción, inducción e hipótesis.* (1a ed.). (Ruiz J.M. Trad.). Buenos Aires, Argentina: Editorial Aguilar Argentina S.A.

_________ (1881). *On the logic of number*. (Traducción no conocida). Boletín de Matemáticas Nueva Serie, Volumen X, Número. 1 (2003) pp. 13-20, Departamento de Matemáticas y Estadística de la Universidad Nacional de Colombia.

_________ (1897). *La Lógica de las matemáticas un intento de Desarrollar mis Categorías desde dentro*. (Vevia Fernando Trad.). www.unav.es/gep/LogicOfMathematics.html. (2009, 15 de diciembre).

_________ (1987). *Obra Logico-Semiotica.* (R. Alcalde y M Prelooker Trad.). España Madrid: Taurus Ediciones.

_________ (1971). *Mi Alegato a favor del pragmatismo.* (Juan Martin Ruiz Trad.). Argentina, Buenos Aires: Editorial Aguilar.

Penrose, Roger. (1989). *La nueva mente del emperador*. (García Sanz Trad.). México DF, México: Fondo de Cultura Económica.

_________ (1994). *Las sombras de la mente*. (García Sanz Trad.). Barcelona, España:

Critica.

_________ (2004). *El camino a la realidad*. (García Sanz Trad.). Barcelona, España: Debate.

Penrose Roger, Shimony Abner, Cartwright Nancy, Hawking Stephen. (1999). *Lo grande lo pequeño y la mente humana*. (1a ed.) (García Javier Sanz Trad.). Madrid, España: Cambridge University Press.

Perazzo, Roberto. (1994). *De Cerebros Mentes y Máquinas*. Buenos Aires, Argentina: Fondo de Cultura Económica.

Petzold Charles. (2008). *The Annotated Turing*. Indianapolis, USA: Editorial Wiley Publishing Inc.

Piccinin Gualtiero. (2003). *Computations and Computers in the Sciencie of Mind and Brain*. Submitted to the graduate of Doctor of Philosophy. University of Pittsburgh, Faculty of Art and Sciences, Pittsburgh EEUU.

Piscoya Hermosa Luis. (2009). *Tópicos en epistemología***.** (1a ed.). Lima, Perú: Universidad Inca Garcilaso de la Vega, Fondo Editorial.

Pratt Terrence. (1987). *Lenguajes de Programación*. (2a ed.). (Pérez Araceli Trad.). Naucalpan de Juarez, Mexico: Prentice Hall.

Post Emil L. (1946). *A variant of a recursively unsolvable problem,* projecteuclid.org/DPubS/Repository/1.0/Disseminate?view=body&id=pdf_1&handle=euclid.bams/1183507843. (2009, 6 de enero).

Putnam Hilary.(1995). *Shadows of the mind by Roger Penrose.* Bulletin of the American Mathematical Society. Volume 32, Number 3, July, PP 370-373.

Quine, W. V. (1972). *Lógica Matemática*. (2a ed.). (Hierro J. Trad.). Madrid, España: Revista de Occidente.

_________ (1951). *Introducción a la Lógica.* (1a ed.). (Bachiller T.R. y Fuentes J.R.

Trad.). Buenos Aires, Argentina: Editorial Espasa Calpe S.A.

_________ (1969). *Los Métodos de la lógica.* (1a ed.). (Sacristán Manuel Trad.). Barcelona, España: Editorial Ariel.

Rapaport William. (2005). *Philosophy of Computer Science: An Introductory Course*. www.cse.buffalo.edu/rapaport. (2009, 25 de julio).

Real Academia Española. (2009). *Diccionario en línea de la Real Academia Española* www.rae.es. (2009, 20 de julio).

Rivas Monroy María Uxia. (1990). *La noción de sentido Fregeana ¿Semántica, epistemología u ontología?. Ágora, Volumen 9, PP 83-95*

_________ (1996). *Frege y Peirce: En torno al Signo y su Fundamento.*

_________ (1998). *Verdad, Realidad y Ciencia en C.S. Peirce.* Volumen 17, Número 2, PP 79-94.

_________ (2002). *Signo, Mediación y Realidad. Una Lectura de G. Frege y C.S. Peirce.* Volumen 21, Número 2, PP 49-72.

Santana de la Cruz Carmen Margarita. (2005). *Explicación, experimentos y tecnología.* Tesis de Grado de Doctor. Universidad de la Laguna de España, Facultad de Humanidades y Ciencias Sociales.

Shaeffer Elisa. (2008). *Complejidad Computacional de Problemas y el Análisis y Diseño de Algoritmos.* www.scribd.com/doc/35873176/Complejidad-computacional-dra-elisa-shaeffer. (2010, 15 de julio).

Shannon Claude. (1956). *A Universal Turing Machine with Two Internal States.* Automa Studies, NJ Princeton: University Press pp:157-165.

Shapiro Stewart. (1976). *Incomplete translations of complete logics.* Notre Dame Journal of Formal Logic, Volume 18, Number 2, April, PP 248-250.

________ (2007). *The Objectivity of Mathematics.* Synthese, Volume 156, Number 2, May, PP 339-381.

Shapiro Stewart y McCarthy Timoty. (1987). *Turing Proyectability.* Notre Dame Journal of Formal Logic, Volume 28, Number 4, October, PP 520-535.

Searle John. (1980). *Mind, brains, and programs.* www.bbsonline.org/documents/a/00/00/04/84/bbs00000484-00/bbs.searle2.html (2008, 28 de junio).

Sercovich Armando. (1987). *Charles S. Peirce: Obra Logica-Semiotica.* (1a ed.) (Alcalde Ramón y Prelooker Mauricio Trad.). Madrid, España: Editorial Taurus.

Sieg Wilfried. (1992). *Church Thesis, Consistancy, Formalization, Proof Theory: Dictionary Enties.*

________ (1997). *Step by Recursive Step: Church's analysis of effective calculability.* The Bulletin of Symbolic Logic, Volume 3, Number 2, June PP 154-180.

________ (1997). *Formal System, Church Turing Thesis & Gödel Theorems: Three contributions to the MIT.* Encyclopedias of Cognitive Science.

________ (2004). *Computability Theory,* The Material was Organized for four Seminars. November, University of Bologna.

________ (2005). *Only Two Letters: The correspondence between Herbrand and Gödel.* Bulletin of Symbolic Logic 11(2): PP 172-184.

Sin Autor. (2008). *Página dedicada a Alonso Church, Biografía*, http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Church.html. (2008, 17 de junio).

Soare Robert. (1996). *Computability and Recursion*. Bulletin of Symbolic Logic, Volume 2, Number 3, PP 284-321.

________ (2007). *Computability and Incomputability.*

www.people.cs.uchicago.edu/~soare/History/siena.pdf. (2009, 15 de febrero)

________ (2009) *Turing Oracle Machine, Online computing and three Displacements in Computability Theory.* www.people.cs.uchicago.edu/~soare/History/turing.pdf. (2009, 14 de febrero)

Sominski I.S. (1975). *Método de Inducción Matemática*. (Vega Carlos Trad.). Moscú, URSS: Editorial MIR.

Soler Toscano Fernando (2005). *Modelos formales de explicación en Lógica e inteligencia Artificial.* Tesis doctoral, Universidad de Sevilla,

Volume XXVII, Number 1, PP 1-12.

Soler Toscano Fernando y Nepomuceno Ángel (2008). *Deducción y abducción.* Teorema, Volume XXVII, Number 1, PP 1-12.

Soler Peña, Francisco. (2008). *Los enunciados de base empírica a la luz de los métodos científicos defendidos por Rudolf Carnap y Karl Popper*. www.redalyc.uaemex.mx/pdf/954/95401305.pdf. (2010, 26 de octubre).

Snow Charles Percy. (1959). *The two cultures.* www.csus.edu/indiv/m/mayesgr/sciencehumanvalu/twoculturessnow.htm. (2007, 20 de noviembre).

Steve Allan Russell. (2008). *Tránsitos conceptuales en la Lógica de Bertrand Russell del 1900 al 1927.* Tesis de Magister. Pontificia Universidad Javeriana de Bogotá, Facultad de Filosofía.

Strier Damian. (2002). *Procesos de Auto organización en Sistemas Biológicos.* Tesis de Grado de Doctor. Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales. Departamento de Fisca.

Suppe Frederick. (1979). *La Estructura de las Teorías Científicas.* (Castillo Pilar y Rada Eloy Trad.). Madrid, España: Editora Nacional.

Suppe Patrick. (1968). *Teoría Axiomática de Conjuntos.* (Castillo Hernando Trad.). Cali, Colombia: Editorial Norma.

Tarski, Alfred. (1944). *La concepción semántica de la verdad y los fundamentos de la semántica.* En Valdés Villanueva L.M. (Ed.), La búsqueda del significado (3a ed., pp. 301-338). Madrid España: Editorial Tecnos S.A.

__________ (1951). *Introducción a la Lógica y a la Metodología de las Ciencias Deductivas* (Bachiller T.R. y Fuentes J.R. Trad.). Buenos Aires, Argentina: Editora Espasa-Calpe S.A.

Torretti Roberto. (1998). *El Paraíso de Cantor: La Tradición Conjuntista en la Filosofía de las Matemáticas*. Santiago, Chile: Editorial Universitaria, Universidad Nacional Andrés Bello.

Trajtenbrot B.A. (1977). *Los algoritmos y la resolución automática de problemas*. Moscú, URSS: Editorial MIR.

Trillas Enric. (1998). *La Inteligencia Artificial Maquinas y Personas*. Madrid, España: Editorial Debate.

Trejo Cesar. (1968). *El Concepto de Número*. Washington DC, USA: The Pan American Union.

Trelles Montero Oscar. (2001). *Apuntes de Lógica Modal*. Lima, Perú: Fondo Editorial Pontificia Universidad Católica del Perú.

Turing, A. (1936). *On computable numbers, with application to the Entscheidungsproblem.* www.thocp.net/biographies/papers/turing_oncomputablenumbers_1936.pdf. (2008, 14 de junio).

__________ (1938). *Systems of logic based on ordinals* www.turingarchive.org/browse.php/B/15. (2008, 14 de junio).

__________ (1948). *Intelligent Machinery*: www.alanturing.net/turing_archive/archive/1/132/L32-001.html. (2008, 14 de junio).

__________ (1950). *Computing machinery and intelligence* www.loebner.net/Prizef/TuringArticle.html. (2008, 14 de junio).

__________ (1951). *Can Digital Computer Think* www.turingarchive.org/browse.php/B/5. (2008, 14 de junio).

__________ (1952). *The Chemical Basis of Morphogenesis* www.dna.caltech.edu/courses/cs191/paperscs191/turing.pdf. (2008, 14 de junio).

__________ (1953). *The Chess.* http://www.turingarchive.org/browse.php/B/7 (2009, 20 de julio).

__________ (1954). *Solvable and Unsolvable Problems.* http://www.turingarchive.org/browse.php/C/24. (2010, 29 de julio)

Turner Raymond y Eden Amonon. (2008). *The Philosophy of Computer Science*. http://plato.stanford.edu/entries/computer-science/. (2009, 20 de marzo).

Valdés Villanueva, Luis. (1998). *Ensayos de semántica y Filosofía de la lógica.* Madrid, España: Editorial Tecnos.

Von Bertalanffy. (2006). *Teoría General de los Sistemas.* (Almela Juan Trad.). México DF, México: Fondo de Cultura Económica.

Von Newmann John. (1958). *The Computer and the Brain.* (1a ed.) Yale USA: Yale University Press.

Von Newmann y Oscar Morgenstern. (1953). *Theory of Games and Economic Behavior.* Princeton, USA: Princeton University Press.

Von Wright Georg Henrik. (1970). *Ensayo de Lógica Modal.* (Demarchi Atilio Trad.) Buenos Aires, Argentina: Rueda Filosófica.

Wang Hao. (1974). *A Logical Journey from Gödel to Philosophy*. Massachusetts, USA: Massachusetts Institute of Technology.

__________ (1991). *Reflexiones sobre Kurt Gödel.* (Castillo Criado Pilar Trad.). Madrid, España: Alianza Editorial.

Wiener Norbert (1956). *I Am a Mathematician.* (1a ed.) Garden City, New York, USA: Doubleday & Company, Inc.

__________ (1958). *Cibernética y Sociedad.* (1a ed.) (José Novo Trad.) Buenos Aires Argentina: Editorial Sudamericana.

__________ (1963). *God and Golem Inc.* Cambridge, Massachusetts: The Massachusetts Institute of Technology Press.

Wittgenstein Ludwig. (1973). *Tractatus Logico Philosophicus.* (Jacobo Muñoz e Isidoro Reguera Vers.). Madrid, España: Alianza Editorial.

__________ (1978). *Observaciones sobre los fundamentos de la matemática.* (Reguera Isidoro Trad.). Madrid, España: Alianza Editorial.

__________ (1997). *Observaciones sobre la Filosofía de la Psicología.* (Segura Felipe trad.). México D.F., México: UNAM.

__________ (2000). *Sobre la Certeza.* (Prades Josep y Raga Vicent trad.). Barcelona, España: Editorial Gedisa.

Zadeh L.A. (1982). *Test-score semantics for natural languages.* short6pgs-test-score_semantics_for_natural_lenguages-1982.pdf. (2007, 22 de diciembre).