LENGUAJES FORMALES
Y
LENGUAJES COMPUTACIONALES

MAGDALENA PRADILLA RUEDA

LENGUAJES FORMALES
Y
LENGUAJES COMPUTACIONALES

Bogotá - Colombia
2015

Publicación sometida a pares académicos (*Peer Review Double Blinded*).

ISBN 978-958-5447-09-7

Diagramación y corrección: Editorial TEMIS S.A.
Calle 17, núm. 68D-46, Bogotá.
www.editorialtemis.com
correo elec. editorial@editorialtemis.com

Diseños y gráficos originales de Editorial TEMIS S.A.

Hecho el depósito que exige la ley.

PRÓLOGO

El lenguaje aquí es tomado en el sentido del Lenguaje Formal en lógica y en el sentido del Lenguaje Computacional en Informática, con el fin de encontrar la génesis de éste es decir la formación, evolución y transformación del primero en el segundo. Con este objetivo, fue necesario recorrer de manera exhaustiva una serie de obras según la sucesión de tiempo y de evolución de conocimiento de los dos tipos de lenguajes que nos llevan a esbozar los momentos de esa génesis. Así mismo, este recorrido tiene como horizonte la comprensión del modo de formación de estos lenguajes, que existe de manera histórica y epistemológica.

Así se ha realizado este estudio a la vez preciso, concreto y claro. Preciso, en el sentido que se apoya en una documentación de primera mano; concreto porque respeta la especificidad de las obras, de los temas y de las épocas; claro porque muestra los encadenamientos de las ideas y la emergencia de los problemas que estos suscitan.

Evidentemente la abundancia del material y los temas que tocan estos lenguajes obligaron a señalar la imposibilidad de su exhaustividad y a seleccionar los temas más dicientes, a limitar los campos teóricos y a distinguir momentos significativos y fructíferos para el desarrollo de su conocimiento. Estas delimitaciones se escogieron de manera juiciosa, desde un doble punto de vista epistemológico y pedagógico.

Este estudio es el resultado del proyecto de investigación "Desarrollos epistemológicos del lenguaje lógico-matemático que fundamenta la génesis del lenguaje de computación", correspondiente a la línea de investigación Lógica Matemática del Grupo de Investigación de Matemáticas y Ciencias de información. Investigación que ha sido concluida y que además ha tenido como productos, artículos publicados y ponencias en diferentes eventos académicos[1].

[1] "Bases epistemológicas de los lenguajes de las máquinas lógicas": Capítulo de libro: Matemáticas en tecnología y educación: una perspectiva republicana, Bogotá, Edit. Temis, 2014; "Contexto y elementos de una sintaxis del lenguaje lógico", artículo en la Revista, *Ingeniería, matemáticas y ciencias de la información*, Corporación Universitaria Republicana, vol. 1, núm. 1, 2014; "Alan Turing, su obra y los efectos sobre la calculabilidad", artículo en la Revista, *Ingeniería, Matemáticas y ciencias de la información*, Corporación Universitaria Republicana, vol. 1, núm. 2, 2014; "Lenguajes de programación", Ensayo del Grupo de Semilleros, diciembre de 2012; "Formalización y Lenguajes Lógicos", Ponencia en el I Simposio de Ingenierías, Matemáticas y Ciencias de la Información, noviembre 2013.

ÍNDICE GENERAL

CAPÍTULO III

SINTAXIS

CAPÍTULO IV

SEMÁNTICA

CAPÍTULO V

LENGUAJES DE COMPUTACIÓN
Y LENGUAJES FORMALES

INTRODUCCIÓN

> "Las ciencias abstractas necesitan vivamente un medio de expresión que permita a la vez prevenir los errores de interpretación y evitar las faltas de razonamiento. Los unos y las otras tienen su causa en la imperfección del lenguaje…".
>
> GOTTLOB FREGE, Que la ciencia justifique el recurso a una ideografía[1].

WITTGENSTEIN declara que "la lógica es una invención muy importante"; RUSSELL, a su vez, afirma que cualquier problema sometido a un análisis y elucidación lógica puede llegar a soluciones consecuentes y claras; también afirma que la lógica moderna puede expandir nuestra imaginación abstracta y proporcionar un número infinito de hipótesis aplicables a cualquier campo del saber. En ese sentido se puede pensar que la *lógica matemática* presenta un salto en el conocimiento, porque muestra un conjunto amplio de problemas susceptibles de recibir una solución; se plantea, entonces, una "revolución" en este saber, por el grado de claridad, precisión y por la apertura de perspectivas y horizontes novedosos que confiere.

[1] Publicado en el *Zeitschrift für Philosophie und philosophische kritik* (81) en 1882. En *Écrits logiques et philosophiques*, trad. francesa por Claude Imbert, Paris, Edit. Seuil, 1970.

Es así como el objetivo principal de este estudio corresponde a una parte de esa nueva lógica que son los *lenguajes formales,* como resultado de la formalización de la lógica y sus relaciones fundantes con los *lenguajes computacionales.*

El primer capítulo: *Génesis de los lenguajes formales,* muestra una visión general de las investigaciones de la lógica de finales del siglo XIX y comienzos del XX, marcados por los *fundamentos de matemáticas* que contribuyen al desarrollo de la lógica y esta, a su vez, es una contribución en la investigación de dichos *fundamentos*. Así, los descubrimientos de Cantor tocan la comparación de colecciones (1874), el método de la diagonalización y la teoría de conjuntos. Teoría, cuyo conjunto de *infinitos no enumerables* presenta antinomias o *paradojas*, ampliadas también por la deficiencia de los sistemas de deducción, que llevan a *reformulaciones* de la lógica existente.

El segundo capítulo: *Formalización*, en el cual el acento está marcado sobre los diversos aspectos y características de esta, su resultado en los sistemas formales y lo que se nos es dado: el *lenguaje*. Lenguaje que conlleva un sistema de inferencia (sintaxis) y un sistema de interpretación (semántica), es decir, el paso entre *probar* e *interpretar.*

El tercer capítulo: *Sintaxis*. En éste se presentan las "reglas de juego" sobre las cuales esta se realiza y las diferentes propiedades a las que la sintaxis se acredita en la formación de expresiones bien formadas. Igualmente, el planteamiento de la diferencia entre aquello que es elemental y lo que no lo es (ideal).

El cuarto capítulo: *Semántica*. El objeto de éste es examinar los sentidos posibles de una expresión lógica; es decir, la *interpretación* en un lenguaje más comprensible, lo que es posible estableciendo una correspondencia entre los dos y, por tanto, de una evaluación. Así, podemos decir que una expresión puede presentarse de varias maneras (diferentes sentidos), pero que conducen a una sola denotación, referencia o valor. Lo que nos llevaría a pensar que si el cálculo de un valor conduce a un "empobrecimiento" del contenido operacional del objeto, porque tenemos dos expresiones de presentarlo: "$((1+1)+1)+1$" y "2×2", que llevan al mismo valor en N, sin embargo el significado operacional es distinto. Entonces que se pierde en identificación de expresiones teniendo la misma referencia o denotación, se gana en libertad de interpretación, y viceversa, pero esto va a depender de los requerimientos propuestos en la interpretación.

La introducción de las nociones de interpretación y los diferentes métodos como variables de función y de modelo de TARSKI, permiten el razonamiento de una semántica. Bien que los métodos sintácticos son naturalmente adaptados en la lógica, son los métodos semánticos que aparecen más inteligentes y evolucionados, por su flexibilidad.

El quinto capítulo: *Lenguajes de computación y lenguajes formales*, basado sobre un rol dependiente de la informática a la lógica. La concepción del programa, algoritmo, se plantea como la construcción de expresiones bien formadas del lenguaje computacional, a partir de reglas de deducción, lo cual conlleva a probar las expresiones, como se realiza en lógica; y ejecutar un programa significa normalizar esta prueba. Es aquí donde intervienen los sistemas de inferencia y la semántica de la lógica. Así mismo, la construcción de lenguajes computacinoales como interpretadores, compiladores y optimizadores toman sentido en la teoría de la recursividad.

Capítulo I

GÉNESIS DE LOS LENGUAJES FORMALES

1. Introducción

En el curso de los últimos ciento cincuenta años las investigaciones en *lógica matemática* han evolucionado de una manera muy importante, cruzándose en los primeros tiempos con las ideas de los *fundamentos de matemáticas*. El presente capítulo muestra la problemática referente a la teoría de conjuntos de Cantor, sus efectos sobre el sistema matemático y las deficiencias del sistema deductivo y de razonamiento. Igualmente, se muestran las corrientes "reformulacionistas" de la lógica que van a hacer frente a los defectos y paradojas del sistema. Por último, se plantean los conceptos base de la formalización y constructividad de los sistemas matemáticos, como resultado de los planteamientos de los matemáticos y lógicos.

2. Teoría de conjuntos de Cantor

Uno de los puntos de interés de Cantor[1] es la manera de determinar *si una colección es menos numerosa que otra,*

[1] Georg Ferdinand Ludwig Philip Cantor (1845, Saint-Petersburg – 1918, Halle), matemático alemán, conocido por ser el creador de la teoría de conjuntos. Establece la importancia de la biyección entre los conjuntos, define los conjuntos infinitos y los conjuntos bien ordenados;

igualmente numerosa o más numerosa que ella. Se trata de mostrar colecciones finitas, lo que se puede hacer por medio de la formación de pares, asociando a un elemento de una de estas colecciones un elemento de la otra colección; es decir, realizando una correspondencia biunívoca entre elementos (uno a uno: 1-1). Si esta es posible, quiere decir que las colecciones son "iguales en número", o que tienen el mismo número *cardinal*.

Esta correspondencia puede tener un efecto *paradójico*, porque si los cuadrados de enteros positivos pueden ser puestos en correspondencia 1-1 con todos los enteros positivos, se puede presentar como *contradictorio*, según el axioma de EUCLIDES, por el cual, *el todo es más grande que cada una de sus verdaderas partes* (*partes que no son idénticas al todo*) (KLEENE, 1971, pág. 184)[2]. Contradicción, que no excluye la correspondencia 1-1 de una colección infinita con una parte de otra colección infinita, cuya posibilidad conduce a formar los pares de otra manera, totalidades, que en este caso, se correspondan de una manera biunívoca.

Así, por ejemplo, en la cadena de *números naturales 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,…,* las colecciones susceptibles de hacer la correspondencia biunívoca con los números naturales se llaman *infinitas numerables.* Un conjunto infinito enumerable, presenta sus elementos en una lista infinita, lo que conduce a una correspondencia 1-1 entre este conjunto y los números naturales.

---

prueba que los números reales son más numerosos que los enteros naturales. El teorema de CANTOR implica la existencia de una infinidad de infinitos, define los números cardinales y los ordinales y su aritmética.

[2] KLEENE llama la atención sobre la posibilidad de esta correspondencia presentada por Galileo, Bolzano, Plutarco y otros como G. d'Ockham.

En este sentido, se pueden llamar *conjuntos enumerables infinitos*, los números naturales, enteros positivos, cuadrados de enteros positivos, los números racionales (0, 1, -1, 1/2, 1/3, -1/2, 2, -2, -1/3, 1/4,…); los números reales algebraicos o el conjunto de números reales que son la raíz de ecuaciones (ecuaciones polinomiales) en una variable x con coeficientes enteros.

Sin embargo, al lado de estos *conjuntos enumerables infinitos*, existen otros conjuntos, es decir, *conjuntos infinitos no enumerables*, que es imposible poner en correspondencia con los números naturales. Cantor va a demostrar que existen potencias superiores a la enumerable, en donde el método de demostración podía explicarse para cualquier potencia (ROUILHAN, 1996, pág. 23)[3]. El método utilizado por CANTOR para mostrar tales conjuntos, es el llamado de la *diagonal*. Vamos a retomar la interpretación de KLEENE (1994, págs. 188-191), así:

Uno de los casos de conjuntos infinitos no enumerables, es el conjunto de *funciones aritméticas a una sola variable*; por ejemplo, las funciones de una sola variable a que recorre el conjunto de los números naturales, con valores en el conjunto de los números naturales. Ejemplos: $a^{2,}$ $3\,a + 1, 5$ (función constante), a (la función idéntica), $[\sqrt{a}]$ (el entero más grande $\leq \sqrt{a}$). Para demostrar que el conjunto de estas funciones no es enumerable, se supone una enumeración dada: $f_0\,(a), f_1\,(a), f_2\,(a), \ldots$ de funciones aritméticas. Se trata de construir una función aritmética $f\,(a)$ que sea diferente de cada función que pertenece a la enumeración dada, lo que

[3] ROUILHAN nos precisa que CANTOR se tiene como el inventor del método, pero DU BOIS-REYMOND lo había utilizado veinte años antes que CANTOR.

probará que esta enumeración no es una enumeración de todas las funciones aritméticas de este tipo. Para construir $f(a)$, se presentan los valores de las funciones $f_0(a)$, $f_1(a)$, $f_2(a)$,… en una tabla infinita:

A	0	1	2	…
$f_0(a)$	$f_0(0)$	$f_0(1)$	$f_0(2)$	…
$f_1(a)$	$f_1(0)$	$f_1(1)$	$f_1(2)$	…
$f_2(a)$	$f_2(0)$	$f_2(1)$	$f_2(2)$	…
…	…	…	…	…

Así, se define $f(a)$ como la función cuyos valores serían obtenidos de la diagonal (en gris) de la tabla, es decir, que

$$f(a) = fa(a) + 1$$

Esta función no figura en la enumeración anterior $f_0(a)$, $f_1(a)$, $f_2(a)$,… porque difiere de $f_0(a)$, según el valor que toma sobre el 0, de $f_1(a)$ según el valor que toma sobre el 1,… etc. Por ejemplo, al retomar las funciones expuestas anteriormente, si

$$f_0(a) = a^2, f_1(a) = 3a + 1, f_2(a) = 5, f_3(a) = a, f_4(a) = [\sqrt{a}],\ldots$$

entonces $f_0(0) = 0, f_1(1) = 4, \quad f_2(2) = 5, f_3(3) = 3, f_4(4) = 2,\ldots$

pero $f(0) = 1, f(1) = 5, \quad f(2) = 6, f(3) = 4, f(4) = 3$

Si el argumento se realiza de otra manera y se supone que la función $f(a)$ figura en la enumeración anterior $f_0(a)$, $f_1(a)$, $f_2(a)$,…; y supongamos que existe un número natural *p,* de manera que existe un número natural *p* tal que

$$f(a) = fp(a)$$

Para todo número natural *a*. Al sustituir el número *p* por la variable *a* en las anteriores ecuaciones, se obtiene:

$$f(p) = fp\,(p) = fp\,(p) + 1$$

Lo que es imposible porque el número natural $f_p\,(p)$ no es igual a sí mismo aumentado de una unidad y, por tanto, se puede decir que el conjunto de todos los números reales es *no enumerable*.

A partir de su *m*étodo, CANTOR desarrolla una *Teoría de conjuntos abstractos*, en la cual trata de conjuntos de diferentes tipos[4]. Él define (CANTOR, 1895, pág. 481) *conjunto* como "… toda colección *M* de objetos *m* diferenciados por nuestra percepción o nuestro pensamiento (esos objetos serán llamados *elementos de M*)"[5].

Él nos da las siguientes precisiones:

• $m \in M$ significa que *m* pertenece a *M*, o bien, que *m* es un miembro de *M* o es un elemento de *M*.

• $m \notin M$ significa que *m* no pertenece a *M*.

• $M_1 = M_2$ significa que dos conjuntos M_1 y M_2 son idénticos si tienen los mismos elementos.

[4] GEORG CANTOR, (1895-1897). "Beitrage zur Begrundung der transfiniten", *Mathematische Annalen*, *46* (1895), 481-512; *49* (1897); reimpreso en 1932. Traducido al francés: *Sur les fondements de la théorie des ensembles transfinis,* por F. MAROTTE. In Mémoires de la Société des Sciences Physiques et Naturelles de Bordeaux, reeditado por Gabay, Paris, 2000. Estas publicaciones fueron las últimas contribuciones significativas de la teoría de conjuntos. El primer artículo define los conjuntos, subconjuntos, etc., de la manera que se conoce hasta hoy, la aritmética de cardinales y ordinales. En el segundo artículo define la teoría de conjuntos bien ordenados y la de los ordinales.

[5] Citado por KLEENE (1998, pág. 191).

• Un conjunto finito puede ser descrito al colocar la lista de sus elementos entre paréntesis (el orden es indiferente); con puntos de suspensión, se puede sugerir un conjunto infinito. Por ejemplo: [1, 2, 3] es un conjunto de tres elementos y [0, 1, 2, 3,…] es el conjunto de los números naturales.

• Un conjunto M_1 es un subconjunto de un conjunto M y se escribe $M \supset M_1$ si todo elemento de M_1 es elemento de M. Así, por ejemplo, el conjunto de tres elementos [1, 2, 3] tiene ocho subconjuntos: [ ], [1], [2], [3], [1, 2], [1, 3], [2, 3], [1, 2, 3].

Sabemos que el primero es el *conjunto vacío,* los tres siguientes son los *singletones* (un solo elemento).

• La noción de *cardinal M* es el abstracto de M y de otros conjuntos susceptibles de ser puestos en correspondencia 1-1 con el conjunto M[6]. Cantor explica que al no tener en cuenta la naturaleza de los elementos del conjunto M, ni su orden, se presenta la *potencia* o *cardinal de M*.

Para Russell en 1902 y Frege en 1884, el *cardinal M* se define *como el conjunto de todos los conjuntos* N *y que pueden poner en correspondencia* 1-1 *con* M. Lo que tiene por consecuencia que los *cardinales* pueden ser tomados como *individuos* en un universo cuyos miembros son *conjuntos*.

Para todo conjunto M, 2^M es el conjunto de todos los subconjuntos de M, pero según lo que hemos planteado con el método de la *diagonal* 2^M es *no* enumerable; o sea, que $2^M \neq M$.

[6] De esta manera, se llega a comprender el concepto de *dos*, procediendo por abstracción de la existencia en el mundo o en el cuerpo: dos manos, dos orejas, dos piernas y lo que es *dos* en sí, no es importante.

Anotaciones. La teoría cantoriana de *conjunto* se desarrolla sobre una base de carácter lógico, porque los conjuntos se pueden concebir como *extensiones de conceptos*, aunque CANTOR plantea que los conjuntos existen independientemente de los conceptos. La noción de conjunto en general, sugiere la operación de *totalizar* y por esta vía introduce el *infinito actual* diferente del *infinito potencial*.

Los matemáticos de la época toman una actitud reservada frente a estas nociones; una es la posición de GAUSS (1831), que prohíbe utilizar *dimensiones infinitas* como realidades finitas; otra es la posición de BROUWER, el cual afirma que el concepto de conjunto infinito, incluso el potencial, requiere otra clase de lógica diferente a la clásica.

3. LAS PARADOJAS

La *teoría de conjuntos* de CANTOR termina por imponerse, a pesar de sus resistencias, hacia 1890-1900, y llega a considerarse como la base del edificio matemático. Así, la aritmética de números finitos, con la cual se había reconstruido las otras partes de las matemáticas, era, a su vez, un caso particular simple de la teoría de conjuntos, el de los conjuntos enumerables. Pero surgen, posteriormente, al interior de la teoría, *antinomias* o *paradojas*[7]; es decir, parejas de teoremas que son contradictorios. A preguntas como "¿el conjunto de todos los conjuntos que no se contienen ellos mismos como elementos, se contiene él mismo como elemento?", se plantean tanto respuestas negativas como positivas. Lo cual presenta una gravedad excepcional para las matemáticas, en

[7] Paradoja viene del griego *paradoxos* (*para*: contra y *doxa*: opinión): opinión contraria a la opinión común.

su fundamentación: si una teoría tiene garantizada su validez por la *coherencia formal*, cualquier fisura en su edificio era suficiente para comprometer su totalidad.

Por tanto, aunque la existencia de paradojas es abundante y remonta a la Antigüedad (megáricas, eleáticas,…), la mayoría son ilustraciones diferentes de unas pocas paradojas estructurantes. Estas paradojas pueden ser planteadas desde dos puntos de vista: las paradojas *lógicas* y las *epistemológicas* o *semánticas*.

A) *Paradojas lógicas*

Las paradojas lógicas, para Russell (1906)[8], se caracterizan porque "si ellas deben ser (*are to be*) resueltas, debe hacerse (*must*) por una modificación de los planteamientos lógicos corrientes" y son aquellas que tocan directamente la inconsistencia del sistema de lógica corriente. Russell clasificaba estas paradojas según nociones lógicas o matemáticas: ordinales, cardinales, clases, relaciones, funciones, proposiciones…

Se tienen como lógicas las paradojas como las de Burali-Forti (1897, teoría cantoriana de los ordinales), Cantor (1899, publicada por Zermelo en 1932), Russell (1902, publicada en *The principles of mathematics*, 1903)…, entre otras. Nos vamos a referir a la paradoja de Russell y la de Cantor de manera ilustrativa de las paradojas lógicas.

Paradoja de Russell de 1902: Russell examina la forma en la que Cantor demuestra que no existe el número cardinal

[8] Bertrand Russell, (1906). "Les paradoxes de la logique". *Revue de métaphysique et de morale*, *14* (1906-1907). Citado por Rouilhan (1966), Paris, puf, pág. 290.

más grande y que al aplicar esta demostración al "número de todas las cosas que existen en el mundo", descubre la paradoja. Así que él considera "una clase muy particular" aquella de las "clases que no son miembros de ellas mismas" y RUSSELL se pregunta si esta clase es miembro de ella misma o no, porque cada término de la alternativa conduce igualmente a su contrario y, por tanto, hay una contradicción.

En su demostración, él va a presentar cómo ninguna aplicación f: $u \rightarrow \beta(u)$, no puede ser *suryectiva* en el caso de la clase de todas las clases. Porque si para toda clase, existe una suryección a saber f: $u \rightarrow \beta(u)$, por ejemplo, en la aplicación determinada $f(x) = x$ si x es una clase y $f(x) = \{x\}$ si no, lo cual es una contradicción.

A nivel más sencillo, se puede explicar como "el conjunto de todos los conjuntos que no se contienen ellos mismos como elemento". Así, sea S este *conjunto*. Supongamos que:

• S se contiene a sí mismo, pero en virtud de la definición de S, S no se contiene a sí mismo.

• Entonces, por reducción al absurdo (se rechaza la hipótesis), se demuestra: S no se contiene a sí mismo.

• Pero, según la definición, S se contiene a sí mismo.

• Lo cual constituye una contradicción, es decir, una paradoja.

RUSSELL propone una versión popular a esta paradoja: "El barbero de un pueblo rasura a todas las personas que no se rasuran por sí mismas. Pregunta: ¿el barbero se rasura él mismo?". Es una paradoja que dice *falso* (porque lógicamente es contradictoria).

La paradoja de CANTOR (1899): relativa a la noción de número cardinal. En 1883 ya había demostrado que no existe el número cardinal más grande. Cantor encuentra este mismo resultado por otra vía y demuestra que existen potencias superiores a aquella de *enumerable* y anota que el método de demostración podía aplicarse a cualquier potencia y prueba corolariamente que existen potencias superiores a aquella del continuo, por el método de la diagonal, ya expuesto.

De manera esquemática, Cantor demuestra: sea T el conjunto de todos los conjuntos. 2^T es el conjunto de todos los conjuntos; entonces, $2^T \subseteq T$, *entonces* M *no* $> N$; porque 2^T *no* $> T$. Pero, en virtud del teorema de Cantor: $2^T > T$, existe entonces una contradicción y, por tanto, una paradoja.

Anotaciones. Rouilhan (1996, pág. 20) nos explica que estas paradojas son la prueba *apagógica* (prueba por el absurdo), de su resolución: la paradoja del 'barbero' es la *prueba por el absurdo* de la hipótesis de la existencia de un tal barbero (que no existe). En la paradoja del conjunto de Russell, es la hipótesis de la existencia de un tal conjunto, que no existe, lo mismo que el conjunto de todos los conjuntos de Cantor. Lógicamente, se da una propiedad característica de ese conjunto, una condición necesaria y suficiente para que un objeto sea un elemento del conjunto, que no es lo bastante precisa para ser comprensible. La información dada no va en correspondencia con la *doxa,* o evidencia que se plantea. ¿Qué falta, si se trata de un *conjunto* (objeto abstracto, no concreto), en el campo de la abstracción para que sea legítimo? Lo que falta, es una compresión más profunda de la noción de conjunto, y la determinación correlativa de un principio limitado en la abstracción para controlar su uso. Así mismo, se observa que los conjuntos presentados en estas paradojas son demasiado "grandes" como el conjunto T de todos los conjuntos en la paradoja de Cantor, lo que puede llevar al libre uso de concepciones intuitivas (base de la definición cantoriana de *conjunto*) que conduce a graves dificultades, como se ha visto en las paradojas expuestas.

Así, la disolución de estas paradojas requiere conocer claramente las fórmulas regulares primitivas, a las cuales se les puede aplicar el principio de abstracción y las fórmulas a las que se hace excepción; un procedimiento que determine en dónde comienza, cómo y el porqué de esta manera de proceder, lo mismo que los esquemas de deducción definidos en el procedimiento.

B) *Paradojas epistemológicas o semánticas*

Una paradoja semántica es relativa a diversas nociones semánticas que relacionan el lenguaje con el mundo de hechos reales; es decir, la noción de verdad *verificacional* de un enunciado o la noción de *satisfacción* de una fórmula, por una o varias entidades. Estas paradojas dependen, entonces, de hipótesis extralógicas, las cuales son posibles de reformular y ser corregidas; la teoría lógica en el marco donde ellas aparecen no está en juego.

Este tipo de paradojas incluyen las de *Richard, Berry, del Mentiroso*…, entre otras. Nos vamos a referir a la paradoja de RICHARD.

En su carta de 1905 a la *Revue générale des sciences pures et appliquées,* J. RICHARD expone su paradoja y propone una solución, así:

Sea un lenguaje en el cual se pueda formular o definir las propiedades puramente aritméticas de los números cardinales. Consideremos así las definiciones que pueden ser establecidas por ese lenguaje. Es claro que para evitar la circularidad o la regresión al infinito, ciertos términos se refieren a propiedades aritméticas que no pueden ser definidas explícitamente. Podemos admitir que comprendemos inmediatamente lo que se presenta como definición; por ejemplo: "un entero es divisible por otro entero", "un entero es el producto de dos enteros", etc. La propiedad de ser un número primo puede definirse así: "no es divisible sino por el mismo y por 1", y así sucesivamente.

Entendemos que cualquier definición es una cadena finita de letras del alfabeto y de términos que pueden ser un espacio o las 27 letras del alfabeto, con la condición que el espacio no figure ni en posición inicial, ni en la posición terminal. En este caso se puede ordenar las definiciones en serie: una definición precederá a otra, si el número de letras de la primera es inferior al de la segunda, y si dos definiciones tienen el mismo número de letras, ellas serán ordenadas en orden alfabético de sus letras. En consecuencia, se podrá asignar a cada definición un entero único que representará el número del lugar que ocupa esta definición en la serie. Por ejemplo, la definición que comprende el más pequeño número de letras llevará el número 1, la definición que le sigue inmediatamente en la serie llevará el número 2, y así sucesivamente. El resultado es una enumeración D_0, D_1, D_2,… de las definiciones descritas.

Como a cada definición se asocia un entero único, se puede, en ciertos casos, que el entero asociado a una definición posea precisamente la propiedad que esta definición determina. Por ejemplo, si la definición D_7 dice: "número divisible por él mismo y por uno", se ve claramente que "7" posee él mismo la propiedad descrita. Al contrario, supongamos que la definición: "ser el producto de un entero por sí mismo” es la definición D_{15}, es claro que el número 15 no tiene esa propiedad. Se puede decir que para este tipo de definiciones tiene la propiedad de *no correspondencia* y para las otras definiciones, como la D_{15}, tienen la propiedad de *correspondencia*. De manera general, se define: “*x* es *no correspondencia”* como una manera abreviada de decir: “*x* no posee la propiedad designada por la definición a la que está asociada, en el conjunto ordenado en serie de las definiciones”; y “*x* es *correspondencia”* en el caso contrario; es decir, las definiciones que corresponden al conjunto enumerable, y efectivamente enumerado, *E.*

Pero la definición de ser *no correspondencia* describe manifiestamente una propiedad numérica de los enteros; ella, entonces, forma parte de la serie de definiciones establecidas aquí y, por consecuencia, asociada a un número de orden que es un número entero a la vez, lo podemos llamar *e.*

En este momento, nos podemos preguntar si ¿“*e* es *correspondencia”?* Encontramos aquí una contradicción, en efecto, *e* es *no correspondencia* si y solo si *e* no posee la propiedad designada por la definición a la cual él está asociado. Es decir, *e* es *no correspondencia* si y solo si *e* no es *no correspondencia*, de manera que “*e* es *no correspondencia”* es verdadero y falso a la vez, es decir, que hay una *contradicción.*

Anotaciones. A partir del análisis de la paradoja de Richard, surge la idea de “principio de círculo vicioso”, que Poincaré (1905-1906) va a retomar posteriormente. Por medio de éste, introduce una distinción en la noción de definición: aquella de *D*, pero que no puede valer solamente para definición de un elemento de *E,* porque ello implica la noción del conjunto *E* mismo. Tal definición es ilegítima, *paradójica*. Este es el origen de la paradoja: la reflexión sobre sí, es autorreferencia y es un círculo vicioso.

Igualmente, se puede decir que la paradoja se debe a la falta de claridad en el uso de las reglas de juego; se dejó de lado una hipótesis tácita pero esencial que toca *la serie ordenada de las definiciones*. Se había convenido considerar las definiciones de las propiedades puramente aritméticas de los enteros, propiedades que se formulan con la ayuda de conceptos como la adición aritmética, la multiplicación,..., y entonces, se pide, sin mucha claridad, aceptar en la serie una definición que contiene una referencia a la *anotación* utilizada para formular las propiedades aritméticas. Más precisamente, el de la propiedad de *no correspondencia*, que no forma parte de la serie inicialmente establecida, porque esta desarrolla conceptos *metamatemáticos*. Lo que indica la necesidad de distinguir las aserciones interiores a la aritmética (que no hacen referencia a ningún sistema de anotación) y las aserciones correspondientes a un cierto sistema de anotación que posibilita la sistematización de la aritmética (*metamatemática*).

4. Reformulación de las matemáticas y de la lógica

Las paradojas lógicas y las semánticas muestran dificultades inherentes a la teoría lógica y matemática realizadas sobre una base *intuitiva*; es decir, faltas en la formalización y rigurosidad tanto en los planteamientos como en las demostraciones, presentadas en la base cantoriana de conjuntos, de clases, entre otros. Estas dificultades van más allá de las matemáticas y atacan su fundamento, lo que conduce a preguntarse cómo modificar la teoría de manera que se pudiera impedir las contradicciones. Así mismo, lleva a preguntarse sobre los puntos particulares que conducen al error, sobre los métodos de construcción y de razonamiento que habían parecido siempre convincentes antes del surgimiento de las paradojas.

Una de las problemáticas más vastas planteada por las paradojas, era la relativa a la naturaleza de las matemáticas, al alcance de sus métodos y particularmente a los métodos deductivos, llamados *axiomáticos*, lo cual nos lleva a preguntarnos sobre estos métodos y sus alcances.

A) *Métodos axiomáticos*[9]

El método axiomático tiene como objeto presentar, a la base de la teoría que se pretende demostrar, algunas proposiciones (axiomas) y reglas que se han escogido por ser las más "dicientes", generales, simples y sin la condición de ser demostradas, sobre las cuales se realiza la deducción de todas las proposiciones necesarias para demostrar la verdad de la teoría en cuestión. Para BLANCHÉ (1990, pág. 11), la axiomática es la forma perfeccionada, que toma hoy una teoría deductiva. Esta descripción se aplica tanto a los griegos del siglo IV, como a los lógicos y matemáticos del siglo XXI.

EUCLIDES reconoce el método, en dos disciplinas: la aritmética y la geometría, en las cuales se propone distinguir los principios o "comunes" de aquellos que no lo son. Igualmente, este método se encuentra, por una parte, en la "silogística" de ARISTÓTELES y, por la otra, en una forma elemental del cálculo proposicional de CHRYSIPPE, lo mismo que

[9] *Axiomático,* viene del griego *axioma* (proposición).

en la “estática” inaugurada por ARQUÍMEDES. En el siglo XIX, el instrumento axiomático se abre a una renovación, con la *Begriffsschrift* de FREGE (1879), que procede en la tradición euclidiana a la estructuración del “cálculo de predicados”. Un recurso análogo pero renovado se encuentra en los *Principia mathematica* de WHITHEAD y RUSSELL (1910-1913), renovación que llega a la obra de HILBERT en *Grundlagen der geometrie* (1971).

De esta manera, en la Antigüedad, se conoce la axiomática con los *Elementos de Euclides* (330-320 a. C.). Este método permitió sistematizar muchos de los conocimientos geométricos y matemáticos. A manera de síntesis, esta axiomática consiste en:

• Presentar las *definiciones* de ciertos términos primitivos, tales como *punto, recta, plan*…, definiciones que sugieren al lector el sentido de estos términos.

• Presentar los *axiomas* o *postulados*, que son proposiciones relativas a los términos primitivos juzgados aceptables sobre la base de los términos.

• Definir nuevos términos definidos por referencia a los términos primitivos.

• Definir nuevas proposiciones, llamadas *teoremas*, deducidos de los axiomas, según reglas de lógica.

La axiomática del tipo euclidiano es llamada *axiomática material*. El desarrollo de esta axiomática en la geometría produjo una gran impresión sobre los pensadores: los axiomas de la geometría, en pequeño número, soportan todo el peso del número infinito de proposiciones que permiten demostrar los axiomas. Además, durante más de dos mil años se consideró que estos axiomas eran *verdades del es-*

pacio y por esto garantizaban la verdad y la consistencia de todos los teoremas. Era, entonces natural, preguntarse si se podía establecer sobre la base de los fundamentos de la axiomática otras ramas del pensamiento.

En este sentido, un clima de fundamentación surge, en el cual los dominios del pensamiento matemático podían ser provistos de un conjunto de axiomas para el desarrollo sistemático de la totalidad infinita de las proposiciones verdaderas de cada dominio. Se asiste de esta manera, a una intensificación de la investigación matemática. Muchos de los problemas matemáticos fueron resueltos, otros se demostraron lógicamente imposibles y otros permitieron la creación de nuevos sectores del estudio matemático.

Uno de los desarrollos más significativos tuvo que ver con la búsqueda de la solución de uno de los *postulados de Euclides*, que había sido reputado menos evidente que los otros: es el *quinto* o *postulado de las paralelas*; así Euclides dice:

"Postúlese… Y que si una recta al incidir sobre dos rectas hace los ángulos internos del mismo lado menores que dos rectos, las dos rectas prolongadas indefinidamente se encontrarán en el lado en el que están los [ángulos] menores que dos rectos".

Una de las interpretaciones:

Por un punto P *dado no situado sobre la recta dada* l*, se puede hacer pasar una recta y una sola paralela a* l (*que no corte* l *en ningún punto*).

Desde la Antigüedad, se trató de demostrar este postulado a partir de los otros teoremas de Euclides, lo que lo hubiera transformado en teorema, pero se demostró (Gauss, Bolyai, Lobatchewski y Riemann) que era imposible deducir el *axioma de paralelas* a partir de los otros axiomas. Este resultado fue de gran importancia en el caso que es posible dar una demostración de la *imposibilidad de demostrar* ciertas proposiciones en un sistema dado. Por otro lado, los matemáticos tomaban conciencia del hecho de que EUCLIDES no representaba la verdad última en materia de geometría, porque se podía construir nuevos sistemas geométricos utilizando axiomas diferentes de aquellos escogidos por EUCLIDES e incompatibles con ellos.

Anotaciones. Tanto el resultado de las paradojas como la falta de consistencia en el sistema axiomático utilizado desde la Antigüedad, plantea como consecuencia que la concepción de las matemáticas como "ciencia de la cantidad" era totalmente inadecuada. Deja abiertos una serie de problemas como aquellos relativos a la *estructura deductiva*, en donde la validez de la inferencia matemática es absolutamente independiente del significado que podrían tener los términos o las expresiones contenidas en los postulados o axiomas. Así mismo, la dificultad en la interpretación de los axiomas y en su aplicación a un sistema de objetos que no sea solo el sistema en el cual se postuló su existencia y en tal caso queda la pregunta sobre el nivel de consistencia del sistema de axiomas. Se es consciente de que las matemáticas eran mucho más abstractas y formales de lo que se creía hasta entonces. Más abstractas, porque las aserciones matemáticas no siempre son construidas sobre conjuntos de objetos o propiedades de objetos determinados de manera intrínseca y clara; y más formales, porque la validez de las demostraciones matemáticas reposa sobre la estructura de las aserciones que ellas contienen y no sobre la naturaleza particular de lo que se está demostrando, lo que va en detrimento también de la consistencia del sistema matemático.

La búsqueda de soluciones a estas problemáticas presentadas, va a ocupar varios de los desarrollos de las matemáticas y la lógica a principios de siglo xx, se diferencian tres corrientes: los logicistas, los intuicionistas y los formalistas.

5. Corrientes reformulacionistas

Logicistas: sus principales representantes son Frege[10] y Russell[11] que procuran fundamentar las matemáticas utilizando únicamente recursos de la lógica tradicional, quitándoles cualquier contenido diferente a esta.

Frege con su obra *Begriffsschrift*: "libera la lógica de una conexión artificial con las matemáticas, mientras que al mismo tiempo prepara una interrelación más profunda entre estas dos ciencias"[12] y le da a esta ciencia su forma moderna. El objetivo era llegar a una cadena de razonamientos donde no falte ni un solo anillo: sería una cadena sin lagunas (*lückenlos*). Pero tal ideal exige el empleo de un simbolismo, la matemática lo tiene pero no el razonamiento matemático, que se expresa en parte con el lenguaje ordinario, de manera ambigua. Así, el primer objetivo de esta

[10] Gottlob Frege (1848, Wismar, Alemania – 1925, Bad Kleinen, Alemania).

[11] Bertrand Russell (1872, Trellech, Reino Unido – 1970, Penrhyndeudraeth, Reino Unido).

[12] J. van Heijenoort, *From Frege to Godel, a source book in mathematical logic 1879-1931,* Cambridge, Mass, Harvard University Press, 1967, pág. vi. Citado por R. Blanche y J. Dubucs, *La logique et son histoire*, Paris, Armand Colin, 1996, pág. 309.

Ideografía es presentar un criterio más seguro de la validez de una cadena de inferencias que permite remontar hasta la fuente de todo aquello que estaba implícito. Es la *ideo* que impulsa la *grafía*, para poder escribir una lengua hay que construirla primero. Con esta obra, la lógica se libera de la esclavitud del análisis de las lenguas naturales.

FREGE expone la primera presentación axiomática y científica del conjunto de las nociones de base de la lógica, integradas a un sistema perfectamente coherente:

• Concibe la introducción del modo de pensamiento matemático en la construcción de la lógica, de manera que establece una prioridad de la proposición sobre el concepto y, por tanto, del análisis de la proposición, que puede, a la vez, descomponerla en sus elementos; la distinción fundamental no es la de *sujeto* y *predicado*, sino aquella que distingue una *función* y un *argumento*; fundamenta en particular la noción de *función*[13], que para el análisis de la proposición, es un paso decisivo en la renovación moderna de la lógica. El análisis presenta así, las siguientes ventajas:

• Permite el análisis clásico, pero en un nivel subordinado: un concepto es una función a un argumento, cuyo valor es siempre valor de verdad; la extensión de un concepto a dos valores, es el recorrido de los valores que transforman esta función en una proposición verdadera.

• Base de la teoría moderna de la cuantificación: si en lógica clásica la cantidad de una proposición general se determina por el “sujeto” (concepto), es decir una función, la cuantifi-

[13] La palabra *función* fue introducida en el lenguaje matemático por LEIBNIZ, y la notación $f(x)$ por CLIRAUT y EULER. FREGE libera la noción de su uso expresamente matemático, y admite argumentos diferentes a los numéricos.

cación moderna se hace sobre el argumento indeterminado: no sobre *hombre* en *todo hombre es mortal*, sino sobre x que el que sea es mortal si es hombre. De esta manera, la cuantificación podrá ser múltiple, en una sola proposición: varias funciones con argumentos distintos: $(x) f(x) \supset (\exists y) g(y, x)$[14].

• Organiza en un sistema deductivo y presenta las relaciones entre las *leyes lógicas* o *juicios del pensamiento puro* que entran en juego en la deducción matemática, así que el contenido de estas se encuentra condensado en unas pocas.

RUSSELL, de su lado, para evitar las antinomias, como "la noción de clase que se contiene a sí misma", va a proponer la "jerarquía de tipos" (RUSSELL, 1908, 1910); es decir, la *jerarquía de predicados*, en la cual la restricción impuesta por esta teoría, es que la clase (o predicado) a la cual pertenece el individuo (último o no) debe ser de tipo inmediatamente superior a la suya, de manera que un enunciado de la forma $x \in x$ *está prohibido.* Teoría propiamente lógica, porque RUSSELL muestra que todas las antinomias conocidas y no solo las de la teoría de conjuntos, cuando consisten en decir por medio de una proposición algo que se refiere a ella misma, son producto de un "círculo vicioso", y entonces pueden ser resueltas por la teoría de tipos.

RUSSELL distingue los enunciados sin sentido y los enunciados falsos, distinción importante para un lógico[15], donde la forma "sin sentido" resulta de la forma misma, es un "error

[14] La teoría clásica se contenta de ordenar esta proposición en los universales sin ir más lejos en el análisis.

[15] Los enunciados de la metafísica, son *sin sentido*, es una de las tesis esenciales del *empirismo lógico*.

de sintaxis", no del contenido mismo del enunciado, que en este caso sería *verdadero* o *falso.*

Los *Principia mathematica* (Russell-Whitehead, 1910-1913), es una verdadera *suma* de *principios matemáticos*; presenta los trabajos de formulación de axiomas de analistas y geómetras, lo mismo que la lógica simbólica como instrumento de razonamiento de las matemáticas. Reconoce la diferencia entre una regla de inferencia y una ley de cálculo. Por otro lado, presenta los desarrollos del *cálculo de relaciones*, en donde una relación es la clase de diadas, triadas, etc., que sostienen entre ellas una relación; por ejemplo, la clase de la relación *capital de* es la clase de la diada (x, y) tal que x sea la capital de y (Bogotá, Colombia; Lima, Perú; etc.). Así, ciertas proposiciones que se tenían como axiomas son demostradas o declaradas *superfluas,* de manera que el campo de las matemáticas se extiende y se precisa por adición de sujetos nuevos y reevaluación de otros.

Anotaciones. Los diversos desarrollos de los *logicistas* están al origen de la lógica moderna; es uno de los instrumentos de escritura simbólica que se utiliza hasta ahora y es el lenguaje común de los lógicos. Sin embargo, es necesario disociar este instrumento lógico de la reducción logicista a la cual el instrumento estaba destinado y por la cual, esta escuela tuvo muchas críticas.

Intuicionistas: entre los cuales figuran Brouwer, Weyl, Poincaré y Kronecker. Argumentan que los principios de la matemática clásica no eran confiables y que la lógica ya no se tiene como verdad absoluta. Su posición es *constructivista*; así, por ejemplo, para la teoría de números, se considera una lista no finita de números naturales:

$$0, 1, 2, 3, 4, 5,\ldots, n, n + 1$$

De manera que cada número, excepto el 0, es sucesor de un número único. Se puede definir recursivamente las operaciones de la adición, producto,..., y, en consecuencia, no es necesario referirse a la existencia de un *conjunto* N que contenga una totalidad infinita de números naturales, ni un conjunto $C(N)$, ni 2^T como el conjunto de todos los conjuntos, etc.

Así, a partir de la *construcción* de los números naturales, pueden construirse los enteros, y luego los racionales a partir de enteros,..., sin hacer mención a la existencia de conjuntos infinitos, ni tampoco a la noción de *totalidad*. Por medio de la construcción se abren las posibilidades al *infinito potencial*, no son dominios cerrados de cosas que existen en sí mismas; por el contrario, las matemáticas clásicas con la creencia del *absoluto* transcienden las posibilidades de realización y ostentan verdades no fundamentadas. Igualmente, las demostraciones por contradicción o reducción al absurdo no son aceptadas en el *intuicionismo* porque para demostrar, por ejemplo, la existencia del número n tal que tenga la propiedad p, el intuicionista construye o encuentra un método que permita obtener el número n con tal propiedad. Al contrario, el matemático clásico concluye, por contradicción, que existe un n con la propiedad p, sin dar cuenta clara del procedimiento utilizado para encontrarlo.

En cambio, de la contradicción, los intuicionistas utilizan el principio de inducción. BROUWER (1908)[16] crítica el planteamiento de que las leyes lógicas poseen una validez independiente del sujeto al cual se aplican, especialmente la ley del *tercer excluido*: $P \vee \neg P$.

[16] “Sur le caractère incertain des principes logiques”.

Así, por ejemplo, sea la proposición $P(x)$, x pertenece al conjunto C, aplicado a $\exists x\, P(x)\ V \neg\exists x\, P(x)$, esta ley afirma que o bien existe en C un x tal que $P(x)$ o bien no existe en C ningún x tal que $P(x)$. En el caso que C sea un conjunto finito $\exists x\, P(x)\ V \neg\exists x\, P(x)$ es verdadera porque nos podemos dar cuenta que $\exists x\, P(x)$ o bien que $\neg\exists x\, P(x)$, mirando si cada valor x de C, $P(x)$ se realiza, como C es finito debe ser posible, en principio, de llevar el procedimiento hasta el final. Pero si C es un conjunto infinito enumerable, como el conjunto de números naturales, el procedimiento no puede humanamente llevarse hasta el final y, por tanto, no se puede creer que $\exists x\, P(x)\ V \neg\exists x\, P(x)$ es verdadero siempre.

Anotaciones. Para los intuicionistas la lógica clásica ha dejado de lado los conjuntos finitos y, por tanto, no cuenta con los límites originales. Igualmente, esta lógica considera a los conjuntos infinitos, como superiores y anteriores a las matemáticas y termina por aplicarlos, sin razón valedera.

Formalistas: entre los cuales se cuentan: ZERMELO, FRAENKEL, VON NEUMANN, SKOLEM, BERNAYS, PEANO y su principal representante DAVID HILBERT, en las décadas de 1920-1930. Pretenden rescatar la matemática clásica con base en la formalización de la matemática. Para ellos, las matemáticas van más allá de aquello que es pensable claramente y justificable por la *intuición*, y van a proponer un programa para salvar las matemáticas clásicas, que deben desarrollarse por una *teoría axiomática formal y consistente*. Antes de su programa, una demostración por consistencia para una teoría, consistía en presentar un modelo o interpretación, en donde todos los axiomas de la teoría debían ser verificados, en términos de la otra teoría tomada como modelo (la geometría de LOBATCHEVSKI es consistente si la geometría euclidiana lo es); es decir, que el método del *modelo* muestra que la teoría es consistente si la otra teoría lo es. Pero ¿cómo establecer la consistencia del análisis, por ejemplo? Seguramente ni reenviándolo al modelo geométrico ni al universo físico; se trata más de una idea que se obtiene por extrapolación. Entonces, demostrar la consistencia de los sistemas matemáticos exige otro método diferente al del *modelo*, que reside en el hecho de que no haya contradicción o paradoja en dos teoremas en el que uno sea la negación del otro.

La *axiomática formal* asegura una completa libertad para escoger los objetos que designan los términos primitivos con la sola condición que verifiquen los axiomas, porque la interpretación de los términos primitivos, en lugar de ser especificada con anterioridad, se deja sin especificar. En este sentido, los términos de la axiomática deben ser tratados como desprovistos de contenido cuando se trata de hacer deducciones lógicas a partir de los axiomas, para evitar los contrasentidos, porque si los términos tuvieran un contenido equivaldría a decir que los teoremas no dependen solo de las propiedades de los términos primitivos, sino también de otras propiedades que se introducirían por intermedio del contenido de esos términos.

Igualmente, para mostrar que la contradicción es imposible, propone la *teoría de la demostración* o *metamatemática*, cuyo objetivo es la realización de demostraciones a partir de esa *axiomática formal*; aquí se utilizan solo métodos *finitistas*, que son intuitivamente más seguros.

Así mismo, los formalistas al admitir que las matemáticas clásicas desbordan el campo de la evidencia intuitiva, explican cómo sus partes no intuitivas pueden tener un valor, haciendo una distinción entre los *enunciados reales* (sentido intuitivo) y los *enunciados ideales* (presuponen el infinito actual). A un sistema construido con enunciados reales se le pueden agregar enunciados ideales, con fines teóricos tales como la simplificación de teoremas o su unificación bajo un punto de vista único; por ejemplo, en la teoría analítica de números cuyos teoremas sobre los enteros se demuestran por medio de la teoría de números reales o complejos.

6. Conclusiones

Los desarrollos en lógica de las tres corrientes fundamentan la *lógica contemporánea* o lógica matemática, en donde las tres vías reformulacionistas aceptan y aportan los presupuestos de esta lógica. Así, la matemática no va a fundarse exclusivamente sobre la lógica, como pretendía Russell y Frege, pero acepta presupuestos como los dados en la *Begriffsschrift* o la *Teoría de tipos de Russell.* Igualmente, las controversias entre Brouwer y Hilbert, llega a su fin con el reconocimiento por los intuicionistas, del hecho de que el programa de Hilbert sería aceptable si se abstenían de ver, en una demostración de consistencia, ciertas razones que pretendían dar un sentido real a algunas partes de las matemáticas consideradas por los intuicionistas como desprovistos de fundamento intuitivo. Para lo cual, Hilbert presenta la distinción de *enunciados reales e ideales*, que hemos planteado. Nos encontramos con una lógica cuya reorientación está basada en la *efectividad*, *formalización* y *operatividad*. Así, los formalistas e intuicionistas van a pertenecer a la misma familia, en el sentido que los dos van a atribuir un lugar prioritario a los razonamientos constructivos.

Capítulo II

FORMALIZACIÓN

1. Introducción

Una gran parte de los esfuerzos de los lógicos del siglo xx se ha orientado a la formalización rigurosa del pensamiento matemático. En este capítulo nos proponemos mostrar cómo se formaliza un dominio de las matemáticas, que en este caso sería la *lógica elemental de proposiciones*. Se muestra de esta forma, los elementos, la estructuración, los diferentes pasos de esta formalización, que conlleva la constructividad y la simbolización para obtener un *sistema formal*. La simbolización requiere, entonces, la estructuración de un lenguaje formal.

2. Proceso de formalización

La formalización consiste en sustituir los enunciados de un discurso usual, la mayoría de las veces equívoco y ambiguo, suprimiendo todo factor intuitivo o subjetivo, por un discurso simbólico, donde cada configuración de signos y cada regla tienen un significado unívoco y claro. Su objetivo es establecer un *sistema* que permita pasar de una configuración de signos a otra configuración, en virtud de reglas definidas con anterioridad, sin considerar otra cosa que los signos y las propiedades de los objetos que estos signos pueden representar. Un tal sistema se llama *sistema formal*.

El proceso de formalización depende de la riqueza del pensamiento que se traduce simbólicamente; si se toma, por ejemplo, como objeto, los razonamientos empleados por las matemáticas clásicas, se tendría, en primera instancia, una lógica elemental de proposiciones. Pero si, al contrario, se considera la deducción lógica usual de la complejidad que proporciona el uso de diferentes modalidades, se constituye entonces una lógica modal[1].

De manera general, la formalización parte de los enunciados correspondientes a una teoría no formal y busca expresar de manera simbólica las propiedades estructurales de estos enunciados; por ejemplo, cuando se formaliza una parte de las matemáticas, el punto de partida es una matemática deductiva pero que ha recurrido a la *intuición,* de allí el nombre dado, por algunos matemáticos (BOURBAKI, WAERDEN,…), de *matemática ingenua*. El proceso que se realiza, en este caso, es el paso de lo intuitivo (no formalizado) a lo formal; se podría pensar en los axiomas y postulados de EUCLIDES, que no enuncian la totalidad de propiedades que se utilizan en las demostraciones y son el punto de partida para HILBERT (1899), para la constitución de los *fundamentos de la geometría*. Él presenta un tratamiento formal de postulados y axiomas de EUCLIDES, por medio de su *axiomática formal,* e incluye las hipótesis "escondidas" de EUCLIDES.

Por otro lado, se puede dar un proceso de lo formal a lo ingenuo o intuitivo; en consecuencia, si tenemos un *sistema*

[1] Es la lógica de proposiciones que ha sido estudiada con mayor asiduidad, tanto por su relativa simplicidad como por la necesidad de fundamentar la teoría de conjuntos.

formal dado, se puede estudiar las propiedades del sistema, que den sentido a las nociones puramente formales de tal sistema. El resultado de este estudio es una *teoría intuitiva* que constituye una *interpretación del sistema* y el estudio de la correspondencia entre el sistema y su interpretación es el objeto de la *semántica*. Se ve claramente que los dos procesos de formalización y de interpretación toman sentido en la relación correspondiente entre estos dos procesos.

Es entonces, el doble pasaje de lo intuitivo a lo formal (formalización) y de lo formal a lo intuitivo (interpretación), lo que hace que se produzcan estos sistemas formales de lógica contemporánea.

La formalización presupone, igualmente, la noción de *constructividad* y la de simbolización.

Anotaciones. La formalización es una noción que está enfocada a la generalización y abstracción de la teoría intuitiva, en las cuales se exponen las diversas proposiciones que soportan nuestro razonamiento, lo mismo que los sistemas de reglas, así que la forma como se desarrolla la formalización se presenta de un modo categorial: no se trata de prescribir o prohibir, sino de llegar a convenciones; se necesita una cierta libertad para la construcción de las proposiciones y de la lógica correspondiente (*principio de tolerancia de la sintaxis*). En este caso, en el desarrollo de una teoría formalizada, el parámetro lógico no posee un sentido absoluto, es relativo al conjunto de proposiciones que se van a formalizar.

A) *Constructividad*

El esfuerzo realizado por una formalización consiste en circunscribir las expresiones dadas, intuitivamente, en un campo formal; de manera que estas expresiones formalizadas puedan ser consideradas como sustraídas de cambios o cuestionamientos posteriores. La *constructividad* asegura esta delimitación y puede asimilarse a lo *enumerable*, propiedad que la separa de la decidibilidad o del procedimiento efectivo mismo.

Los elementos que intervienen en la formalización, por intermedio de la constructividad, son totalmente distinguibles los unos de los otros y del resultado de esta. En conjunto con el proceso de la simbolización (lenguaje simbólico), son expresiones claras y precisas propias para ser analizadas con objetividad; por ejemplo, un conjunto de enteros puede enumerarse de manera precisa, cuando el sistema y sus resultados se conocen claramente.

La noción de *constructividad* pertenece a lo "actual" y requiere un carácter "efectivo" (un procedimiento), pero puede enfocarse también sobre operaciones "futuras", como lo planteó HILBERT (1925) para el tratamiento de lo *infinito*. En este sentido, dos tipos de elementos pueden representarse efectivamente: los unos se relacionan con las operaciones o realizaciones *efectivas* y los otros referidos a posibilidades de ser efectuados; cualquier evocación de *futuro* (*virtual*) tiene necesariamente como soporte lo *actual*; si los dos tipos de elementos se sitúan en lo *efectivo*, la posibilidad de una operación de totalización es imposible, no se puede realizar en efecto y se queda en el campo del *concepto*.

Por este hecho, la constructividad, en general, no tiene un sentido completamente riguroso, se puede ver en la exclusión de las *totalizaciones infinitas* y no permite la introducción del infinito sino bajo su forma *potencial*; sin embargo, esta representación no es estática: la constructividad suministra la posibilidad de ir más allá de lo *actual*.

Por ejemplo, en el caso de *funciones recursivas*, la cadena de enteros puede caracterizarse por medio de la noción *sucesor*, a través de la cual se reconstituye la cadena de los enteros a partir de 0; esta cadena de enteros permite abordar los elementos que vienen después de la cadena, pero no la totalización entera de la cadena.

De manera que lo propio de la constructividad es la *apertura*: después de cualquier entero hay más enteros, ninguna enumeración puede agotar aquello que es accesible por el pensamiento constructivista; sin embargo, esta noción se mantiene en lo *enumerable*: si se determina un objeto que no entra en una cadena de enumeración dada, es posible determinar una nueva enumeración en la cual este objeto sea incluido e ir más allá de los términos efectivamente accesibles, y así sucesivamente.

Aquí es necesario distinguir un *horizonte* que, al contrario de los términos que se pueden describir efectivamente, no se da en forma actual, pero no obstante se da con cada uno de estos términos. El *horizonte* es lo que hay de presente, posibilita la percepción de aquello que puede ser traído al presente por la intuición y se vuelve un componente necesario de la estructura de la experiencia matemática.

Anotaciones. En la noción de constructividad subyacen dos posiciones sobre el *ser* y la *existencia* de un *objeto matemático* (LADRIERE, 1957, pág. 30), así:

• Posición de los *intuicionistas* (BROUWER y su escuela), *nominalistas*, *constructivistas* y *empiristas*: la existencia matemática se reduce a la *constructividad*: el ser matemático no es una realidad autónoma, no existe sino en el *acto* por el cual se engendró (Boniface, 2004)[2]; la exigencia de constructividad lleva a no tolerar cualquier consideración del *infinito actual* y no admite sino el *infinito potencial* (de aquí la concepción de continuo; o sea, que contiene partes que se dividen indefinidamente). Desde el punto de vista lógico, esta exigencia se traduce por no utilizar el *tercer excluido* (para las *colecciones infinitas*): no se puede afirmar *a priori* que la solución de un problema matemático debe ser de tipo A o $\neg A$, sino hasta que se tenga un procedimiento efectivo que permite obtener esta solución, si es el caso.

• La posición de Hilbert en su *metamatemática*, exige la constructividad, porque para Hilbert el ser matemático existe, en principio, en el pensamiento y, a la inversa de Brouwer, la existencia matemática se prueba por la *no contradicción;* por ejemplo, si se construye un sistema formal para la aritmética, se puede luego demostrar la no contradicción que se extiende progresivamente a todo el *análisis*. La constructividad se sitúa en criterios de demostración, muy restrictivos, no admite los razonamientos por inducción completa, solo la utilización de una recurrencia en el campo de lo finito.

El *sistema formal* integra estos dos aspectos: por una parte, las nociones constructivistas que admiten el infinito potencial y, por la otra, la noción hilbertiana de la demostración

[2] Boniface nos señala que el pensamiento matemático se funda sobre una intuición originaria, aquella de la división de la unidad, como fuente de la dualidad, la cual es la misma de la estructura del tiempo y es ella misma la que es la base de la noción del número entero cuya posibilidad de una repetición indefinida, da lugar al carácter indefinido de la cadena de enteros.

de no contradicción con criterios efectivos de demostración, que podría verse en el infinito actual.

B) *Simbolización*

La formalización exige el recurso a *símbolos* diferentes de aquellos del lenguaje ordinario: es prácticamente imposible satisfacer las exigencias estrictas de la formalización, si se expresan en el lenguaje usual, con su imprecisión e irregularidades; es por esto por lo que la formalización supone la *simbolización*. Esta requiere un progreso en la *abstracción*: un enunciado puede utilizar símbolos sin asegurar por esto que sean *formales*; es necesario, además, que intervenga un nivel de *abstracción* necesario en la creación de un simbolismo nuevo.

Este proceso suministra a las expresiones que se formalizan, el lenguaje necesario para la rigurosidad de la formalización. Este lenguaje posee una triple propiedad:

• Se presenta como una "exterioridad" accesible a la percepción;

• Representa los requerimientos intuitivos de la "teoría intuitiva" que se formaliza;

• Posibilita un análisis riguroso del proceso de formalización, porque su análisis no se hace sobre los contenidos intuitivos, sino sobre la representación formal.

Anotaciones. EL núcleo de la simbolización radica, entonces, en el lenguaje y en este caso se trata de un *lenguaje formalizado* (véase cap. II, sec. 4: "Lenguajes simbólicos formales").

3. Sistemas formales

El resultado del proceso de formalización, con las propiedades de constructividad y simbolización, es un *sistema formal*, que se presenta a la base de las teorías deductivas para llevar a cabo la axiomatización de estas teorías.

En este sentido, el *sistema formal* se sitúa en la evolución de la *axiomática*, en la cual se distinguen cuatro estadios (Ladriere, 1957, págs. 35-36):

• *Axiomática intuitiva*: se separan los conceptos (intuitivos), los enunciados fundamentales (evidencias) y los procedimientos de deducción que son los de la lógica natural, por ejemplo, los *Elementos de Euclides*.

• *Axiomática abstracta*: los conceptos fundamentales se precisan y califican como *generales* y se verifican los axiomas que se aplican a los objetos; no se tiene en cuenta sino ciertas propiedades que son enunciadas explícitamente.

• *Axiomática formal*: los conceptos fundamentales no desempeñan ningún papel, su sentido se fija a partir de las relaciones establecidas entre ellos por los axiomas; estos conservan todavía expresiones del lenguaje ordinario, cuyo sentido esta dado por la intuición, por ejemplo, los axiomas de Peano.

• *Sistema formal*[3]: en donde "la referencia a un sentido exterior al sistema se elimina, gracias a la utilización de un

[3] Marion (2004, págs. 103-107) relata esta evolución: Frege en su "Ideografía" de 1879 concibe varias nociones innovadoras para la historia de la lógica, así como el reemplazo de la idea *sujeto/predicado* por la de *función/argumento*, introduce la teoría de funciones, los cuantificadores y la idea de *sistema formal*. Con esta noción, se propone desarrollar un sistema de lógica formal a partir del cual los teoremas de

lenguaje simbólico rigurosamente definido; los procedimientos de deducción admitidos se presentan completamente, pero la intuición no puede ser eliminada completamente, esta no se refiere al contenido de los conceptos o expresiones, sino que se reduce a la intuición de las manipulaciones definidas sobre un sistema de signos. Basta con poder identificar un signo, distinguir dos signos diferentes, reemplazar un signo por otro según un modelo de sustitución [...]. El sistema axiomático no tiene más como objetivo que el de hacer aparecer el orden natural que existe entre los enunciados de una teoría [...]. Se pide al sistema responder a ciertas condiciones de simplicidad y de claridad que conducirán la selección de axiomas: lo que hay que obtener es la eliminación de toda ambigüedad" (LADRIERE, 1957, pág. 37).

Así, por ejemplo, el paso de una axiomática intuitiva a un sistema formal se puede ver, al considerar una teoría matemática intuitiva (geometría euclidiana), que reposa sobre un cierto número de *definiciones*, *axiomas* y *postulados* explícitamente enunciados, que no requieren demostración y son admitidos sin pruebas, en donde:

• Las definiciones: caracterizan los objetos de la teoría; se asocian a representaciones intuitivas, la mayoría de las veces (representación de la recta, del plan,...).

---

la teoría elemental de números serían inferidos; se demuestra que a todo teorema de la aritmética corresponde un teorema al interior del sistema formal, y *viceversa*; se opera de esta manera una *reducción* de la aritmética a la lógica. Esta noción siguió una evolución gracias a las críticas hechas al logicismo, entre otros, por L. WITTGENSTEIN (parág. 4.1273 del *Tractatus logicus philosophicus*), por la circularidad existente en la noción de relación *ancestral*, noción fundamental para la reducción logicista de la cadena de números naturales.

• Los axiomas y postulados: enuncian las propiedades fundamentales; se asocian a evidencias variables, según el caso ("dos cantidades iguales a una tercera son iguales entre ellas").

A partir de estos, era posible deducir las proposiciones nuevas; sin embargo, la deducción no está explícitamente definida y admite sin crítica particular todos los tipos de razonamiento estimados correctos. Además, en las demostraciones se suponía más de lo que estaba explícitamente enunciado; en el inventario de los principios y el lenguaje utilizado se hacen llamados a la intuición, en fórmulas variadas como "es evidente que…", "se ve que…". De aquí, la necesidad de evitar evidencias no controladas y renunciar a apoyarse sobre representaciones de los seres matemáticos ideales, como la recta o el círculo trazados sobre papel.

La formalización de una axiomática intuitiva y de una abstracta de una teoría intuitiva dada, se presenta así:

• A los objetos de la teoría matemática estudiada y a las relaciones que los unen corresponden los símbolos por sí mismos desprovistos de significación. Este es el esquema de la geometría dada por Hilbert (1971) en los *Grundlagen der Geometrie* o de la axiomatización de la aritmética de Peano.

• A las definiciones, axiomas y postulados corresponden enunciados donde aparte de los símbolos figuran "palabras" que expresan relaciones lógicas (y, o, si… entonces,…).

• A partir de los axiomas y por deducción formal, es decir, por un razonamiento que no tiene en cuenta el sentido que pueden tener los símbolos, se establecen nuevas relaciones. Para encontrar el sentido concreto de las relaciones, obtenidas de esa manera, es necesario hacer corresponder los objetos matemáticos a los símbolos.

Anotaciones. En principio, se puede constituir un sistema axiomático, de manera arbitraria, a condición de que los axiomas no tengan ninguna contradicción. Sin embargo, muchos de los sistemas efectivamente empleados, se construyen de manera que se puedan deducir por medio de la lógica usual todas las expresiones que traducen los teoremas de una teoría intuitiva dada y a estas expresiones solamente. Cuando estos sistemas satisfacen esta condición, se dice que ellos proporcionan una *axiomatización adecuada* de la teoría intuitiva considerada.

En consecuencia, la adecuación de una teoría axiomática formal a una teoría intuitiva dada, permite verificar que un número finito de teoremas dado se traduce en expresiones derivables del sistema de axiomas considerado. Se puede verificar que ciertos esquemas de deducción frecuentemente empleados en la teoría intuitiva, pueden ser sustraídos de la teoría axiomática. Sin embargo, establecer una correspondencia estricta de término a término entre teoría axiomática formal y teoría intuitiva no se da, en la mayoría de los casos, se tiene una adecuación global, la teoría axiomática formal no pretende dar una imagen idéntica de la teoría intuitiva, ella sustituye y amplía su sentido.

Pero antes de interrogarse sobre la fidelidad de una axiomatización, es necesario preguntarse sobre la consistencia de esta adaptación. Si por principio, no se hace llamado al sentido o contenido que pueden recibir los símbolos, ni a la evidencia de los axiomas, se encuentra que ciertos enunciados son tomados por axiomas, no porque la proposición intuitiva que les corresponde presenta una evidencia particular, pero porque ellos permiten deducir fácilmente los enunciados que corresponden a los teoremas de la teoría intuitiva.

Así mismo, la consistencia de los sistemas axiomáticos no puede ser establecida, sino por las demostraciones de no contradicción. Sin duda, se puede contentar con establecer la no contradicción relativa de una teoría (T_2), mostrando que puede ser traducida en una teoría segura (T_1), de manera que una contradicción en (T_2) se traduciría por una contradicción en (T_1). Se concluye que si (T_1) es no contradictoria, (T_2) lo es también. Así, HILBERT demostró la no contradicción de su geometría axiomática, traduciendo los axiomas en la teoría de números reales. Es, entonces, necesario establecer la no contradicción de ciertas teorías fundamentales como la teoría de números reales o la aritmética elemental. Se constata, por tanto, que no es posible considerar que todo lo que corresponde a la deducción lógica se da de hecho, sino que debe ser axiomatizada con la misma importancia que la parte propiamente matemática de una teoría. Esta necesidad aparece con gran evidencia cuando se constató que las teorías matemáticas particulares, el álgebra, por ejemplo, podrían ser consideradas como capítulos particulares de la teoría general de conjuntos, al realizar la axiomatización.

Anotaciones. El examen de las paradojas presentadas anteriormente (véase cap. I: "Génesis de los lenguajes formales"), muestra que las dificultades encontradas eran de orden lógico al menos más que matemático y mientras no se definieran nociones como *propiedad* o incluso *definición matemática*, se continuaba en las contradicciones. Por este hecho, los sistemas lógicos que permiten presentar una axiomatización total de la matemática (teoría de tipos, cálculo de predicados de primer orden o de orden superior) tomaron una importancia considerable y fueron el objeto de estudios importantes, durante la primera mitad del siglo XX.

A) *Construcción de un sistema formal*

Se ha presentado que el *sistema formal*, es la última instancia de formalización de una axiomática intuitiva, abstracta y formal. En este sentido, es preciso mostrar los pasos necesarios para la construcción de un *sistema formal* (*S*) (MARTIN, 1968, págs. 13-16), así:

1. Definición de un inventario completo o conjunto de signos, estructurados por categorías, a las cuales cada signo pertenece a una sola categoría. Este conjunto puede llamarse *alfabeto* o *vocabulario* del sistema. Se puede designar por $\sum$ y se considera como enumerable, así si $\sum$ es finito se puede escribir efectivamente la lista completa de los signos que constituyen el vocabulario.

2. Combinación de signos tomados del vocabulario que forman cadenas finitas de signos que se pueden llamar *palabras*. Un mismo signo puede figurar varias veces en una palabra. Las palabras son entonces arreglos (con repetición eventual) de signos prestados del vocabulario. Se designa por Γ el conjunto de palabras. El conjunto de palabras es enumerable como el conjunto de signos del vocabulario:

$$p_1, p_2, \ldots, p_n$$

3. Entre las cadenas finitas producidas con los elementos de $\sum$, se distinguen ciertas que se llaman *expresiones bien formadas* que en su abreviación serán: *F* y el conjunto de *F* se designa por Φ. Este conjunto se define mediante *reglas de formación;* y se puede decir que ellas son las *reglas de*

la gramática del sistema. La definición de F tiene tres tipos de condiciones:

a) Condición que fija las F iniciales y define un subconjunto Φ_0 de Γ.

b) Condición que permite obtener a partir de una F nuevas F; ellas son de tipo: "Si F_1, F_2,…, F_n, son F, entonces la expresión F_p que se obtiene a partir de F_1, F_2,…, F_n, es una F". Esta condición se llama *regla de formación a* n *antecedentes, n* puede variar de una regla a otra.

c) Condición de cierre: cualquier palabra no es F si no se obtiene por la aplicación de las dos primeras condiciones.

Se ve que el conjunto de las F de Φ es el más pequeño subconjunto del conjunto Γ de palabras que contienen las F iniciales Φ_0 y es cerrado con respecto a la aplicación de reglas presentadas en la segunda condición.

4. A) Entre las F de S se distinguen unas que se llaman *F demostrables en S* o, igualmente, *tesis* de S. El conjunto T de *tesis* de S se define como el conjunto de las F resultado de la aplicación de reglas y la definición de *tesis* hace intervenir tres tipos de condiciones:

a) Condición que fija las *tesis* iniciales, es decir, que fijan un subconjunto T_0 de Φ. Las tesis iniciales de S se llaman *axiomas de S*.

b) Condición que permite obtener a partir de una T nuevas T; ellas son de tipo: "Si T_1, T_2,…, T_n, son *tesis*, entonces la *tesis* T_p que se obtiene a partir de T_1, T_2,…, T_n, es una *tesis*", se llaman también *teoremas*. Esta condición se realiza por medio de las *reglas de deducción a* n *antecedentes, n* puede variar de una regla a otra.

c) Condición de cierre: una F es una *tesis* si se obtiene por la aplicación de las condiciones a) y b).

Se observa que el conjunto T de *tesis* es el más pequeño subconjunto del conjunto Φ de las F que contiene T_0 y es cerrado con respecto a la aplicación de reglas.

B) Conjunto de demostraciones en S. Una cadena de F: $F_1, F_2,\ldots, F_n$ es una *demostración en S* si y solo si una F F_i $(1 \leq i \leq n)$; o bien, F_i es una *tesis inicial* (*axioma*), o bien, F_i se obtiene aplicando una regla de deducción con k antecedentes que preceden a F_i en la cadena considerada y es, entonces, un *teorema*. Se observa que toda F que pertenece a la demostración de S es una *tesis* de S, se llama entonces *demostrada* por medio de las F que las preceden.

Las reglas son semejantes a lo que constituiría la *sintaxis* de un *lenguaje formal*, pero donde se ignora su significado: las reglas de formación indican cuáles son las expresiones dotadas de sentido, mientras que las reglas de deducción permiten encadenarlos.

B) *Sistema formal particular*

De manera ilustrativa de un *sistema formal*, vamos a describir un sistema S para formalizar la *lógica proposicional*, así:

• *Vocabulario.* Son los *signos formales* que estructuralmente desempeñan el papel de vocabulario o alfabeto del *lenguaje formal*, organizados por categoría:

Variables

— a, b, c,… se necesita un gran número enumerable de variables, potencialmente para su uso.

— *p*, *q*, *r*, *s*,… son nominadas *variables proposicionales*, cuando las variables (*a*, *b*, *c*,…) se reemplazan por proposiciones.

Constantes. Son *conectores proposicionales* y *signos de puntuación*.

Conectores:

— $\neg$: no

— V : o

— $\supset$: si… entonces

— $\wedge$: y

Signos de puntuación:

— ( ): paréntesis de izquierda y derecha.

• *Reglas de formación.* Según su definición, estas se establecen de manera que ciertas combinaciones de signos, que tendrán normalmente la forma de *proposición*, son las *expresiones bien formadas* o *fórmulas,* así:

— "F" representa *una expresión bien formada*, "$\neg F$" también lo es.

– "F_1" y "F_2", cada una, son *expresiones bien formadas*, "$F_1 V F_2$", "$F_1 \supset F_2$", "$F_1 \wedge F_2$" también lo son.

• *Reglas de transformación*

— Reglas de sustitución, para variables proposicionales, se estipula que de una *expresión bien formada* que contiene variables proposicionales, se puede derivar otra *expresión bien formada* sustituyendo las variables por las expresiones, sabiendo que si se efectúa esta sustitución se debe hacer por cada ocurrencia de esta variable, así:

— "$p \supset p$", al sustituir la variable "p" por "q" se obtiene "$q \supset q$";

— "$p \vee q$", se puede obtener: "$(p \vee q) \supset (p \vee q)$".

— Regla de *modus ponens* o de *separación*: se estipula que de dos *expresiones bien formadas* F_1 y $F_1 \supset F_2$, es posible deducir la expresión F_2. Por ejemplo:

las dos expresiones bien formadas: "$p \vee \neg p$" y "$(p \vee \neg p) \supset (p \supset p)$", se puede derivar "$(p \supset p)$".

• *Axiomas.* Los axiomas o tesis iniciales del cálculo proposicional son expresiones bien formadas, así:

— "$(p \vee p) \supset p$"; en lenguaje corriente: "*si p o p, entonces p*".

— "$p \supset (p \vee q)$"; en lenguaje corriente: "*si p, entonces o p o q*".

— "$(p \vee q) \supset (q \vee p)$"; en lenguaje corriente: "*si o p o q, entonces o q o p*".

— "$(p \supset q) \supset ((r \vee p) \supset (r \vee q))$"; en lenguaje corriente: "*si (si p, entonces q), entonces ((si o r o p), entonces (o r o q))*".

• *Teoremas.* Los *teoremas* o *tesis demostradas*, suponen, según la definición dada, que son deducidos a partir de las tesis iniciales o axiomas, y que por medio de las reglas de transformación se establece un número indefinido de teoremas; por ejemplo:

— "$((p \supset q) \supset ((r \supset s) \supset t)) \supset ((u \supset ((r \supset s) \supset t)) \supset ((p \supset u) \supset (s \supset t)))$".

Anotaciones. En síntesis, un sistema formal se caracteriza por su vocabulario, las *F* iniciales, las reglas de formación, las tesis iniciales (axiomas), los teoremas y las reglas de deducción o de transformación.

De acuerdo con estos sistemas, se distinguen aquellos cuyas reglas de formación y reglas de deducción o transformación permiten decidir por medio de un método fijado con anterioridad y por medio de un número finito de etapas, si una palabra es una *F* y si una cadena de *F* es una demostración. Estos sistemas se llaman *efectivamente definidos* o *efectivos.*

El resultado de una demostración en estos *sistemas efectivos* se reduce a la escritura de expresiones exclusivamente compuestas por signos. Aquí se produce justamente lo mismo que cuando se realiza un cálculo en matemáticas; la diferencia radica en la naturaleza de reglas que se utilizan[4]. Se puede ir más lejos (desarrollo de los lógicos de los años treinta del siglo XX: ALAN TURING, EMIL LEON POST,…) y calificar el cálculo de mecánico, cuando se trata de un sistema efectivamente definido, la construcción de *F* y la derivación de los teoremas que podría asegurarse por una máquina. Pero de lo que se trata aquí como *mecánico* es solo el cálculo como tal.

4. LENGUAJES SIMBÓLICOS FORMALES

Un sistema formal es, en todo rigor, una cadena de signos construidos y encadenados conforme a reglas, sin tener en cuenta la interpretación que pudieran recibir estos signos.

La formalización completa que se ha visto en un *sistema formal*, se ha obtenido combinando la *simbolización*, que se ha empleado en matemáticas modernas, y el tratamiento *simbólico* de los trabajos en lógica contemporánea de BOOLE, PEIRCE, FREGE, WHITEHEAD, RUSSELL… Gracias a estos dos aspectos, se puede constituir un *lenguaje formal* enteramente *simbólico* y *artificial* con respecto al lenguaje natural u ordinario.

[4] De ahí el nombre que se da a diversos sistemas formales: *cálculo proposicional, cálculo de predicados.*

En su obra *Begriffsschrift* (Frege, 1879, 1999), presenta la primera empresa de un *lenguaje formalizado* que expresa una representación lógica, con la exigencia de explicitar lógicamente las relaciones de deducción y precisar los términos que tienen como objeto las definiciones precisas; así mismo, su dinámica determinada por reglas y operaciones explícitas, en donde las proposiciones son compuestas y derivadas.

Por su parte, Wittgenstein, en el *Tractatus*, va a enmarcar este tipo de lenguajes, en su búsqueda de los *límites del lenguaje*, delimitando la diferencia entre lo "decible" (lógico) y lo "indecible" (lo que se muestra), que determina, a su vez, las márgenes de las expresiones con *sentido* y las expresiones *sin sentido*.

De esta manera, el *lenguaje formalizado* se presenta bajo diferentes connotaciones en el contexto de la lógica matemática, cuya terminología no está completamente fijada y muestra distintos puntos de vista (Rivenc, 1989, págs. 36-37)[5]:

Ciertos autores describen como *lenguaje*, lo que Church (1956) llama, *sistema logístico* o *cálculo no interpretado* (según Carnap, 1934); por ejemplo, el sistema formado por los símbolos primitivos dados y las reglas de formación y de transformación, sin ninguna consideración semántica.

El Carnap de 1939 amplía la noción de lenguaje a elementos en vías de la realización de comunicación, incluyendo la interpretación o la semántica.

El lenguaje como un conjunto de símbolos con las reglas de formación de las fórmulas y sus interpretaciones; los axio-

[5] François Rivenc nos presenta estas connotaciones. *Introduction à la logique*, Paris, Payot, 1989, págs. 36-37.

mas y reglas de inferencia serían consideradas ulteriormente como parte de la "definición de lógica" para este lenguaje.

El lenguaje (formalizado) es el que exhibe, manifiesta o reproduce la estructura o la *forma lógica*, lo que no se puede realizar con un lenguaje ordinario. La formalización puede regularse por un "lenguaje lógicamente perfecto", que reflejaría "como un espejo" la estructura lógica del mundo (punto de vista de RUSSELL[6] y FREGE).

QUINE (1977, cap. V)[7] pretende la simplificación de la teoría lógica con el fin de generalizar los artífices de la notación de la lógica moderna y poder extenderla a los enunciados particulares del lenguaje ordinario.

A) *Ejes principales del lenguaje formalizado*

A la pregunta sobre las condiciones de posibilidad y validez de este lenguaje, se pueden reconocer dos ejes principales del lenguaje de diversa naturaleza, presentados dentro del *Tractatus* de WITTGENSTEIN (HOTTOIS, 2002, págs. 138-139), así:

[6] RUSSELL en *The philosophy of logical atomism*, 1918, dice: "Un lenguaje lógicamente perfecto [...] sería completamente analítico y revelaría como un espejo la estructura lógica de los hechos afirmados o negados. El lenguaje construido en los *Principia mathematica* es concebido como siendo un lenguaje de este tipo. Es constituido de una sintaxis y no contiene vocabulario; [pero] está concebido para que al agregarle un vocabulario se tendría un lenguaje lógicamente perfecto". Citado por FRANÇOIS RIVENC, *Introduction à la logique*, Paris, Payot, 1989, pág. 37.

[7] WILLARD VAN ORMAN QUINE, *Word & Object*, MIT Press, 1960; trad. francesa: *Le mot et la chose*, par Dopp y Gochet; Paris, Flammarion, 1977, cap. V.

a) La naturaleza proposicional y analítica, correspondiente a la sintaxis del lenguaje.

b) La naturaleza representacional, correspondiente a la semántica del lenguaje.

a) N*aturaleza proposicional y analítica del lenguaje.* La *analicidad* del lenguaje corresponde a la posibilidad que tiene este de descomponerse en elementos y en la existencia de *elementos últimos y estables* que le dan al lenguaje su condición de existencia. Así mismo, el elemento constitutivo del lenguaje es la *proposición*, la cual no tiene *sentido* sino cuando es totalmente analizable: *comprender* una proposición es, en alguna medida, efectuar su análisis. Este corresponde a presentar una explicación radical (sin ninguna indeterminación), cuyo contenido de la proposición debe ser "presentada delante de nuestros ojos".

Se puede distinguir varios tipos de *proposiciones*: la proposición *elemental* o *atómica* ("*p*"), la proposición *compuesta* ("*p v q*") y dos tipos de proposiciones *especiales:* las *tautologías* y las *contradicciones*.

Una *tautología* es una proposición del cálculo de proposiciones cuyo valor es *verdadero* siempre, en cualquier tipo de interpretación. Su verdad no depende de lo que pasa en el *mundo* inscrito porque es verdadera en cualquiera de los *mundos*. Una *contradicción* o *antilogía*, al contrario, su valor es *falso* siempre (véase cap. IV: "Semántica").

Así, por ejemplo, una proposición como "*p v ⌐p*" es una tautología, que corresponde a la definición misma del conector "*v*". Se dice que las tautologías son verdaderas en virtud de su *forma lógica* y no por la interpretación sobre el mundo. Las tautologías expresan principios de la lógica[8]; por ejemplo, algunas de ellas:

$p \equiv \neg p$: principio de identidad proposicional

$p \vee \neg p$: principio del tercer excluido

$\neg(p \wedge \neg p)$: principio de no contradicción

$(p \vee q) \equiv (q \vee p)$: principio de conmutatividad de "$\vee$"

$(p \wedge q) \equiv (q \wedge p)$: principio de conmutatividad de "$\wedge$"

Así mismo, Wittgenstein distingue dos tipos de análisis:

• A partir de signos simples, que son los *nombres* (*análisis nominal*), la proposición presenta así un sentido perfectamente determinado. La proposición elemental o atómica es "verdadera" o "falsa" y no comporta partes "verdaderas" o "falsas" por sí solas.

• A partir de la noción de "verifuncionalidad" que une las proposiciones elementales.

El primer tipo responde a la noción esencial de articulación. Si los elementos constitutivos del lenguaje son proposiciones (eje proposicional), significa que las palabras aisladas, sin articulación, no son lenguaje por sí solas. La proposición analizada es la proposición expresada de tal manera, que la articulación del sentido aparezca en su estructura.

Por ejemplo, para dos proposiciones elementales: "*p*" ("está nevando") y "*q*" ("hace sol"), las dos proposiciones

[8] Esta definición implica una modificación del estatuto epistemológico tradicionalmente acordado a los enunciados que expresan verdades lógicas, es decir, a las leyes lógicas. Estas fueron consideradas como "verdades del pensamiento". La interpretación contemporánea, presenta las leyes como tautologías; es decir, a los enunciados que deben su verdad únicamente al significado del vocabulario lógico que ellos contienen.

verdaderas, la proposición compuesta "$p \wedge q$" ("está nevando y hace sol"), está articulada y es verdadera, según el conector "$\wedge$".

El segundo tipo responde a la función de verdad[9] y la proposición se expresa, por ejemplo, según una *Tabla de Verdad*[10] de proposiciones elementales. De esta manera, el lenguaje puede ser descompuesto en sus elementos, señalando que existen elementos últimos y estables y la proposición tiene sentido si es completamente analizable y comprenderla es efectuar este análisis integral, excluyendo toda ambigüedad.

Por ejemplo, la Tabla de Verdad de la proposición compuesta "$p \wedge q$" corresponde a:

P	Q	$p \wedge q$
V	V	V
V	F	F
F	V	F
F	F	F

Aquí, "$p \wedge q$" es verdadera si y solo si "p" y "q" son verdaderas y falso en todos los otros casos.

[9] La "función de verdad" se identifica a toda proposición compuesta de la "lógica de proposiciones", cuyos valores varían a partir de los valores de los enunciados simples que la componen. El cálculo de estas variaciones se puede hacer con la ayuda de una *Tabla de Verdad*.

[10] *Tabla de Verdad* entendida como un procedimiento de decisión, permitiendo de una manera semimecánica, saber en qué caso una expresión es verdadera y, en particular, de decidir si la expresión es verdadera para todas las sustituciones de valores de sus variables. Para tal uso, no se necesita ninguna utilización de la imaginación, como es el caso de la demostración.

La analicidad exige un procedimiento legítimo para explicitar lo compuesto y la elementaridad, de manera que se presenta una relación biunívoca entre los nombres y objetos representados: un nombre por un objeto y un objeto por un nombre. Lo "compuesto" posee dos propiedades: la función de verdad de las proposiciones elementales y la propiedad de "generalidad"[11] que depende de la manera como las variables se especifican; esta última se asegura por las proposiciones compuestas. La "elementaridad" asegura la finitud y la univocidad del análisis con las proposiciones elementales e igualmente define y estabiliza el sentido del lenguaje.

Las constantes lógicas

Hay una diferencia de estatuto entre las expresiones simples y las compuestas, porque estas se componen de variables y de constantes lógicas[12]. WITTGENSTEIN distingue dos tipos de signos primitivos o expresiones primitivas: por un lado, los nombres (que se relacionan a los objetos) y las proposiciones ordinarias (que se relacionan a los hechos extralingüísticos); por el otro, las constantes que permiten producir las expresiones compuestas, pero que estas, al contrario, no tienen objeto relacionado (ningún objeto o hecho extralingüístico). Las constantes pueden ser consideradas como signos puramente operativos, que permiten la generación de proposiciones compuestas pero que no tienen sustancia propia.

La constante es el instrumento de articulación de las proposiciones, razón por la cual la proposición puede es-

[11] La proposición "(para todo x) φx".

[12] LUDWIG WITTGENSTEIN, *op. cit.*, parg. 4.0312: "… lo que se llama las constantes lógicas (las expresiones «y», «o» «si… entonces», «no», «todo», «al menos uno») no representan nada y no tienen representación lógica en los hechos".

tructurarse como función de verdad[13]. Así, la constante y su repetición ordenada son la fuente de la articulación o de la estructuración de la proposición, estando de esta manera al origen de la forma proposicional que es la esencia del lenguaje formal.

"Esta esencia no es formal por carencia o defecto: es «esencialmente formal» porque la constante no representa nada. Es decir, la esencia del logos, lo que constituye el lenguaje como tal es la pura operatividad. Las implicaciones filosóficas de esta conclusión son enormes. Una implicación mayor es que la esencia del lenguaje se encuentra asimilada a una maquinaria pura, que es radicalmente tecnológica… Otra consecuencia de la naturaleza formal-operatoria de la esencia del lenguaje es la imposibilidad de describir esta esencia"[14].

Por consiguiente, "tener sentido" y "ser analítico" son equivalentes y en estos casos se distinguen el "sentido y la referencia (o denotación)" (según WITTGENSTEIN):

Sentido y referencia (*denotación*)[15] (véase cap. IV, sec. 2: "Nociones generales de la semántica")

[13] GILBERT HOTTOIS, *Penser la logique: une introduction technique et théorique à la philosophie de la logique et langage*, 2ème ed., Bruxelles, De Boeck Université, 2002, pág. 123. HOTTOIS da el ejemplo siguiente: "el operador «!» confiere a un enunciado simple su identidad, y su sentido, así: «*p*» no es nada, «!*p*» determina la representación afirmativa de un hecho simple susceptible de ser verdadero o falso. De la misma manera, «!*p*» puede ser formulado con la ayuda del operador «↓», lo que demuestra la estructura compuesta de la simple afirmación «$!p \equiv (p{\downarrow}p)\,{\downarrow}(p{\downarrow}p)$»".

[14] *Ibidem*, pág. 124.

[15] Esta problemática del "sentido y denotación" (denotación entendida también como referencia) se ha presentado desde los desarrollos

El *sentido* de una proposición esta dado por su estructura lógica, que presenta las condiciones de posibilidad de la proposición, de ser *verdadera* o *falsa* y *denota* o *reenvía* a los objetos extralingüísticos que son las referencias de los nombres, sin los cuales el lenguaje se quedaría sin referente, llevaría al infinito o a un círculo vicioso. Los objetos fijan la semántica del lenguaje. Es así como el sentido lleva a la *referencia* o *denotación* del objeto a través de un cálculo. El sentido y la referencia o denotación son contrarios en su orientación: el sentido es intralingüístico (está dentro del lenguaje lógico) y la referencia o denotación y, por tanto, la verificación de la verdad de la proposición es extralingüística poniendo en relación el lenguaje y la realidad.

En esta perspectiva, toda proposición tiene un *sentido*; la *denotación* no se aplica sino a las expresiones simples (palabras), no a las proposiciones o enunciados (que sean simples o compuestos). Las expresiones que tienen una denotación se asimilan a nombres propios y los nombres propios no tienen un *sentido*, ellos designan o hacen referencia a una cosa, objeto o persona.

Técnicamente, comprender el *sentido* de una proposición es saber cómo el enunciado o proposición puede ser resuel-

---

de la *Escolástica*, con la diferencia entre *vox* (signo, sonido) y el *sermo* (significado). G. Frege va a traer la problemática en 1892 (*Sinn und bedeutung: ecrits logiques et philosophiques*, Paris, Seuil, 1971, págs. 102-126), al interrogarse sobre el límite exacto de la relación de identidad (si "*A* = *B*", ¿se refiere al significado del signo como tal o a lo que él representa?). La diferencia entre Wittgenstein y Frege, es que el primero concierne la articulación de los términos y solo los nombres tienen una denotación o una referencia y la proposición articulada tiene un sentido; y para Frege, tanto las proposiciones como los nombres tienen sentido y referencia.

to de forma transparente ("Tabla de Verdad", un procedimiento de cálculo o un algoritmo), de manera que una proposición de la forma "$p \wedge q$", indique que no es verdadera sino en el caso que "*p*" y "*q*" son verdaderas simultáneamente.

b) N*aturaleza representacional del lenguaje.* El lenguaje se refiere a la realidad extralingüística, así la proposición compuesta no se verifica inmediatamente, porque ella es el resultado de operaciones de cálculos o de combinaciones cuya efectuación es independiente de la inspección de los "objetos reales". De manera que la relación *lenguaje-realidad* se produce sobre la proposición elemental, donde existe la descripción del "hecho" de la realidad y que, en consecuencia, reenvía o denota tal hecho. Por tanto, se puede decir que el lenguaje es *representacional*, porque está representando este hecho; en cierta medida es una *imagen*, la cual ayuda a determinar las condiciones de posibilidad de esta proposición (verdadera o falsa), lo que se puede llamar *sentido*. Esto significa que se pueden presentar *hechos posibles* diferentes de los *hechos reales*, concretos o físicos; se tiene, por un lado, una representación de *objeto posible* y, por el otro, el *objeto actual* o *presente*, entonces dos tipos de representación: la posible y la actual.

En consecuencia, las proposiciones elementales son simples, no pueden ser derivables[16], tienen un *sentido*; los nombres propios de la proposición (variables sustituidas y constantes) tienen una referencia o denotan el objeto real.

[16] Ellas no pueden ser identificadas a los axiomas, porque estos son verdades formales, son vacíos de contenido y no reflejan los hechos. Al contrario, las proposiciones elementales son imágenes de hechos elementales, ontológicamente posibles y son verdaderos o falsos si los hechos representados se realizan empíricamente o no.

De manera que la teoría del *sentido* lleva a la teoría de la *referencia* o *denotación* a través del cálculo (operativo); el *sentido* se reduce al objeto (extralingüístico) y a la operación o cálculo.

B) *Características del lenguaje formal*

Los dos ejes del lenguaje formal nos llevan a presentar las características específicas de este lenguaje:

• Explícito: la necesidad de presentar una anotación material, en donde el *sentido* de la proposición sea claramente expresado.

• Unívoco: sin ninguna ambigüedad, ni plurivocidad que pueden llevar a la confusión y a la indecisión: *a cada nombre corresponde un objeto*.

• Funcionalidad: es el complemento *operacionalista* de la exigencia de lo unívoco: *a todo signo corresponde una función*.

• Distinción de niveles lógicos: debe ser claro si se está tratando de lo *real* (*hechos*) o del lenguaje lógico o *lenguaje objeto* (proposición atómica, proposición compuesta, nombre…), o del *metalenguaje*, lo que evitaría la *confusión de los tipos o categorías* del lenguaje (RUSSELL[17], RYLE,…). En esta última opción, debe precisarse a qué nivel *metalingüístico* pertenece. Una distinción importante es la del *nombre propio* y los *predicados*.

El nombre propio designa un objeto (una referencia extralingüística), pertenece al *lenguaje de los objetos*, es decir, un lenguaje inmediatamente referencial (directamente articulado al real extralingüístico).

[17] BERTRAND RUSSELL, (1910).

Al contrario, el término predicativo describe una clase de objetos, es decir, que se refiere a una colección de nombres propios[18]. El predicado pertenece, entonces, a un nivel "metalingüístico": aquello que designa un predicado no es un objeto extralingüístico, sino un constructo lingüístico (l*a clase*). Las "casas son azules" pero "el azul" como cosa no existe. Así, dentro de un lenguaje lógico la naturaleza o uso de un término debería expresarse claramente[19].

C) *Estructuración del lenguaje: lenguaje objeto y metalenguaje*

Al desarrollar un sistema formal (como el del ejemplo particular), en sus definiciones, transformaciones y la evaluación de su resultado, se utiliza un tipo de lenguaje: *lenguaje-objeto*. Por ejemplo, si el sistema formal se refiere a la lógica proposicional, sobre las conexiones entre proposiciones, sabemos que aquellas dependen de la manera como las proposiciones estén construidas. Es así como las proposiciones son tratadas por intermedio de enunciados declarativos, teniendo en cuenta la articulación de símbolos vacíos de contenido, sin tener en cuenta su interpretación, y se expresan en este *lenguaje-objeto*. Por el contrario, el estudio del lenguaje y de la lógica *sobre* el lenguaje-objeto se llama *lenguaje del observador* o *metalenguaje.*

[18] Para aclarar, veamos un ejemplo en lenguaje ordinario: para afirmar que "las casas son azules", es necesario identificar ciertas cosas como "casas" y designarlas como siendo "casa". La palabra "casa" funciona como un "nombre propio" del objeto designado.

[19] Gran parte de los errores lógicos vienen de esta falta de distinción que son corrientes en el lenguaje natural.

Así, la dinámica general de un lenguaje formalizado, en un *sistema formal*, produce una cadena de símbolos construidos y estructurados conforme a reglas, que pueden ser interpretados posteriormente, como por ejemplo:

Para "1 + 1 = 4" *si y solo si* 1 = 2.

Proposición elemental "*p:* 1 + 1 = 4".

Proposición elemental "*q:* 1 = 2".

Proposición compuesta, la cadena de símbolos construidos sería: "$(q \supset p) \vee (p \supset q)$" *o* "$p \equiv q$".

De esta manera, el lenguaje de la cadena de símbolos es el *lenguaje objeto*, pero la explicación de esta cadena de símbolos necesita un lenguaje, que puede ser el español corriente, eventualmente enriquecido de anotaciones y de términos matemáticos cuyo uso será requerido, es el *metalenguaje* o lenguaje del observador[20]. Se distingue, entonces, en el seno del pensamiento lógico, una parte del lenguaje que sirve de objeto (manipulación de signos), mientras que el otro sirve de instrumento, permitiendo hablar de las expresiones del sistema y de razonar sobre ellas.

El *metalenguaje* es, por tanto, una parte de las *matemáticas informales*: se puede constatar que si en la presentación de un sistema formal, las cadenas de símbolos ocupan un lugar importante, subsiste, sin embargo, un buen número de frases redactadas en español, francés o inglés ordinario… Por ejemplo: *"si* F *es una expresión bien formada, entonces* ¬F *es una expresión bien formada"*; *"*«p ⊃ p» *es un axioma"*; estas son frases que no pertenecen al sistema propiamente

[20] El prefijo *meta* marca la diferencia que separa el lenguaje objeto propiamente dicho y el discurso sobre el sistema.

dicho, son frases sobre el lenguaje formal o *lenguaje-objeto*, no son frases del *lenguaje objeto*.

La constitución de este metalenguaje se realiza de la siguiente manera:

• La designación de signos del *lenguaje formal*, en el metalenguaje: según la indicación de HILBERT, es necesario hacer la distinción entre el nombre atribuido al símbolo formal y al trazo físico de ese símbolo, para evitar cualquier confusión entre el lenguaje y el metalenguaje: el símbolo formal *evoca* el trazo de este. Es decir, que el símbolo formal puede ser identificado por un nombre o el trazo mismo entre comillas, así el nombre p_1 será en símbolo formal "p_1"; el conector *v* será en símbolo formal: "*v*".

Los *nombres* utilizados en esta designación se llaman *nombres propios*, porque a cada símbolo corresponde un *nombre* y a un *nombre* corresponde un solo símbolo.

Se consideran y se distinguen en el metalenguaje, los elementos siguientes:

– los nombres aislados de *variables* y *constantes* del lenguaje formal u objeto; por ejemplo: "p_1", "*a*",…;

– los nombres aislados de *variables* y *constantes* del metalenguaje mismo;

– las palabras formales y expresiones bien formadas; por ejemplo: "si *F* es una expresión bien formada, $\neg F$ es una expresión bien formada".

• La realización del metalenguaje va a depender de la complejidad de la teoría que se va a formalizar; una vez que el metalenguaje está constituido, se estudian las propiedades del *sistema formal* (*S*) en dos tipos de estudios:

– los unos conciernen la estructura de (S) y sus propiedades fuera de toda referencia a una interpretación intuitiva de los símbolos de (S) y constituye lo que se llama la *sintaxis*;

– Los otros estudian (S) en sus relaciones con una interpretación intuitiva y su estudio se llama la *semántica*.

La oposición entre estos dos tipos de estudios no es totalmente absoluta: si se quiere construir una semántica rigurosa, se debe contar con una *teoría intuitiva* en un metalenguaje axiomatizado; la semántica es en ese caso *formal*, lo mismo que la sintaxis, retomando así sus mismos métodos.

Anotaciones. A nivel general, el *metalenguaje* forma parte de la estructuración de niveles lógicos que ha sido definido para corregir los problemas lógicos de autorreferencia o reflexibilidad e igualmente de indecidibilidad, presentados por las paradojas. De manera que no se pueda incluir en el conjunto E el nombre que permite designar este conjunto y el criterio o término que lo identifica, con el fin de que un nombre que designa los objetos, no sea utilizado para designarse a sí mismo como uno de los objetos; se diferencia entonces el conjunto "E" del término que permite identificar este conjunto, respetando así los diferentes niveles lógicos[21].

Algunos lógicos consideran el *lenguaje formal* como el sistema formal:

Autores, como CHURCH, reservan la apelación de *lenguaje formal* a los sistemas formales que tienen al menos una interpretación. Esta condición se justifica por el hecho de que un lenguaje, en el sentido usual de la palabra, es un instrumento de comunicación.

[21] El *metalenguaje* fue planteado por RUSSELL en su *Teoría de tipos* y de HILBERT con la presentación de la *Metamatemática*.

Al contrario, KLEENE (1987, pág. 208) considera que el *lenguaje-objeto* y el *lenguaje del observador* tienen una extensión más vasta que un sistema formal y un metalenguaje, porque estos últimos son empleados únicamente con métodos finitistas y los lenguajes no.

Capítulo III

SINTAXIS

1. Introducción

La sintaxis de un lenguaje formalizado está concernida por las expresiones del lenguaje, dejando de lado los objetos, estados o cualquier cosa que puede ser designada o referenciada por las expresiones; es decir, toda referencia a una interpretación intuitiva de sus símbolos. La relación de designación o referencia se tendrá en cuenta tangencialmente porque es la relación del sistema semántico (Carnap, 2001)[1].

La sintaxis concierne varios aspectos: el lenguaje mismo, las relaciones con otros lenguajes, la sintaxis como función deductiva.

2. Lenguaje mismo

La sintaxis fundamentalmente debe sentar las bases de la estructura del lenguaje formal, presentando las abrevia-

[1] Carnap, en el estudio de la sintaxis, toma como base los escritos y lecturas de Frege, los *Principia mathematica* de Whitehead y Russell, Hilbert y su “Metamatemática”, a la cual los lógicos poloneses, especialmente Ajdukiewicz, Lesniewski, Lukasiewicz y Tarski, le agregaron una “metalógica”. Así mismo que Gödel con el método de “aritmetización”. Wittgenstein y sus reflexiones concernientes a las relaciones entre sintaxis y la lógica de la ciencia; igualmente, los escritos de autores como Weyl, Brouwer y Lewis. Prefacio (*Foreword*, pág. xvi) de la *Logical syntax of language* (1937).

ciones necesarias para la escritura de las expresiones, las definiciones y propiedades, así:

• Introduce *abreviaciones* aligerando la escritura o sugiriendo ciertas propiedades formales; por ejemplo, se puede anotar: "$A \vee B$" la expresión "$\neg A \supset B$", donde A y B son *fórmulas* o *expresiones bien formadas*. Igualmente, se puede anotar "A & B" la expresión $\neg(A \supset \neg B)$. En estas anotaciones, los símbolos V y & pertenecen al metalenguaje. No se les atribuye ninguna propiedad particular y se podrá leer $\neg A \supset B$, cada vez que se encuentra $A \vee B$. El interés de estas *abreviaciones* aparecerá cuando se haya demostrado que ellas poseen ciertas propiedades; por ejemplo, que V & B son conmutativas, en el sentido que $(A \vee B) \supset (B \vee A)$ y $(B \vee A) \supset (A \vee B)$.

Las abreviaciones consideradas son solo convenciones de escritura.

• Presentar *definiciones nuevas*, con nuevas nociones, que deberán ser introducidas en el metalenguaje. Por ejemplo, una de las principales es la de *deducción* que generaliza, en cierta medida, la noción de *demostración*. Así, por ejemplo:

— Sea el conjunto finito H de expresiones bien formadas $H_1, H_2, \ldots, H_n$, como hipótesis.

— La cadena de expresiones bien formadas $E_1, E_2, \ldots, E_m$, podrá llamarse *deducción* en S a partir de las hipótesis $H_1, H_2, \ldots, H_n$ si cualquier expresión bien formada e_i $(1 \leq i \leq m)$, es:

✓ Una de las expresiones bien formadas $H_1, H_2, \ldots, H_n$.

✓ Un *axioma* del sistema (S).

✓ Una expresión bien formada pero obtenida por medio de las reglas del sistema (S) a partir de expresiones bien formadas E_j, E_k,…, E_l,…, ya deducidas, es decir, que $j < i$, $k < i$, $l < i$,…

— Cualquier expresión bien formada e_i $(1 \leq i \leq m)$ será llamada *derivada* de H_1, H_2,…, H_n.

— Determina las *restricciones* que dependen de la naturaleza misma del sistema formal (S).

✓ Para la regla de *sustitución*, se prohíbe efectuar sustituciones sobre las variables que figuran en las fórmulas h_1, h_2,…, h_n.

✓ La noción de *deducción* es relativa a un sistema (S), lo cual obliga a precisar si el sistema contiene o no una regla de sustitución y su naturaleza. La noción de *demostración* es también relativa al sistema (S), pero no es necesario precisar la naturaleza de las reglas del sistema (Martin, 1964, págs. 35-36).

— Define propiedades importantes de (S), así:

✓ *Consistencia*. Un sistema se dice *absolutamente consistente* si existen expresiones bien formadas que no son tesis. Así, por ejemplo:

Se encuentra que "$p \supset (\neg p \supset q)$" es un teorema del cálculo proposicional. Existe una cierta expresión bien formada (fórmula) F que se puede deducir de los axiomas al mismo tiempo que su negación $\neg F$. Al sustituir la variable "p" del teorema por "F" (regla de sustitución), se obtiene $F \supset (\neg F \supset q)$. Con F, que se supone demostrable, se obtiene por la regla de independencia (separación): $\neg F \supset q$. Como $\neg F$ es demostrable, se obtiene al utilizar una vez la regla de separación: q. Como la fórmula compuesta de la variable "q" es demostrable, se sigue que al sustituir "q" por cualquier fórmula, cualquier fórmula puede ser deducible de los axiomas. Es claro, que si una fórmula *F* (expresión bien formada) y su negación ⌐*F*, cualquier fórmula sería demostrable. Lo que lleva a concluir que si existe una expresión bien formada o fórmula que no es un teorema (no es derivable de los axiomas), el cálculo es *consistente*.

✓ *Completitud sintáctica*. Un sistema formal se llama *completo sintácticamente*, si para toda expresión bien formada *F proposicional*, o bien *F* es una tesis.

✓ *Decidibilidad.* Un sistema se llama *decidible* si existe un procedimiento *efectivo* que permite para toda *F* definir si es o no un teorema. Es *efectivo* si permite finalizar el procedimiento al cabo de un número finito de etapas, por medio de operaciones definidas con anterioridad[2]. Pero se constata que los sistemas "pobres" en medios de expresión son decidibles; por ejemplo, ninguno de ellos permite formalizar la aritmética elemental, ni el cálculo de predicados de primer orden con predicados con varias variables, tampoco es decidible.

✓ *Independencia*. Un axioma o una regla de un sistema formal es *independiente* si al *suprimirlos* disminuyen las posibilidades de demostración o de deducción.

Sea un sistema (*S*), con un alfabeto $\sum$, provisto de reglas de formación $R_1, R_2, \ldots, R_k$, de *axiomas* $T_1, T_2, \ldots, T_m$, y de reglas de deducción $R'_1, R'_2, \ldots, R'_n$, el *axioma* T_i es independiente de los otros axiomas si no es demostrable en el sistema (*S'*), con el alfabeto $\sum$, provisto de reglas de formación $R_1, R_2, \ldots, R_k$, y reglas de deducción $R'_1, R'_2, \ldots, R'_n$, y

[2] Esta definición de *efectividad* es bastante imprecisa, pero se puede volver rigurosa a partir de las *funciones recursivas*. La *efectividad* equivaldría a *recursividad general*.

de *axiomas* $T_1, T_2, \ldots, T_{i-1}, T_{i+1}, \ldots, T_m$, de *axiomas* $T_1, T_2, \ldots, T_m$. Las consideraciones de independencia tienen un gran interés en la medida que permite determinar la función de un axioma (independencia del quinto postulado de EUCLIDES, independencia del axioma de selección…).

3. SINTAXIS EN LAS RELACIONES CON OTROS LENGUAJES

En este caso, la sintaxis puede ser considerada según las relaciones entre dos lenguajes, provistos de léxicos, y de reglas de formación y transformación semejantes. Supongamos que existen dos sistemas (*S*) y (*S'*), provistos de alfabetos y de reglas de formación idénticas, en este caso, se pueden señalar ciertas propiedades entre estos sistemas (*S*), así:

• *Fuerza:* se puede decir que (*S'*) es más *fuerte* que (*S*) si toda expresión bien formada o fórmula demostrable en (*S*) es demostrable en (*S*'), mientras que ciertas fórmulas demostrables en (*S'*) no lo son en (*S*).

• *Consistencia relativa:* un sistema (*S*) *es consistente con relación al sistema* (S'), si la consistencia de (*S'*) lleva a la consistencia de (*S*). El interés de esta consistencia relativa está en el hecho de que a veces es posible demostrar la consistencia de un sistema relativamente a un sistema *débil* (aritmética), o incluso si es fuerte y se tilda de segura (teoría de conjuntos), pero sin embargo no se puede demostrar su consistencia absoluta.

4. SINTAXIS COMO TEORÍA DEDUCTIVA

La gran mayoría de las propiedades sintácticas de un *sistema formal* deben ser establecidas mediante *deducción*, constituyéndose en la principal característica de una sintaxis. Así, por ejemplo, si se pretende *demostrar* que una expresión dada es una expresión bien formada o fórmula antes de demostrar que es una *tesis*, requiere una deducción precisa.

De manera que una primera noción de la *deducción*, es la que generaliza la *demostración* y obedece a reglas usuales de la lógica, utilizando el principio de *recurrencia*; la *demostración*, en sintaxis, por el contrario, comporta elementos intuitivos y utiliza no solo los recursos habituales de la lógica, sino también aquellos de las matemáticas.

Sabemos que a partir de un *alfabeto* finito, un sinnúmero de *fórmulas* se pueden producir; es decir, que se debe proceder por *recurrencia*, demostrando que las fórmulas iniciales poseen la propiedad en cuestión y que las reglas de formación posibilitan la conservación de estas. De acuerdo con esto, se puede plantear el siguiente procedimiento deductivo, para la demostración propuesta:

• Presentación de las expresiones bien formadas iniciales, constituidas por una variable proposicional tomada aisladamente, de la forma e_i, el *teorema* es verdadero para esas expresiones.

• Si el teorema es verdadero de una fórmula F, es verdadero de la fórmula $\neg F$.

• Si el teorema es verdadero de las fórmulas F y G, es verdadero de la fórmula $(F \supset G)$.

• Las fórmulas son las *palabras* obtenidas a partir de fórmulas iniciales (axiomas) como resultado de la aplicación de las reglas de formación y transformación establecidas. De esta manera, las fórmulas pertenecientes al (S_0), poseen la propiedad enunciada.

La *deducción* propuesta sigue entonces a las reglas corrientes de la lógica y utiliza la recurrencia, como se procede en matemáticas, para su completa realización. De aquí, una dificultad o *círculo vicioso*, porque se emplea un razonamiento matemático para establecer un sistema formal, con lo cual se espera un aporte sobre la naturaleza de la deducción lógica en general y la de las matemáticas en particular. Igualmente, se han utilizado nociones propiamente matemáticas, como el *conjunto* de símbolos, se han distinguido los *subconjuntos*; también se han enumerado variables como a_1, a_2,…, como *cadena de enteros* en el *conjunto* de símbolos.

Una respuesta general consiste en tomar el "círculo vicioso" en el sentido estricto, en el cual la teoría de *sistemas formales* no se puede presentar como premisa de aquello que se quiere establecer. Sin embargo, la tarea de la lógica consiste en desarrollar un discurso coherente, con el fin de conferir, por este medio, una *consistencia* al discurso inicialmente dado.

Para establecer esta *consistencia*, es necesario precisar el grado de *fuerza* de las matemáticas que se permite emplear en sintaxis (véase cap. III, sec. 3: "Sintaxis en las relaciones con otros lenguajes"). Así, el estudio de un sistema que formaliza las matemáticas, no requiere como herramienta, una matemática tan fuerte como aquella que el sistema estudiado permite formalizar, es necesaria una herramienta matemática menos fuerte. De manera que el estudio de un sistema *fuerte* necesita como herramienta una matemática más débil.

Consecuentemente, una de las tareas de la *sintaxis*, en la deducción, es discernir el uso de razonamientos matemáticos y de caracterizar de manera muy precisa la matemática a la cual se permite recurrir, según el objetivo perseguido; por ejemplo, si se desea por medio de la formalización fundar las matemáticas en su conjunto, es decir, reconstruirlas con la mayor solidez, se puede tomar una matemática bastante *débil* y, por tanto, *evidente* (MARTIN, 1964, pág. 46)[3]. Así pues, la matemática finitista de HILBERT constituye una

[3] Es la misma situación del matemático que puede estudiar un círculo cuyo diámetro es igual a aquel del sol, trazando solo una figura muy aproximada de ese círculo.

De esta manera, se define el alfabeto de un lenguaje lógico como un *conjunto* de símbolos, donde se distinguen *subconjuntos*. Las variables se han numerado, tales $a_1, a_2, \ldots$, utilizando la cadena de *números enteros*. Pero si para construir una lógica que permite formalizar las matemáticas, se debe recurrir a nociones específicamente matemáticas, se encuentra un problema que debe ser resuelto (objeción de POINCARÉ a los primeros trabajos de HILBERT).

Esta es la posición adoptada por HILBERT (HILBERT, D. y BERNAYS, P. *Fondements des mathématiques*, t. I, 1934, parg. 2, pág. 32), que pretende fundar las matemáticas por medio de la *formalización*; es decir, reconstruirlas con máxima solidez, toma como instrumento *metamatemático* una *matemática débil* como la aritmética finitista. En esta *aritmética concreta*, los objetos y sus propiedades se presentan con plena certeza, abstracción hecha del *trazo mismo*. Una *cifra* (*Ziffer*) se considera como una *abreviación* (para una cadena de “1”: 2 abrevia la cadena “1,1”…). Así, las operaciones de adición y de multiplicación se definen concretamente a partir de esta noción de *cifra*. Sus propiedades se establecen por medio de una *inducción* completa apoyada en una *construcción efectiva* de la cadena de enteros. Aquí, hay una deducción que se basa en la construcción concreta de cifras.

En analogía a la designación de HILBERT, los lógicos de Warsaw (LUKASIEWICZ y otros) han hablado del “cálculo metaproposicional”, de *metalógica*. Sin embargo, la palabra “metalógica” es una designación para el subdominio de la sintaxis que utiliza frases lógicas en un sentido estricto.

aritmética mínima y, en este sentido, si se desea adoptar, por ejemplo, el vocabulario de la teoría de conjuntos, es una teoría de conjuntos mínima, la que hay que definir con anterioridad y reconstruir luego la aritmética por medio de la noción de conjunto finito, no a través de la noción de un conjunto *general*.

Otra dificultad que se presenta, es la de la delimitación de una matemática intuitiva, la cual por su naturaleza imprecisa puede convertirse en una tarea imposible, pero que, sin embargo, un acercamiento a un grado de precisión sería de mucha utilidad para una sintaxis deductiva rigurosa; de manera que la matemática que se emplearía se definiría sin la menor ambigüedad y fuera de cualquier "círculo vicioso". Una de las respuestas a esta problemática, es la de formalizar el *metalenguaje*, el cual sabemos pertenece a la matemática *informal*. Tarski (1931), con el objetivo de darle un sentido preciso a la *semántica*, pretende definir formalmente la *sintaxis* de un sistema, pero presenta solo un *esquema de formalización* más que una formalización completa, se contenta con una formalización parcial: precisa los puntos de litigio enumerando cuidadosamente las prohibiciones que deben respetarse. Así mismo, en el metalenguaje se designan los puntos que pueden ser presentados en lenguaje natural y cuáles se podrían formalizar, aclarando que la totalidad de la formalización conduce a volver muy pesada la escritura y el razonamiento mismo, por tanto no va más allá de la presentación de un esquema en esta formalización.

Las respuestas a este "círculo vicioso" llevan a plantear una *sintaxis elemental* y, además, una *sintaxis como cálculo*.

A) *Sintaxis elemental*

La distinción entre el grado de *fuerza* de la sintaxis conduce a establecer la distinción entre una *matemática* débil y aquello que le concierne a una *matemática fuerte*. El primer caso, comprende lo que se llama *sintaxis elemental*, que utiliza métodos finitistas, en el sentido de HILBERT (1934). Lo cual supone como condición previa, que las reglas de formación y de deducción permitan, por ejemplo, decidir si un término o términos, al cabo de un número finito de etapas, es una *palabra* del lenguaje, si una palabra dada es una *fórmula*, si una fórmula es un *axioma*, o si una cadena dada de fórmulas es una *demostración*. Recíprocamente, el uso exclusivo de métodos finitistas asegura que el empleo de reglas derivadas, que se pueden establecer por medio de demostraciones sintácticas, deje intactas las posibilidades de decisión, propuestas en primera instancia. Es decir, la propiedad de *homogeneidad*, que es a la vez verificada por los axiomas y conserva las reglas de inferencia (toda conclusión sacada de las premisas homogéneas es entonces homogénea), de donde se concluye por recurrencia, que todos los teoremas del sistema son homogéneos.

La sintaxis elemental presenta así, ventajas y límites:

• Se trabaja con una matemática mínima, evidente e incontestable, pero insuficiente en ciertos casos: algunas propiedades de *sistemas efectivos* no pueden ser demostrables por los métodos finitistas, en cuyo caso, estaríamos concernidos por lo que se llama la *sintaxis teórica*. Esta se define como *teoría matemática general*, que compete a varios sistemas formales, que tienen como objeto de estudio, su estructura formal, sin tener en cuenta su interpretación. Una sintaxis teórica no está condicionada por el grado de fuerza de la matemática empleada.

• La sintaxis elemental, por su solidez, ha permitido obtener un gran número de resultados importantes que van de los más evidentes y de los más difíciles a establecer, como por ejemplo:

— las propiedades de la deducción, como la reflexibilidad, con la cual se permite adjuntar a las hipótesis de una deducción todas las hipótesis que se deseen; la eliminación de hipótesis redundantes, la inversión del orden de las hipótesis y la transitividad;

— el teorema de la deducción, la no contradicción, la completitud sintáctica y la decidibilidad del cálculo de proposiciones.

Anotaciones. Hilbert (1968, 1967) sienta las bases de esta *sintaxis elemental*, pues en búsqueda de una prueba de consistencia (las inferencias lógicas no pueden conducir a demostrar, a partir de axiomas, a la vez un enunciado y su negación) para fundamentar las teorías matemáticas, presenta una prueba *sintáctica*. Las fórmulas de la aritmética son consideradas como concatenaciones de símbolos o características morfológicas sin contenido. Presenta la propiedad de *homogeneidad*, por medio de la "conservatividad" de las reglas de inferencia (toda conclusión sacada de las premisas homogéneas es entonces homogénea), así, todos los teoremas del sistema son homogéneos y la negación de una fórmula homogénea no es ella misma homogénea, en ese sistema, una fórmula homogénea, e igualmente, la negación de un teorema no es ella misma demostrable. (Esta prueba tiene un problema de "circularidad": crítica de Henri Poincaré, "Les mathématiques et la logique", *Revue de métaphysique et de morale*, vol. 14, 1906, págs. 17-34: en donde, al principio del razonamiento por inducción, aplica un argumento que reposa sobre el principio de inducción).

Hilbert en 1927, presenta la "teoría de la demostración", con una vasta formalización mecanicista de la empresa matemática y un método general de decisión que permitiría, una vez las matemáticas redactadas en un formalismo adecuado, determinar de manera segura en un número finito de etapas si una fórmula dada es o no un teorema (*problema de la decisión*). Sin embargo, reconoce luego, que la prueba de consistencia no puede ser absoluta en sentido estricto, y, al contrario, debe contar con un conjunto de métodos "elementales" admitidos sin justificación, intuitivamente correctos.

Hilbert califica de *elemental*, un enunciado que se relaciona con una clase C de objetos *concretos* (cadenas de cifras, conjuntos de símbolos), provisto de relaciones decidibles por simple constatación.

De esta manera, la cuantificación universal sobre una cierta clase es considerada como *elemental*, si se interpreta de la manera siguiente: "$\forall x\Phi(x)$" significa que si nos presentan concretamente un objeto prototipo de la clase C, podríamos constatar que este objeto "prototípico" de la clase C, se podría definir como un objeto de Φ. El conjunto R de enunciados elementales (Hilbert los llama "reales" o "finitistas") contiene todas las proposiciones del tipo $\forall x_1\ldots, \forall x_n\Phi(x_1,\ldots, x_n)$, en donde Φ es una relación decidible (muchos de los enunciados matemáticos importantes son de ese tipo; por ejemplo, el último teorema de Fermat, el teorema de cuatro colores, la famosa hipótesis de Riemann).

B) *Sintaxis como cálculo*

Con las propiedades referentes a la sintaxis deductiva y la elemental, se perfila una *sintaxis* como *cálculo*, que presenta CARNAP (1937), en su *sintaxis lógica,* en donde es la aplicación de reglas formales sobre *signos* desprovistos de contenido, que los lenguajes pueden desarrollar los cálculos aplicados a la deducción y, por lo cual, estos lenguajes evolucionan en su estructura sintáctica. Así, son estas reglas que retoman toda su importancia, porque van a determinar tanto la *formación* de las "expresiones bien formadas" (EBF), con sus propiedades, la relación de signos, como las posibilidades de estructuración entre ellos, lo mismo que las reglas sintácticas que aseguran la *transformación* de las expresiones en una deducción. Estas reglas rigurosamente planteadas, son la base de las reglas que van a conducir el cálculo. Se está, en el dominio de lo operativo, con un procedimiento efectivo, casi mecánico, que proporciona los resultados de la deducción. Igualmente que para la sintaxis deductiva, el procedimiento más simple para la construcción de un cálculo consiste en seleccionar algunas frases como frases primitivas (algunas llamadas postulados o axiomas) y algunas reglas de inferencia.

Las *reglas de transformación* muestran cuáles "manipulaciones" son toleradas y cuáles no; es decir, las reglas que no alteran la validez dada como punto de partida. Por ejemplo, la condición para que una regla de sustitución, que puede reemplazar en una proposición válida una variable por una EBF y que puede producir una nueva proposición válida.

El uso de estos dos tipos de reglas asegura la presentación del esquema deductivo, como una cadena de EBF, expresiones que pueden ser: un axioma o la consecuente de tal esquema, en donde el antecedente precede al consecuente, la última proposición del esquema es un teorema (objeto de la demostración y constituye una EBF; muchas veces puede ser reutilizada en otra demostración y puede ser una proposición refutable cuya negación se deriva del sistema).

Anotaciones. La *sintaxis formal* ha sido la base de la mayoría de los sistemas matemáticos, cuyas pruebas están relacionadas con métodos efectivos; así mismo, las propiedades centrales de la sintaxis, tales como las de *consistencia*, *completitud* y *decidibilidad*, son la base de cualquier sistema formal y base para la *semántica formal*.

Capítulo IV

SEMÁNTICA

1. Introducción

La construcción de un *sistema formal*, conduce a la pregunta sobre la correspondencia de su interpretación en el seno de una teoría menos abstracta. El propósito es, el de relacionar las dos teorías (la formal y la intuitiva), por medio de una estructura que exprese el sentido o varios de los sentidos posibles de la interpretación. De ahí, el nombre de *semántica* dada a estos estudios, por oposición a los de la *sintaxis*. Es difícil trazar una frontera exacta entre la semántica y la sintaxis e incluso, muchos estudios de la semántica se realizan desde el campo mismo de la sintaxis.

Así, a la semántica le corresponde la tarea de confirmar o precisar un concepto vago; por ejemplo, si el cálculo de proposiciones o el cálculo de predicados de primer orden, expresan de manera conveniente, en lenguaje natural, lo que se entiende por ellos en lógica formal. Carnap (1947) llama *explicandum* (nuevo concepto interpretado en lenguaje natural) y *explicatum* (el concepto de base).

2. Nociones generales de la semántica

La *semántica* de un sistema formal, se puede entender como la interpretación[1] o interpretaciones de este sistema

[1] *Interpretación* tomada en un sentido más general que el de *traducción*, que podría pensarse como una noción más *mecánica* y reducida.

en una o varias teorías intuitivas, que corresponden a una *realidad* o *varias realidades*, cuyo contexto es el del *lenguaje extralingüístico*, por oposición al *lenguaje formal*. Es decir, que para el desarrollo de una interpretación, es necesario contar con el estudio de las relaciones de los *signos* y su *significado*, lo que constituye los innumerables estudios realizados sobre el *sentido* y la *referencia* y sobre la *verdad* de las expresiones que se interpretan y la validez de esa interpretación.

Anotaciones. Como base conceptual del desarrollo de la semántica, es necesario contar con un *análisis del significado* de las expresiones del lenguaje formal para interpretar, con el fin de comprender el significado de la expresión dada y la manera como se aplica en el mundo real. En ese sentido, se puede plantear la diferencia entre *sentido* y *referencia* (o *denotación*) (véase cap. II, sec. 4, apdo. A: "Ejes principales del lenguaje formalizado"); puede presentarse así:

FREGE propone una teoría tripartita del *signo*, que comprende el *signo* en su materialidad (*Zeichen*), el *sentido* (*Sinn*) y la *referencia* (*Bedeutung*). El *sentido* es el modo como se *da* la referencia; es decir, la manera como se hace referencia al objeto, y la *referencia* es el objeto mismo. Este análisis permite explicar la diferencia entre las identidades $a = a$, que no tiene valor como conocimiento mismo y $a = b$ que aporta una información. Si a y b son diferentes, ¿cómo se puede decir que son iguales, y cómo su igualdad (identidad) puede ser informativa? La respuesta reside en el doble hecho de que a) la igualdad reside en la identidad de la referencia (a y b reenvían al mismo objeto de referencia), y b) el valor informativo reposa sobre la diferencia del sentido. La expresión "estrella de la mañana" no significa lo mismo que la expresión "estrella de noche"; la una reenvía al hecho que se ve una estrella en la mañana y la otra reenvía a la noche. Incluso, si las dos expresiones designan o denotan el planeta Venus, ellas no reenvían de la misma manera, son dos formas de mostrar la referencia.

Este análisis del signo se extiende a la expresión o proposición misma. El sentido de una proposición es un *pensamiento* (*Gedanke*) y su *referencia* es un valor de verdad (verdadero o falso); así el pensamiento se vuelve objetivo, no se puede confundir la conciencia de una proposición con su verdad (una proposición no cesa de ser verdadera, porque no la pensamos). Estas bases lógicas adquieren un estatuto objetivo, porque FREGE admite un principio realista y antisicologista, de manera que los signos y el concepto toman valor, por el principio de *contextualización*, según el cual los signos adquieren significado en el marco de la proposición (principio de proposicionalidad).

El *sentido* y la *referencia* de una proposición compuesta representan la *función* de los *sentidos* y *referencias* de sus componentes: expresión del principio de *composicionalidad.*

La lógica de PORT ROYAL (1662), distingue entre la *comprensión* y *extensión.* En donde la comprensión o intensión es el significado propio del objeto y la extensión es el sujeto u objeto que comporta esa propiedad.

Como continuación de esta distinción fregeana y la de *comprensión* y *extensión*, el análisis del significado puede dividirse en teoría de la referencia (nominación, denotación, extensión) y teoría del significado, como tal (intensión), distinciones que están presentes en la interpretación o en el desarrollo de la semántica.

A) *Noción de interpretación formal*

Una *interpretación i* de un lenguaje, de primer orden, está formada por (D_i, R_n^k), donde D_i (el dominio de *i*) es un conjunto no vacío y R_n^k es un número *n* de relaciones *k* definidas sobre D_i. Entonces, (D_i, R_n^k) es una estructura en el sentido formal del término; en el caso del lenguaje *L* de S_0, se considera que D_i comporta una colección de objetos diferenciados a_1, a_2,… Se distinguen, igualmente, entre las relaciones R_n^k $(k, n > 0)$ un *conjunto* F_n^k $(k, n > 0)$ de funciones definidas en D_i.

El conjunto D_i es el conjunto de los valores que pueden tomar las variables individuales x_1, x_2 del lenguaje *L* de S_0; los elementos de D_i son los objetos interpretados que corresponden al lenguaje *L.* La colección de elementos diferenciados a_1, a_2,… de D_i, es el conjunto de los objetos que corresponden a las constantes individuales a_1, a_2,… de *L*.

Así, a partir del sistema formal S_0 se puede plantear como objetivo, interpretar o presentar la relación (R_n^k) de este sistema con una teoría enunciada en lenguaje natural; es decir, presentar el desarrollo de su *semántica*. Para ello, suponemos que esta teoría es coherente, donde se distinguen las expresiones o aserciones verdaderas de aquellas que son falsas. La adecuación, la interpretación del sistema formal y la teoría intuitiva resultante, pueden presentarse así:

• Correspondencia entre símbolos del sistema formal y las palabras de la teoría intuitiva.

• Análisis de las expresiones traducidas.

• Determinación de la correspondencia entre las tesis del sistema formal y las proposiciones verdaderas de la teoría intuitiva, y viceversa.

B) *Correspondencia entre símbolos*

La correspondencia entre los símbolos del sistema formal, palabras o grupos de palabras referidas al vocabulario de la teoría intuitiva, puede presentarse como una *traducción* realizada por intermedio de un tipo de *diccionario* constituido formalmente, según condiciones del sistema formal y la teoría intuitiva, así:

• Definición de las reglas que permiten obtener, a partir de la traducción de expresiones bien formadas o fórmulas del lenguaje *L*, símbolos de la teoría intuitiva (D_i). A ciertas expresiones bien formadas corresponden frases con *sentido* (posibilidad de ser verdaderas o falsas) del lenguaje ordinario, de acuerdo con las reglas. Estas serían las frases "lógicamente verdaderas" en lógica usual.

• Distinción en cada una de esas expresiones bien formadas de dos clases de símbolos:

— Las *constantes* ($a_1, a_2, \ldots, a_n$) y *operadores*, de varias clases a las cuales corresponde el mismo valor a lo largo de la traducción. La traducción es, entonces, biunívoca e invariable; es decir, que a cada símbolo corresponde una sola traducción, a lo largo de la interpretación; así por ejemplo, para los operadores:

✓ "$\neg$" se traduce por "no"

✓ "$\supset$" se traduce por "implica"

✓ "( )" se traduce en "paréntesis"

— Las *variables* ($x_1, x_2, \ldots, x_n$) a las cuales conviene hacerles corresponder diferentes valores a lo largo de la traducción.

• Correspondencia de cada variable y constante a una *frase* o *expresión* que sea dotada de *sentido* (susceptible de ser *verdadera* o *falsa*). Así, para una expresión de la *teoría intuitiva* con sentido y que contiene *n variables* distintas, tendremos tantas traducciones posibles como formas posibles de escoger *n* frases (diferentes o no) en el sistema formal.

• Traducción de una *expresión compuesta*, realizada a partir de los símbolos (simples) que la constituyen, de manera que a una *palabra formal* (sistema formal) se hace corresponder una cadena de símbolos que se ha obtenido de la interpretación de símbolos concatenados, de la teoría intuitiva. Las traducciones son estructuradas en el mismo orden que los símbolos que les corresponden en la palabra formal. Por ejemplo:

— "$a_1 \supset a_1$" corresponde a "hace sol *implica* hace sol"

— "$a_2 \supset a_2$" corresponde a "2 + 3 = 5 *implica* 2 + 3 = 5"

— "$a_3 \supset a_3$" corresponde a "las vacas vuelan *implica* las vacas vuelan"

C) *Análisis de expresiones traducidas*

Con el análisis de las frases o expresiones obtenidas por traducción, se pretende conocer, si estas frases son verdaderas o no, en la lógica que se está interpretando (S_0).

Para los enunciados presentados como lógicamente verdaderos, serán verdaderos sea cual fuere la traducción adoptada para a_1, a_2 o a_3. Al contrario, la traducción de $\neg(a_1 \supset a_1)$ es falsa en cualquier caso; pero para el caso "$a_2 \supset a_3$", puede ser tanto verdadera como falsa, según la traducción adoptada para a_2 o a_3, lo que correspondería a "2 + 3 = 5 *implica* las vacas vuelan", es falsa por el principio que lo verdadero implica lo falso; para el caso de "$a_3 \supset a_2$", que correspondería a "las vacas vuelan *implica* 2 + 3 = 5", es lógicamente verdadera por el principio que lo falso implica lo verdadero, incluso si el contenido de a_3 no corresponde a ninguna realidad.

Pero se presenta una objeción a esta parte del análisis, porque no es evidente a primera vista que la proposición: "las vacas vuelan *implica* 2 + 3 = 5" sea lógicamente verdadera; es necesario, entonces, presentar las reglas lógicas para llegar a la respuesta. Se ve que hay una dificultad para determinar a primera vista la respuesta, que es la falta de rigurosidad en la presentación de las proposiciones y sus reglas; es decir, la falta de una *axiomática formal* para determinar de manera competente, la traducción de un símbolo en una teoría intuitiva. Esta dificultad se acrecienta a medida que las expresiones formales se vuelven más complejas, porque las expresiones resultantes de las traducciones no se pueden apreciar directamente.

Así, si se traduce $\neg(a_2 \supset a_3) \supset (a_3 \supset a_2)$, haciendo corresponder a a_2: 2 + 3 = 5 y a a_3: *las vacas vuelan*, para discernir la proposición resultante es necesario examinar atentamente cada proposición simple. No se cuenta con un método que determine de manera general, si esta proposición sigue una ley lógica que defina si es verdadera o falsa; es decir, que no hay un procedimiento *decidible* para llegar a ese resultado, lo que hace necesario demostrarlo, a partir de un esquema deductivo riguroso.

D) *Determinación de la correspondencia entre las tesis del sistema formal y las proposiciones verdaderas (tautologías) de la teoría intuitiva, y viceversa*

El análisis presentado nos sitúa frente a las expresiones de la teoría intuitiva traducidas (D_i), lo que posibilita la solución de un tema importante en semántica formal que es la correspondencia entre las tesis o axiomas del sistema formal y las proposiciones verdaderas de la teoría intuitiva (tautologías), y viceversa (Martin, 1968, pág. 59). De esta manera se va a examinar la forma de:

• Determinar la correspondencia entre las tesis del sistema formal y las proposiciones verdaderas (tautologías) de la teoría intuitiva, lo que asegura que las demostraciones formales conducen a proposiciones verdaderas de la teoría intuitiva;

• Determinar la correspondencia entre proposiciones verdaderas (tautologías) de la teoría intuitiva y las tesis formales, lo que asegura que el sistema formal realiza completamente las posibilidades de demostración de la teoría intuitiva.

Para el primer aspecto, el conjunto de proposiciones verdaderas de la teoría intuitiva, en correspondencia con las tesis formales, puede ser infinito y el problema no se resolvería con una lista exhaustiva para demostrar que todas las tesis del sistema formal y las proposiciones verdaderas de la teoría intuitiva se corresponden. Es necesario utilizar un método recurrente, para que se pueda demostrar:

a) que las tesis iniciales, es decir, los axiomas, de S_0 se corresponden con las expresiones verdaderas (intuitivas) y que las reglas de deducción de S_0 *conserven* esta propiedad de correspondencia[2];

[2] Los axiomas de un sistema formal no tienen ninguna evidencia intuitiva, no son escogidos en razón de la evidencia particular de su interpretación, sino en razón de la facilidad importante con la cual se pueden deducir de ellos las tesis las más corrientemente utilizadas.

b) que si en una demostración los antecedentes son una tesis, los consecuentes obtenidos por aplicación de reglas inferenciales en vías de una interpretación, lo son también[3].

Se evidencia, en este primer caso, que el número de los axiomas (formales) es con frecuencia menor que aquel de los consecuentes (intuitivos) que se pueden colegir y la afirmación que todas las tesis o axiomas corresponden a expresiones verdaderas interpretadas, reposa solamente sobre una generalización formulada a partir de numerosos ejemplos, producto del método recurrente planteado. Es decir, que no se puede tener una respuesta general, en tanto que cada una de las expresiones sea analizada.

Así mismo, para el segundo aspecto, si todas las expresiones verdaderas (teoría intuitiva) se corresponden con las tesis (sistema formal), se puede decir que el conjunto de las proposiciones lógicamente verdaderas no se pueden definir con precisión y solo se puede proceder por medio de pruebas para asegurar en la mayoría de los casos posibles:

• Que a una proposición lógicamente verdadera corresponda una proposición interpretada verdadera en virtud del diccionario utilizado;

• Que todas las proposiciones verdaderas interpretadas (intuitivas) así, son *tesis* (formales), pero una verificación de este género puede servir de base a una *inducción* y no se podría llegar a demostrar rigurosamente y de manera general que el sistema considerado posee tal propiedad semántica.

Anotaciones. Es necesario decir que el contenido de las proposiciones, que en la interpretación corresponden a las va-

[3] Es difícil justificar esta demostración de manera rigurosa, mientras se esté en el dominio de la semántica intuitiva.

riables del sistema formal S_0, no desempeña ningún papel. Solo importa el hecho de que estas proposiciones sean, según el caso, verdaderas o falsas. Si se estima verdadera la proposición "las vacas vuelan *implica* 2 + 3 = 5", no es porque vemos una relación entre la morfología de las vacas y una verdad aritmética, pero porque nos inclinamos a aceptar que de lo falso se puede deducir lo verdadero y que en el caso contrario, de lo verdadero se deduce lo falso, no se puede sostener.

Por otro lado, vemos que la respuesta a las dos problemáticas planteadas, no tienen una respuesta directa y es necesario, en consecuencia, analizar cada una de las expresiones correspondientes según las reglas lógicas para definir la condición de tesis de las proposiciones verdaderas (teoría intuitiva), y viceversa.

En este sentido, se puede ver que las respuestas no corresponden a un método totalmente formalizado y que para obtener las propiedades semánticas necesarias a la interpretación, es importante recurrir a otro método, más preciso, que es el de las variables de verdad o el método de modelo.

a) M*étodo variables de verdad.* La interpretación, en lugar de hacerse sobre las *expresiones* o *proposiciones particulares* del sistema, se puede hacer sobre variables, pero solo sobre los dos valores de verdad V y F. Así, las "tablas de verdad" ilustran de manera directa el proceso de composición lógica donde el valor de verdad de una expresión está referido únicamente a la función de verdad de sus componentes atómicos o simples. Es un método que permite determinar el valor de verdad de la expresión o proposición compuesta sobre la base del valor de verdad de sus componentes. Toda proposición contiene *n símbolos* enunciados, expresiones: $a_1, a_2, \dots, a_n$ que corresponden a una función de verdad a *n* argumentos. La forma de proposición $(a_1 \wedge \neg a_2) \supset a_2$, por ejemplo, genera una función de valor de verdad que puede ser representada por una tabla de verdad compuesta de 2^n ($n = 2$) líneas, donde cada línea indica una asignación posible de valor de verdad para $\{a_1, a_2\}$, lo mismo que el valor de la función para esta asignación y las asignaciones posibles de valor de verdad para los componentes atómicos (véase cap. II, sec. 4, apdo. A, lit. a: "Naturaleza proposicional y analítica del lenguaje").

De manera que si se trata de interpretar una expresión compuesta, tal que:

$(\neg a_1 \supset \neg a_2) \supset (a_2 \supset a_1)$ (1)

Donde $a_1 = V$ y $a_2 = F$

La expresión se traduce así:

"(no-V *implica* no-F) *implica* (F *implica* V) (2)

La pregunta es: ¿si la expresión (1) tiene el valor V o el valor F?

Para responder, es necesario:

a) conocer el valor de *no*-V y de *no*-F;

b) conocer el valor de una expresión de la forma "*A* implica *B*" cuando *A* y *B* son V o F.

Se tienen en cuenta, las reglas siguientes:

no-V = F ; no-F = V

"*A* implica *B*" = F si *A* = V y *B* = F

"*A* implica *B*" = V en todos los otros casos

"=" significa la identidad y la posibilidad de sustituir cualquier expresión situada a la izquierda.

De esta manera, de la expresión (2) se deduce sucesivamente:

"(F *implica* V) implica (F *implica* V)"

"(V *implica* V)"

V

$a_1 = F$; $a_2 = V$

"(no-V *implica* no-F) implica (F *implica* V)"

$a_1 = V$; $a_2 = V$

$a_1 = F$; $a_2 = F$

Al hacer corresponder a las variables de la proposición (1), los valores de verdad V y F, permite en pocas etapas todas las posibilidades de traducción adoptada; la expresión (2) toma el valor V. Este procedimiento proporciona la herramienta para determinar las expresiones de S_0 que son expresiones verdaderas o *tautologías* y *contradicciones*, o las *tesis* o *axiomas* según los valores atribuidos a las variables.

Esta forma de proceder tiene la ventaja que permite tratar con rigor las preguntas semánticas sin salir del marco de la sintaxis de S_0; es posible dar a la teoría de *funciones de verdad* una forma puramente matemática. Al considerar un conjunto *D* con dos elementos {V, F} y dos funciones ⌐ y ɔ, la primera con un argumento (definido en el conjunto D_0), la segunda con dos argumentos (D_0 *x* D_0), el uno y el otro toman valores en D_0. Estas dos funciones se representan por tablas de verdad clásicas, donde *A* y *B* designan expresiones cualquieras:

A	⌐A
V	F
F	V

A	B	A ⊃ B
V	V	V
V	F	F
F	V	V
F	F	V

El objetivo es encontrar un método que permita asociar a cada fórmula de S_0, uno de los dos valores V o F, donde el valor de la fórmula depende únicamente de los valores atribuidos a cada variable proposicional que figuran en ella.

A la pregunta: ¿todas las tesis de (S_0) son expresiones verdaderas, en todos los casos (tautologías) en la teoría intuitiva? Es suficiente proceder por recurrencia y demostrar:

• que las tesis iniciales de (S_0), es decir, los axiomas, son tautologías en la teoría intuitiva;

• que las reglas por medio de las cuales se deducen las tesis o los axiomas en S_0, conservan esta propiedad en la teoría intuitiva. La demostración asegura la consistencia en los dos sentidos:

— (S_0) es absolutamente consistente, lo mismo que D_i del lenguaje de la teoría intuitiva. En efecto, si todas las tesis son tautologías, una expresión que no es una tautología no es una tesis. Una variable proposicional tomada aisladamente es una expresión y no una tautología, porque ciertas asignaciones pueden aplicarla sobre el valor F. Existe en (S_0) y en D_i, una expresión que no es una tesis.

— (S_0) y D_i son consistentes con respecto a la negación. Si A es una tesis, $\neg A$ es necesariamente una contradicción y no una tesis. Es imposible, cualquiera que sea A, de tener A y $\neg A$.

A la pregunta semántica: ¿todas las tautologías del sistema formal (S_0) son tesis (teoría intuitiva)? Se puede responder también afirmativamente apoyándose en una demostración por recurrencia. Por otro lado, si (S_0) es semánticamente completo con respecto a la interpretación en términos de función de verdad, esta propiedad de "completitud" permite establecer dos otras propiedades importantes de (S_0):

- (S_0) y D_i son sintácticamente completos;
- (S_0) y D_i son decidibles. A una tautología o expresión verdadera de (S_0) y D_i, es posible determinar al término de un número finito de etapas si ella es o no una tesis (teoría intuitiva). Es necesario examinar si toma el valor V para todas las asignaciones posibles. Entonces, si en la fórmula figura n variables distintas, estas asignaciones son en número de 2^n, su valor es finito y es posible examinar una después de la otra (son enumerables).

b) M*étodo del modelo*. Sea un sistema S formal efectivo[4], se llama *modelo* de este sistema todo campo de interpretación C constituido de *dominios no vacíos*, que cumple las cuatro condiciones:

- Todo axioma de S es válido en C.
- Si una regla permite en S deducir B de $A_1, A_2, \ldots, A_n$.
- Si $A_1, A_2, \ldots, A_n$ son válidas en C, B es válido en C.
- Existe por lo menos una expresión de S que no es válida en C.

[4] Sistema o conjunto de las fórmulas del conjunto de las demostraciones son recursivas mientras que el conjunto de tesis es recursivamente enumerable (Roger Martin, *Logique contemporaine et formalisation*, Paris, PUF, 1964, pág. 83).

Esta definición exige no solo que las tesis de *S* sean válidas en *C*, sino que también las relaciones de *deducibilidad* en *S* tengan su imagen en *C*.

La noción de *modelo* permite formular de manera más precisa, pero general, lo que se llaman problemas de la semántica, a saber:

• ¿El sistema admite un *modelo*?

•¿Si *M* es un modelo de *S*, toda expresión verdadera en ese modelo es una tesis de *S*?

• Esta expresión permitiría igualmente enunciar las relaciones que unen las propiedades semánticas de un sistema a sus propiedades sintácticas.

En efecto, si un sistema *S* admite un modelo *M*, es *consistente* en los sentidos que se han indicado:

• En sentido absoluto: porque toda tesis de *S* es válida en *M*, una fórmula que no es válida en *M* no puede ser una tesis o axioma. Entonces, se supone que existe una expresión de *S* no válida en *M*, por lo cual existe al menos una fórmula de *S* que no es una tesis.

• *S* es también consistente con respecto a la negación. Si se puede establecer en *S*, a la vez *A* y ⌐*A*, resulta que *A* es válida en *M* al mismo tiempo que ⌐*A*, lo que prohíbe la definición misma de *modelo*. Se concluye que si *S* admite un modelo, no puede tener a la vez *A* y ⌐*A*[5].

A la pregunta: ¿en un *modelo* dado de *S*, toda expresión válida en este modelo es una tesis de *S*? La respuesta toca la completitud semántica. Así, *S* será semánticamente com-

[5] La expresión constituida por una variable de proposición tomada aisladamente no puede ser una tesis, se necesita que sea válida.

pleto con respecto a *M*, si existe al menos un *modelo* con respecto del cual es *completo*.

Pero se puede presentar el caso, en el que un sistema puede ser *completo* con respecto a un modelo y no en relación con otro, se puede pensar, entonces, en un *modelo principal*, un modelo en relación al cual este sistema es *completo semánticamente.*

E) *Propiedades semánticas*

A partir del análisis y del desarrollo de la correspondencia entre tesis y expresiones verdaderas (tautologías), y viceversa, establecidas por el método de variables de verdad y el método de modelo, se plantean las principales propiedades semánticas, así:

• *Conservación*: las reglas por medio de las cuales se deducen las tesis a partir de los axiomas conservan esta propiedad.

• *Tautología*, se obtiene el valor de verdad para todas las asignaciones posibles de una expresión o proposición; su opuesta la *contradicción*, se obtiene el valor de verdad F a todas las asignaciones posibles del valor de verdad de la expresión y sus componentes; la *indeterminación* que no es ni tautológica ni contradictoria.

• *Consistencia* de (S_0), como *absolutamente consistente* y *consistente con respecto a la negación*; su opuesta la *inconsistencia* o *no contradicción*.

• *Completitud* semántica y sintáctica; su opuesta la *incompletitud.*

• *Decidibilidad:* se cuenta con un procedimiento efectivo para realizar la interpretación entre el sistema formal y la teoría intuitiva; su opuesta la *indecidibilidad*.

a) P*ropiedades semánticas de base.* Las propiedades semánticas que van a dar soporte a cualquier desarrollo de la semántica de un sistema formal se pueden resumir en tautológica, contradictoria, indeterminada, consistente e inconsistente, sabiendo que el resto de propiedades expuestas son aplicadas en este desarrollo, pero son estas propiedades que soportan, en primera instancia, su aplicación.

a') Tautología. Una forma de proposición A es *tautológica*, si A obtiene el valor de verdad V en todas las asignaciones posibles de valor de verdad de sus componentes atómicos, es decir, de las variables que ella contiene. Se dice, así mismo, que esta proposición es lógicamente verdadera. Por ejemplo:

$$A = (B \vee \neg B)$$

$$A = (A_1 \supset A_2) \vee \neg(A_1 \supset A_2)$$

Para cualquier enunciado que se sustituya a A y a B, la expresión que resulta obtendrá el valor de verdad V, bajo todas las asignaciones posibles de valor de verdad de sus componentes atómicos.

b') C*ontradicción o antilogía.* Una forma de expresión o enunciado A es una contradicción, si A obtiene el valor de verdad F en todas las asignaciones posibles de valor de verdad de sus componentes atómicos, es decir, a las variables de enunciado que ellas contienen. Se dice que una forma que es contradictoria es *lógicamente falsa*. Por ejemplo:

$A = (B \wedge \neg B)$, que se puede escribir: $A = (a_1 \wedge \neg a_1)$ y cuya tabla de verdad:

a_1	$\wedge$	$\neg$	a_1
V	F	F	V
F	F	V	F

Ilustra el carácter contradictorio de esa forma de expresión o enunciado; de lo cual A es falso si $\neg A$ es verdadero, de la misma manera que A es verdadero, si $\neg A$ es falso.

c') *Indeterminación.* Una forma de expresión o enunciado A es lógicamente *indeterminado*, si A no es ni tautológico ni contradictorio. Por ejemplo:

$$a_1$$

Según la tabla de verdad:

a_1
V
F

d') C*onsistencia.* Un conjunto finito no vacío de expresiones o enunciados $E = \{a_1, a_2, \ldots, a_n\}$ es *consistente*, si existe al menos una asignación de valor de verdad de los componentes atómicos de $a_1, a_2, \ldots, a_n$ bajo la cual $a_1, a_2, \ldots, a_n$ son todos verdaderos.

El conjunto $E = \{p_1 \, v \, p_3\}$ es un conjunto de enunciados consistente y se verifica si la tabla de verdad de los enunciados que la componen indica que existe una asignación de valor de verdad sobre la cual los enunciados son verdaderos, así:

p_1	v	P_3
V	V	V
V	V	F
F	V	V
F	F	F

La propiedad de *consistencia* de un conjunto de enunciados E engloba aquella de la consistencia de un solo enunciado o expresión en el caso particular del conjunto E si contiene un solo elemento. Un conjunto finito no vacío de enunciados $E = \{a_1, a_2, \ldots, a_n\}$ es consistente, si el enunciado que resulta de la conjunción de sus elementos $a_1 \wedge a_2 \wedge \ldots \wedge a_n$ es *consistente*.

e') *Inconsistencia o contradicción.* La propiedad de consistencia recubre aquella de *no contradicción*; la inconsistencia de una forma de enunciados se debe a su carácter contradictorio: una forma de enunciado es *inconsistente*, *si* no existe ningún valor de verdad de los componentes atómicos de A bajo los cuales A sea verdadera. *Contradicción* e *inconsistencia* son sinónimos, porque no existe, para una forma de enunciado falso, ninguna asignación bajo la cual se obtenga el valor de verdad V.

Anotaciones. Con estas propiedades semánticas fundamentales se tiene el soporte de una *semántica formal*, en donde los términos de (S_0) se asocian a los símbolos de otra teoría intuitiva, presentado en un lenguaje formalizado, sin acordar a estos símbolos otro significado que aquel que reciben los axiomas y las reglas que definen la estructuración o manejo.

La fundamentación de una semántica formal está asegurada si se conoce de manera precisa su estructura sintáctica. De ahí resultan, que si se quieren dar, en materia de semántica, definiciones muy generales, es necesario fijar un marco sintáctico muy amplio, para la mayoría de los sistemas usuales e indicar las reglas de interpretación de un sistema teniendo esta estructura.

Se puede decir también, que la relación de adecuación entre el sistema formal y la teoría intuitiva en la cual se va a interpretar este sistema, es inversamente proporcional al grado de rigurosidad esperada, de manera que la adecuación es mayor si el grado de rigurosidad es menor, y viceversa.

b) O*tras propiedades semánticas.* Se han planteado las propiedades semánticas que dan la base a la *semántica formal*, pero es necesario completar esta base de la semántica con otras propiedades, como la *consecuencia semántica* y la *categórica*, que amplían y solidifican el soporte lógico de la semántica. Igualmente, la relación de la sintaxis y de la semántica de un sistema formal trae consecuencias y condiciones necesarias para el desarrollo de la semántica formal y su funcionamiento, representadas en ciertas propiedades de esta semántica.

a') *Consecuencia semántica.* Es introducida por Tarski (1931), para presentarla de forma general; se define un sistema lógico de primer orden, así:

Sea F un conjunto, finito o infinito, de fórmulas cerradas de un cálculo de primer orden. Una fórmula A de este sistema se llama consecuencia de F si es válida en todo modelo de F[6].

Sin embargo, la *consecuencia semántica* permite estudiar sistemas que formalizan una teoría matemática quedándose en un punto de vista más cercano del matemático, que considera una teoría caracterizada por sus axiomas puramente matemáticos y no formula explícitamente axiomas lógicos. En ese caso, se tratan los axiomas como hipótesis, se examinan las consecuencias (obtenidas por medio de re-

[6] Esta noción no es efectiva, es imposible al término de un número finito de etapas (sin aplicar el tercero excluido a conjuntos infinitos), si A es una consecuencia de F.

glas lógicas) y se determinan las propiedades (tautológica, consistencia, independencia, completitud,…, por ejemplo) construyendo modelos. Estas propiedades no son totalmente independientes de la lógica empleada.

Un sistema de axiomas F se llamará *consistente* en relación con la *consecuencia semántica*, si A y $\neg A$ no son simultáneamente consecuencia de F. Por su parte, la independencia de un axioma B con respecto a otros axiomas que constituyen F, se expresa en el sentido que B no es consecuencia de estos otros axiomas.

Un sistema F será llamado *completo* con relación a la *consecuencia semántica*, si cualquier fórmula A del sistema S aplicada a F, constituye los axiomas no lógicos; o bien, A es válida en todo modelo de S, o bien, A es falsa en todo el modelo de S.

La propiedad semántica llamada *categórica*, la cual presenta la condición de correspondencia entre la teoría intuitiva y el sistema formal es *isomorfa*[7], donde se puede exigir que todos los modelos del sistema formal considerado sean isomorfos, lo que va a depender de la lógica que se dé como instrumento de deducción.

b') *Relación entre propiedades semánticas y sintácticas.* Es evidente que, de manera general, la *completitud semántica* no conlleva la *completitud sintáctica*. En efecto, si el

[7] Dos modelos M y M' de un sistema S son isomorfos, si existe entre sus dominios respectivos un conjunto Φ de correspondencias biunívocas, tales que cualquiera que sea la asignación considerada en M, el valor tomado en M por una expresión A, sea idéntico al valor tomado por A en M', para la asignación aplicando variables y constantes de A sobre las imágenes, para Φ de los elementos sobre los cuales se les aplica la asignación correspondiente en M.

sistema S_1 es completo con relación a un *modelo*, y cuyo dominio es enumerable, se establece que:

• Si un sistema de axiomas *F* es consistente con relación a la *consecuencia*, el sistema aplicado *S*, en donde *F* constituye los axiomas no lógicos, es sintácticamente consistente.

• Si un axioma *A* es *independiente* con respecto a la *consecuencia* de los otros axiomas que constituyen *F*, es también sintácticamente independiente.

• Si *S* es sintácticamente *completo*, *F* es completo con relación a la consecuencia.

• Si *S* es categórico, *F* es completo con respecto a la consecuencia.

Anotaciones. A nivel general, se puede decir que las investigaciones semánticas exigen, en el momento en que se realizan sobre sistemas complejos, medios lógicos más potentes que el estudio puramente sintáctico.

Delante de la dificultad que existe de trazar una frontera neta entre semántica formal y sintaxis, se ha propuesto revisar el concepto de *fuerza* del sistema matemático que se va a interpretar, y plantearla como un criterio para distinguir esa frontera.

3. El metalenguaje

El planteamiento de la metáfora de un *diccionario*, con el fin de realizar la correspondencia entre los símbolos del sistema formal y las expresiones de la teoría intuitiva, utilizado específicamente como una traducción, nos lleva a presentar un sistema formal que puede ser considerado como un lenguaje totalmente artificial que se va a traducir en lenguaje corriente.

Pero, para estos dos sistemas (formal y lenguaje corriente), se tienen dos conceptos de *validez* diferentes: para el lenguaje corriente, toda frase declarativa provista de sentido es verdadera o falsa, mientras que en el sistema formal, la propiedad que puede tener una expresión verdadera consiste en ser demostrable o no.

Es decir, que es necesario verificar si una forma de enunciado o una clase de tales formas, posee las propiedades semánticas expuestas (LEROUX, 1998, pág. 46)[8]. Para lo cual, es necesario contar con una *metapropiedad*, en el metalenguaje del lenguaje, que posibilite esta *validez*. Así, se define para las expresiones verdaderas de un sistema formal *S* una propiedad que se llame *verdad*, anotada verdad-*S* (en sentido de las reglas de la semántica en el metalenguaje), para distinguirla de la verdad atribuida a las frases del lenguaje corriente. La pregunta sería: ¿cuáles son las relaciones con las propiedades sintácticas de las expresiones?, se busca para un sistema *S*, si es necesario y suficiente que una expresión sea demostrable para que ella sea verdadera-*SM*.

Si se plantean reglas de traducción rigurosas, es decir, puramente sintácticas, que permiten obtener de manera unívoca por cada *X* una traducción *x*, podríamos considerar

[8] Ciertos métodos de verificación son de naturaleza semántica. El método de Tabla de Verdad es uno de ellos, pero se vuelve insostenible rápidamente, sabiendo que, por ejemplo, para un enunciado que contenga ocho enunciados atómicos comporta 28 = 256 líneas. Otros métodos semánticos, como el método de formas naturales, el de resolución de formas de enunciado y el método de árboles de consistencia ayudan a resolver esta problemática. Además, existen los métodos sintácticos, axiomáticos, que pueden apoyar a los métodos semánticos; por ejemplo, la congruencia de la noción sintáctica de derivabilidad y la semántica de verdad lógica para el cálculo de conectores.

que la expresión X es verdadera-*SM* si x es una definición adecuada de la verdad-*SM* para el sistema S.

Pero declarar "X" es verdad si "x" tiene sentido, cuando se está en medida de determinar si la condición x se realiza; es decir, cuando se puede establecer que x es verdad (o falsa). La definición de la verdad-*SM* parece así apoyarse sobre un conocimiento anterior de la verdad, en el sentido usual del término, que esta verdad reposa sobre la experiencia, por ejemplo, cuando decimos que una cierta fórmula de la aritmética usual es verdad.

Entonces, una definición de la verdad-*SM*, no presenta solución en la búsqueda de un criterio último de verdad. Es evidente que una definición semántica de la verdad no pretende ser criterio de verdad: decir "*il pleut* es verdad si llueve", no permite saber si llueve de verdad, que sería el caso de la semántica formal; es decir, que es un problema que está fuera de su dominio. Por el contrario, es decir, cuando se está en el dominio de una teoría intuitiva, formalmente construida y por la cual lo verdadero, en sentido usual de la palabra, está bien definido (caso de la aritmética elemental, por ejemplo), es totalmente eficaz.

Sabemos que la concepción intuitiva de lo *verdadero* puede conducir a antinomias, como en la antinomia del *mentiroso* (véase cap. I: "Génesis de los lenguajes formales"); así que es necesario, a partir de la noción usual de lo *verdadero*, presentar un tipo particular que no conlleve antinomias. Sin duda, se podrá objetar que si el dominio de la semántica se reduce a sistemas que formalizan las teorías matemáticas su uso es restringido. Al situarse en la perspectiva de la lógica matemática, la cual no pretende dar una imagen fiel del pensamiento lógico, en su totalidad, pero solo del pensamiento lógico tal como lo emplea el matemático, la definición "*X* es verdad si *x*", no se formula en el lenguaje que constituye el sistema *S* estudiado, pero es una frase del lenguaje usual en el cual nosotros hablamos de *S*. Pertenece, por tanto, al metalenguaje *M*(*S*) que, en el caso presente, es el español.

Por tanto, se requiere una rigurosidad de la semántica que permita evitar las antinomias, como es una *axiomatización*[9], realizando, por ejemplo, un inventario minucioso de los recursos que debe presentar el metalenguaje, tanto en posibilidades de designación como en las de razonamiento.

Así, por ejemplo, en la frase de definición de la verdad-*SM* de un enunciado "*X*" de *S*, intervienen: el nombre *X* de este enunciado, su traducción *x*, la relación lógica "si y solo si" y las exigencias a las cuales deberá satisfacer el metalenguaje. Estas últimas son de tres clases:

1. Permitir hablar de las expresiones de *S*; es decir, designarles por medio de nombres y enunciar sus propiedades sintácticas bajo la forma de relaciones entre nombres de "objetos" formales.

Por ejemplo: "A_i es una tesis"; "A_j se obtiene a partir de A_f por sustitución de la fórmula A_g de la variable x_p", etc. Ella contiene lo que se llama una descripción sintáctica de la estructura de *S* (elementos llamados *structural-descriptions* por Tarski —1931—).

2. Permitir la traducción de cada expresión de *S*. Mientras que el metalenguaje es el lenguaje de comunicación usual, estas traducciones no son otra cosa que las expresiones de

[9] Tarski (1931) va a decirnos, al contrario, que la axiomática no es la panacea universal, pero realiza la formalización casi completa de la semántica.

la teoría intuitiva formalizada por S. Si el metalenguaje se axiomatiza, es necesario escoger un metalenguaje en el cual figuren las expresiones aptas para servir de traducción a las expresiones S.

3. Ser lo bastante amplio para que se pueda expresar y demostrar la existencia de relaciones de S y las traducciones de estas últimas: es necesario disponer, no solo de una lógica en el sentido estricto de la palabra, sino también de una matemática; según el caso, por ejemplo, algunas veces un fragmento de la teoría de conjuntos es suficiente y en otros casos, es una teoría de conjuntos "fuerte" que se debe utilizar. Se verá que de la misma manera que en sintaxis, la *fuerza* de la matemática empleada tiene una importancia considerable.

Por ejemplo, el metalenguaje M puede pertenecer a una lógica análoga al de los *Principia mathematica* de WHITEHEAD y RUSSELL: ella está basada sobre una teoría de tipos y contiene, además, los cálculos elementales de las proposiciones y de predicados de primer orden, una parte del cálculo general de clases y cálculo de relaciones. En este metalenguaje, la relación de inclusión "$\subset$" traduce la relación de inclusión "I" que figura en el alfabeto de S, de manera que una expresión de S como $I\ (x_1, x_2)$ se traduce por $E_1 \subset E_2$, donde E_1 y E_2 están en el metalenguaje $M(S)$ de las variables de clases.

Con estos requerimientos, se puede dar una definición de verdad-*SM*, pero la dificultad viene en el sentido que esta definición debe aplicarse a una infinidad de expresiones; en consecuencia, no es posible dirigir una lista exhaustiva de las expresiones y de las condiciones de verdad de cada una. Sin embargo, si se cuenta con las reglas generales que permiten determinar la verdad de una expresión cualquiera, se puede proceder por recurrencia, con las condiciones de verdad de las expresiones más simples y a partir de ellas, se enunciarían las condiciones para las expresiones compuestas. Definir las condiciones de verdad no tiene sentido para las expresiones cerradas (no contiene variables libres), de manera que en aritmética intuitiva, la expresión "$x < 13$" no es ni verdadera ni falsa (mientras que "$12 < 13$" es verdadera" cualquiera que sea x); sería, entonces, vano buscar una definición permitiendo fijar una vez por todas la verdad o la falsedad de $P^2(x_1, x_2)$. De ahí, la necesidad de definir otra exigencia de la semántica:

4. Establecer si una expresión posee la verdad o falsedad. TARSKI (1931) define, para una expresión, la posibilidad de ser *satisfecha* y agrega la definición de verdad: será llamado *verdad* todo enunciado susceptible de ser satisfecho en todos los casos posibles. La definición de *satisfacción* depende de la estructura del sistema que se estudia, de las expresiones y de la riqueza del metalenguaje en el cual se enuncia la definición. Las investigaciones de TARSKI en su memoria sobre la verdad, están subordinadas a una teoría de categorías semánticas que permite clasificar los diversos sistemas por complejidad creciente; en este marco se quiere definir la verdad para los sistemas de más en más ricos. Se puede enunciar, en el metalenguaje adoptado, las condiciones recurrentes: variables formales aplicadas, por ejemplo, sobre clases de la teoría intuitiva formulada en el metalenguaje; los predicados formales a dos variables aplicados sobre las relaciones binarias entre clases, mientras que los axiomas lógicos en el sentido estricto reciben su interpretación habitual.

De suerte, que decir que una fórmula del sistema formal puede ser satisfecha por una *n-uple* de clases intuitivas, quiere decir, afirmar que es realizable en el sentido que se haya definido.

Anotaciones. Para responder a la pregunta: ¿qué es interpretar un sistema formal?, hemos tenido que plantear una *sintaxis formal*, lo suficientemente axiomatizada y rigurosa que facilite la tarea de la interpretación. Esta sintaxis posibilita el planteamiento de las nociones generales y los métodos requeridos por las propiedades inherentes a la semántica y al desarrollo de la interpretación que le compete, dando la base a una *semántica formal*. Son entonces, las preguntas propias de la semántica (tautologías - tesis - verdad) que facilitan la definición de métodos que responden a esa problemática.

Se posibilita así, la correspondencia de la estructura formal abstracta del sistema formal a la estructura de la teoría intuitiva en cuestión, teniendo como resultado la interpretación del sistema formal en la teoría intuitiva, es decir, la semántica del sistema formal.

Capítulo V

LENGUAJES DE COMPUTACIÓN Y LENGUAJES FORMALES

1. Introducción

La lógica formalizada o lógica matemática tiene dos salidas importantes: la teoría de modelos y el desarrollo de la informática y esta como dominio privilegiado de la aplicación de la lógica, liberándose así del reduccionismo implícito del formalismo. Igualmente, luego de los teoremas de incompletitud de Gödel[1], se realiza un paso entre lo formal a lo mecanizable o calculable, por intermedio de "códigos". Igualmente se develan *a priori(s)* que no habían sido tomados en cuenta en la realización de la axiomática o procedimientos de cálculo. Así, el planteamiento de la mecanización,

[1] Gödel ha probado primero un teorema de completitud y luego de incompletitud. Por completitud, se entiende la adecuación entre una interpretación y un sistema formal. En cuanto a la incompletitud, por ejemplo, se tiene el teorema 1 que concierne la relación entre un sistema formal de aritmética, T, y la interpretación de esta aritmética en los enteros naturales; correspondería a decir que el sistema formal T no demuestra todo aquello que puede legítimamente interpretarse como verdadero, hay una incompletitud. Pero se puede presentar el problema según las reglas requeridas de la lógica pura con respecto a todas las interpretaciones (modelos) posibles y, en ese caso, hay completitud. Contribución muy memorable de Gödel a la Lógica.

por medio de un computador: una máquina "física" propia para hacer *deducciones*, porque razona de manera formal, presenta igualmente la noción de una máquina dinámica, susceptible de cambiar de punto de vista; por ejemplo, si se trabaja sobre un dato tal como "el vuelo 180 para París sale a las 13:30", quiere decir que ese dato desaparecerá de la memoria del computador a las 13:30.

Vemos, entonces, que al lado de las verdades universales de las matemáticas y de la lógica, se encuentran verdades transitorias, no reutilizables.

La lógica matemática, en principio, ha sido pensada en relación exclusiva a las matemáticas y privada de una relación, igualmente importante, con el mundo físico. Así que en un mundo de fenómenos, los principios lógicos no tocan la verdad universal, sino los hechos; por ejemplo, "el vuelo con número 180" que es un hecho de la realidad. En este sentido, se considera el cambio de ciertos principios lógicos instaurados desde siempre; así, por ejemplo, con $ 10.000 se puede comprar un paquete de caramelos, pero con los mismos $ 10.000 se puede comprar un paquete de chocolates, pero no los dos. De manera que el principio $A \rightarrow A \,\&\, A$ de la lógica clásica, es *dinámicamente* falso cuando se toca el mundo físico y, por tanto, a las aplicaciones de la informática: realizar A no es lo mismo que realizar "A y *realizar* A". Así mismo, si en lógica y en matemáticas se utilizan conjuntos de valores, que en su mayoría son numéricos o algebraicos, en informática el conjunto de valores no es necesariamente numérico[2].

[2] Un computador solamente va a manipular "0 y 1" (valores numéricos), pero se llega a estos por vía de traducciones de valores totalmente diferentes a los numéricos.

Por otro lado, el aspecto físico de la máquina va a presentar ciertos requerimientos como la necesidad de "estructuras de información" (listas, pilas, tablas, vectores, árboles,...) para la organización de la información en las memorias de la máquina y el conjunto de valores finitos, registrados en los soportes físicos propios de la máquina. Así mismo, es necesario distinguir la manera de acceder a esas estructuras (secuenciales o directas), características estas, solamente relativas a la parte física de la máquina, pero que es necesario considerarlas en el procedimiento o algoritmo.

Uno de los resultados de esta formalización conforme a la parte lógica y física de una máquina son los "lenguajes de computación" que se emplean tanto para la definición y estructuración de los esquemas de aplicación (algoritmos), como en la codificación de estos (programación) o en la ejecución del programa como tal (compiladores, interpretadores, ensambladores o de máquina). La información que recibe un computador debe darse en forma exacta, por cadenas de símbolos, sin ninguna ambigüedad y sin dejar nada a la imaginación. La construcción de cadenas de símbolos utilizadas con este fin debe someterse a reglas sintácticas estrictas, del género de aquellas solicitadas por HILBERT (2002) en su teoría de la demostración. Así mismo, es la efectividad de procedimientos calculables requerida en una axiomática, que posibilita el desarrollo de los lenguajes de computación.

2. PROCEDIMIENTOS EFECTIVAMENTE CALCULABLES

El pasaje de lo formal (lógica) a lo mecanizable o calculable (computador) tiene como base la *efectividad* o *pro-*

cedimiento efectivamente calculable. Este procedimiento "mecánico" llamado *algoritmo* o *programa*, ha sido formalizado siguiendo un método de lenguaje formalizado pasando de una noción informal a un concepto matemático (STERN, 1990, págs. 81-83).

A) *Propiedades de los procedimientos efectivamente calculables*

Estos procedimientos requieren, como en lógica, de tres propiedades síntesis: la decidibilidad, completitud y consistencia.

a*) Decidibilidad.* La decidibilidad, tanto en lógica como en informática, se define por la búsqueda de un procedimiento efectivo que determine con anterioridad un resultado o la "parada" del procedimiento. Con base en esta condición, se requiere la estructuración y formalización completa de cada uno de los elementos del procedimiento, tales como el lenguaje, sus reglas de formación y transformación, los códigos de la información dada, las funciones para la realización de cálculos y la definición de estructuras de información que presentan las configuraciones de las expresiones y, en general, del procedimiento mismo.

La propiedad de *decidibilidad* permite resolver los problemas de tipo "sí o no". Por ejemplo, en el contexto de una *gramática algebraica* (Z, A, P, z_0), nos podemos preguntar ¿si el lenguaje engendrado $L\ (z_0)$ es vacío? Las respuestas a esta pregunta admiten un "sí o un no". ¿Por tanto, la *decidibilidad* admite un solo tipo de problema que es de la forma: $x \in A$?

b*) Completitud y consistencia en informática.* La utilización de tablas de verdad, en lógica, permite sistemáticamente verificar que una expresión *F* o un conjunto finito de expresiones Σ, son contradictorias. El inconveniente de este método es su costo: la tabla comporta 2^n entradas, en la cual n representa el número de variables que tienen una ocurrencia en F o Σ. Este método no se ajusta completamente a las prácticas deductivas en informática, donde de *F* o Σ se deducen consecuencias sucesivas según reglas conocidas, hasta obtener una expresión o enunciado contradictorio $p \wedge \neg p$. En este último caso, un sistema de reglas es suficiente para probar la contradicción de los conjuntos de fórmulas o enunciados que son contradictorios y el sistema en ese caso es *completo*.

Así, la propiedad de completitud, en los procedimientos efectivamente calculables, está representada por el conjunto de reglas que asegura la *consistencia* de ciertas fórmulas o enunciados y señala la contradicción de otras fórmulas o enunciados, que pertenecen al mismo sistema.

c*) Ejemplos de procedimientos efectivamente calculables y sus propiedades.* Uno de estos procedimientos, que presenta el conjunto de reglas requeridas, se denomina la *regla de resolución*, aplicado en informática a partir de ciertos objetos informáticos como son los árboles.

d*) La regla de resolución*[3]. Con el fin de establecer la aplicabilidad del método de resolución a enunciados com-

[3] Podremos referirnos a KLEENE (1952), con el fin de familiarizarnos con otros sistemas de pruebas; por ejemplo, el cálculo de secuencias de GENTZEN. El método de resolución se debe a J. A. ROBINSON, "A machine oriented logic based on the resolution principle", J. ACM 12, 1, 23, 41 (1965), obra citada por JACQUES STERN, (1990, págs. 166 y 311).

puestos, nos vamos a referir a "cláusulas"[4], por medio de las cuales podemos definir una aplicación sencilla, que asocia a cada enunciado F un conjunto de cláusulas C (F) de longitud < = 3, para llegar a F es contradictorio si y solo si C (F) lo es. Se presenta C como un conjunto de cláusulas y se llama "prueba por resolución" a partir de C, una cadena de cláusulas tal, que toda cláusula de la cadena sea un elemento de C o sea obtenido como una resolución de dos cláusulas de la cadena. El último elemento de la cadena es el objetivo o conclusión de la prueba. Si este objetivo es la cláusula vacía, la prueba se llama "refutación de C"[5].

Por consiguiente, toda cláusula que aparece en una prueba por resolución a partir de *C* es una consecuencia de *C*, mientras que un conjunto de las cláusulas a partir de *C* que admite una refutación es contradictorio. Así, se determina la consistencia del sistema y la unidad de las reglas que admiten la prueba de resolución y determina su completitud.

Ilustramos la prueba de resolución, a partir de un árbol de resolución (Stern, 1990, pág. 159).

[4] "Cláusula" es toda disyunción (relación por "o") de literales (la palabra 'literal' designa una variable proposicional o la negación de tal variable). La longitud de una cláusula es el número de literales sobre la cual se realiza la disyunción.

[5] Jacques Stern (1990), anota que en el conjunto de las variables de C a n elementos, hay máximo 22^n cláusulas, lo que da un límite sobre el número de etapas del algoritmo. Este límite es gigantesco y el método es ineficaz, de manera que, por ejemplo, las cinco cláusulas iniciales producen ocho resoluciones, en donde no hay ninguna vacía. Con el fin de limitar la explosión combinatoria, se puede restringir el uso de la regla de resolución, como la formación de las resoluciones a partir de parejas c1, c2, en donde ninguno contiene a la vez un literal u y su opuesto no-u. Este método se llama "resolución restringida".

e*) Árboles de resolución (véase cap.* V, sec. 2, apdo. B: "Estructuras de información"). Sea C un conjunto de cláusulas. Un árbol R es llamado de resolución a partir de C y dispone de las propiedades siguientes:

a) Las hojas de R son etiquetadas por los elementos de C;

b) Cualquier nodo no terminal comprende exactamente dos sucesores y su etiqueta es una resolución de los dos sucesores.

Dado un tal árbol R, que puede aceptar una prueba por *resolución* a partir de C, cuyo objetivo es la etiqueta de la raíz R: para esto es necesario organizar las etiquetas de R respetando el nivel de los nodos, es decir, la altura del subárbol correspondiente.

Inversamente, a partir de una prueba de *resolución R* de C, sea: $c_1, \ldots, c_n$, es posible construir, de manera recursiva, un árbol de resolución cuya raíz tiene como etiqueta c_n, así:

• si c_n pertenece a C, el árbol buscado se reduce a la raíz;

• si c_n es la resolución de c_i y c_j, $i < n$, $j < n$, el árbol buscado admite como subárbol izquierdo correspondiente a c_i y como subárbol derecho un árbol correspondiente a c_j.

Si la raíz de un árbol de resolución a partir de un conjunto de cláusulas C se etiqueta por la cláusula vacía, se dice que este árbol rechaza C.

Anotaciones. La práctica de la informática, en posesión de soportes como los procedimientos efectivamente calculables, con sus propiedades características, asegura una verdadera estructuración tanto del cálculo, como de la información y del lenguaje, requeridos por un algoritmo o programa.

Por consiguiente, es importante presentar las estructuras de información que dan la unificación y configuran un algoritmo o programa, los diferentes tipos de objetos informáticos y el lenguaje formalizado que expresa y relaciona los distintos elementos de la estructura de información determinada, por palabras o números enteros sobre un alfabeto finito.

B) *Estructuras de información*

Las *estructuras de información* son el resultado del método de construcción de los procedimientos o algoritmos, a partir de *esquemas elementales*, como listas lineales, pilas, tablas, árboles, procedimientos recursivos y que configuran los *objetos informáticos*.

A manera de ilustración, se presentan los casos de estructura de información referentes a árboles, *pilas*, *procedimientos recursivos*, así:

a*) Árboles.* Se define un "árbol" como un conjunto finito parcialmente ordenado y que posee las propiedades siguientes:

• existe un elemento más pequeño que todos los otros, llamado "raíz del árbol";

• los elementos inferiores de cualquier elemento forman un conjunto totalmente ordenado;

• los elementos de un árbol son llamados "nodos" (cimas o puntos). Los sucesores o hijos de un nodo "*a*" son los nodos "*b*" más grandes que "*a*", tales que ninguna cima no se encuentra entre los "*a*" y "*b*";

• los nodos que no tienen sucesor se llaman *terminales* u hojas del árbol. Los otros nodos se llaman *internos*;

• si se le da una relación de orden a cada nodo con el conjunto de sus sucesores, se obtiene un árbol ordenado;

• el *grado* es el número máximo de sucesores de un nodo del árbol, será binario si cada elemento tiene máximo dos sucesores.

La utilización de estos árboles puede formalizarse a partir de términos aritméticos. Se consideran árboles *etiquetados*, porque cada nodo tiene una *etiqueta*, como elemento de un alfabeto $\sum$. Para un alfabeto dado, $\sum$ tal que $\sum = \{x, y, z, +, *\}$, se define un término aritmético T con variables x, y, z de un árbol ordenado y etiquetado, con las propiedades siguientes:

• las hojas de T son etiquetadas por uno de los símbolos x, y o z;

• los otros nodos de T (si existe) son etiquetados por uno de los símbolos + o * y tienen cada uno, dos sucesores.

Así mismo, es posible sustituir esta definición por otra que le es equivalente. Cada árbol T de variables x, y, z, se llama *término aritmético*, cuando se tiene una de las propiedades siguientes:

i) T se reduce a una hoja etiquetada x, y o z;

ii) la raíz de T se etiqueta por uno u otro de los símbolos + o *; la raíz tiene dos sucesores y cada uno de los subárboles de izquierda o derecha es un término aritmético de las variables x, y, z.

En la (figura 1, pág 116) se observa una representación gráfica de este árbol: cada nodo se representa por un círculo con una etiqueta; el orden del árbol se lee de arriba hacia abajo.

Esta definición es *recursiva*, en el sentido que para verificar que ella se aplica a un árbol T, es indispensable aplicarla a dos *subárboles* (y así sucesivamente). Con el fin de evitar que

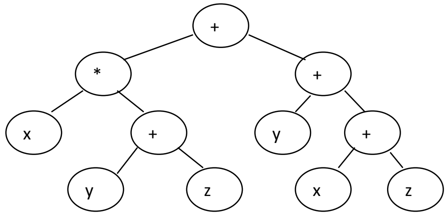

Figura 1. Término aritmético

alguna circularidad resulte, se puede introducir la altura del árbol (como límite), que es el número máximo de elementos de una parte totalmente ordenada (o rama). Se constata que la cláusula i) trata los casos de los árboles de altura 1 y que la cláusula ii) define el conjunto de los términos aritméticos de altura $< = n$ a partir de los de altura $< n$.

La estructura de *árbol* puede ser presentada por un esquema elemental (procedimiento), en lenguaje algorítmico, donde la representación del término aritmético (*p*) utiliza un registro que comprende una 'etiqueta' y dos apuntadores de los dos nodos sucesores (*p*↑. *derecha*) (*p*↑. *izquierda*):

Procedimiento *Escribir* (*p*: término)
Principio
    si *p* < > entonces
    Principio
        si ((*p*↑. *etiqueta* = '+') **o** (*p*↑. *etiqueta* = '*'))
        entonces escribir ('(');
        escribir (*p*↑. *etiqueta*);
        escribir (*p*↑. *izquierda*);
        escribir (*p*↑. *derecha*);
        si ((*p*↑. *etiqueta* = '+' o (*p*↑. *etiqueta* = '*'))
        entonces escribir (')').
    fin
fin

Si se le afectan valores numéricos a las variables x, y, z, se pueden extender estos valores a todos los nodos del árbol y obtener así, a nivel de la raíz, el valor del término aritmético.

b*) Procedimientos recursivos.* Estos procedimientos utilizan la *autorreferencia* porque deducen el tratamiento de un dato o de un conjunto de datos, del mismo tratamiento aplicado a un dato más pequeño o a un conjunto de datos menos numeroso hasta obtener un caso evidente. Este procedimiento es completamente diferente de un procedimiento que se llama a "sí mismo" de manera total, cíclicamente sin "parada"; es entonces necesario un "nivel superior" u "monitor" con el propósito de que el procedimiento pueda tener fin.

Se puede ilustrar la noción de este esquema recursivo, así: calcular la suma de enteros de 1 a N, $N = 3$. Una de las formas para realizar la suma es el de adicionar a N la suma de los enteros de 1 a N-1. Con el fin de calcular tal suma agregamos N-1 a la suma de N-2, etc. La "parada" del procedimiento (forma adoptada para salir de la recursividad) se representa por la condición, suma de enteros: $N = 1$. Así, disponemos de la definición llamada *recursiva*, porque se utiliza ella misma: "La suma de los enteros de 1 a N (para $N >= 1$) es igual a 1 si $N = 1$, sino adicionar a N la suma de enteros N-1".

Este enunciado puede traducirse por el procedimiento *Suma*, en lenguaje algorítmico, así:

Procedimiento *Suma*: número entero (N = número entero)

Principio

Si $N < 2$ *Entonces* Resultado 1 *sino* Resultado N + Suma (N-1)

{*si* $N >= 2$, la función no podrá llegar a su resultado *sino* cuando ella calcule (N-1)}

Fin

El funcionamiento del cálculo de Suma (3) para *N* = 3 es el siguiente:

a) Principio de Suma (3)
  el resultado será 3 + Suma (*N*-1 = 2)
  llamado de Suma (2)
    b) Principio de Suma (2)
      el resultado será 2 + Suma (*N*-1 = 1)
      llamado de Suma (1)
        c) Principio de Suma (1)
          el resultado es 1
    b) el resultado de suma (2) es 2 + 1 = 3
a) el resultado de Suma (3) es 3 + 3 = 6

Al final del cálculo de *Suma* (3), vemos que en el momento de calcular suma (1), tres ejemplares del mismo procedimiento interactuaron. Los diferentes valores de la variable de control de la "boucle" se generaron automáticamente por la llamada a *Suma* (*N*-1) para cada valor de *N*. El desarrollo del procedimiento *Suma* equivale, entonces, a reemplazar esta variable de control (*N*) por *N*-1, lo que conduce a un determinado resultado con el fin de evitar la recursividad cuando *N* = 1, lo que permite no caer en la regresión al infinito y llegar al fin del procedimiento y la "parada" del programa.

c*) Pilas.* Igualmente, se puede examinar un esquema recursivo en los "autómatas de pila", utilizado particularmente en los problemas relativos en la sintaxis del lenguaje[6]. Este esquema utiliza términos precisos que representan los diferentes "estados" del procedimiento:

[6] Los "autómatas de pila" fueron introducidos hacia 1960 por el lenguaje de programación IPL, utilizado en inteligencia artificial.

• "empilar": suspender las operaciones de la tarea en realización, sin olvidar su propia localización, con el fin de abordar una nueva tarea. Generalmente, esta nueva tarea se considera de nivel inferior, con respecto a la precedente.

• "depilar": el inverso del anterior, es decir, abandonar las operaciones situadas a un nivel dado con el fin de retomar las operaciones a un nivel superior, en el punto en el cual se habían interrumpido.

Una pila es, por tanto, una tabla que indica (1) donde se estaba en el desarrollo de las tareas interrumpidas (dirección para el retorno) y (2) los elementos para conocer los puntos de interrupción (valores dados a las variables). De esta manera, al retomar la ejecución de una tarea, la pila indica la dirección a la cual nos debemos dirigir.

Así, las funciones de "empilar" y "depilar" que se realizan recursivamente, tienen una salida de la recursión correspondiente al final de la pila[7]. Estos procedimientos son igualmente utilizados en el tratamiento de lenguajes naturales u ordinarios.

Se puede notar que la *recursividad* es un principio, en donde un mismo "fenómeno" se produce en diferentes niveles, pero los "hechos" encontrados en los diferentes niveles no son idénticos. Así, el procedimiento recursivo se basa en la selección de procesos semejantes que permiten dividir una tarea dada en subtareas.

En el ejemplo del procedimiento "Suma", se indicaba la existencia de una "boucle" que ordena ejecutar un conjunto

[7] Los términos "pila", "empilar", "depilar" se inspiraron en los platos empilados de las grandes cocinas, donde generalmente, debajo de la pila tiene un resorte que sirve para mantener el plato superior y tiene una altura más o menos constante.

determinado de operaciones, para luego volver al punto de partida y "reejecutar" las operaciones y así consecutivamente hasta que la condición dada se realice por completo. Aquí, el cuerpo de la "boucle" o el conjunto de instrucciones que se repiten no está fijado completamente. Para que el procedimiento llegue a su fin, las instrucciones de la "boucle" sufren ciertas variaciones previstas (en el ejemplo, el número de iteraciones va a depender de la variable N). En conclusión, la "boucle" es semejante pero no idéntica a las otras dentro del mismo procedimiento. Una "boucle fija" no permitiría cambios o la dinamización del procedimiento y, por consecuencia, no podría llegar a un resultado.

Por esa razón, la utilización de las "boucles", en el procedimiento, se puede resumir en una cadena de operaciones semejantes ejecutadas de forma iterativa y cuyo procedimiento "tiene un fin" cuando las condiciones específicas se realizan. Dos casos se presentan: a) el número máximo de iteraciones se conoce con anterioridad, "boucle limitada"; b) el programa se ejecuta y el procedimiento para, "boucle libre".

Anotaciones. En los esquemas de las estructuras de información presentadas, se pueden definir varios elementos característicos:

• Acciones que definen la composición del esquema o algoritmo. Estas acciones son definidas según:

— una secuencia, que es finita (en los ejemplos se cierra la secuencia con el símbolo ";");

— la secuencia está realizada según una condición (**si** *condición* **entonces** *secuencia de acciones* **fin-si**);

—la secuencia puede ser iterativa, donde la acción se repite, pero por formalización se escribe una sola vez y puede tomar el nombre de "boucle", y si es un procedimiento completo, es una "rutina", "bloque"…; en los ejemplos de esquema, tenemos: "Suma".

• Valores, son de varios tipos:

— "simples", como los numéricos (1, 7, –5,…); caracteres lexicográficos (*a*, *b*, *c*,…); booleanos (0, 1);

— "estructurados", son valores compuestos, en donde en un conjunto finito de valores, cada elemento es accesible por un identificador. En el esquema del árbol, el identificador "etiqueta" puede ser "+" o "*".

• Operadores, sobre el conjunto de valores, pueden ser lógicos o numéricos, que son los mismos utilizados en lógica y matemática.

• Las expresiones del lenguaje que definen y representan los elementos del esquema expuestos.

Se señala, además, la importancia del lenguaje en general para la descripción de estos esquemas, que serán traducidos o interpretados en los diferentes lenguajes de programación, interpretadores, compiladores, ensambladores y de máquina. Es decir, que el esquema, el algoritmo o el programa solo pueden ser realizados por la máquina si pasan por las expresiones de un lenguaje formalizado particularmente para las aplicaciones de la computación.

3. El lenguaje computacional

De acuerdo con los requisitos de un *procedimiento efectivo calculable*, para su aplicación es necesario "vehicular la información" (cálculos y datos) por medio de la descripción y representación de expresiones precisas y rigurosas, que como en los lenguajes formales, ningún lenguaje ordinario podría realizarlas[8]. Un lenguaje de computación es, entonces, un lenguaje formal, artificial, que asegura la representación, simbolización y normalización de las expresiones de cálculo, propias de un computador y que son controladas por medio de reglas explícitas.

En estos lenguajes computacionales, una expresión se representa por un *código alfanumérico*. Representar una entidad, en estos códigos, tiene como fin realizar operaciones, que sería muy difícil efectuar sobre la expresión misma; se trata de facilitar el proceso de operación o manipulación de las expresiones, dejando de lado algunas propiedades de la expresión representada. De esta manera, es posible realizar operaciones sobre las entidades representadas, operaciones que hubieran sido irrealizables sobre la entidad misma.

Igualmente, estos códigos representan las expresiones o información que se refieren tanto al mundo físico o de los hechos (datos), como a las aplicaciones de cálculo, lo que indica una correspondencia estrecha entre el lenguaje y la realidad, así que este tipo de lenguajes es *relacional*.

[8] DANIEL LAURIER, (1993, págs. 8-9), anota que se presentan dos actitudes en el momento de relacionar el lenguaje y el conocimiento: los de actitud negativa, el lenguaje es un medio adecuado de expresión, y los de actitud positiva consideran el lenguaje no adecuado para esta relación. Estos últimos consideran que el lenguaje es una fuente de error (los místicos, LAO-TSÉ, PLOTIN, BERGSON); los reformadores (LEIBNIZ, FREGE) que proponen la construcción de lenguajes artificiales con reglas explícitas, y los terapeutas (WITTGENSTEIN), cuyos problemas resultan de un mal funcionamiento del lenguaje. Estos dos últimos proponen construir lenguajes artificiales controlados por reglas explícitas que previenen los errores del lenguaje.

Sin embargo, a nivel general, esta *información* para que sea utilizable por un computador, debe constituir un conjunto finito de números binarios, que representan dos tipos de información: la información "dada" (correspondiente al "hecho real") y la información programa (correspondiente a las instrucciones del algoritmo o programa). Estas dos informaciones deben ser colocadas en la memoria central del computador y codificadas según tipo de información. Así, por ejemplo, para el cálculo de un polinomio $P(x)$ para un valor dado de la variable x, se dispone de estas dos clases de información:

• "Información dada", que comprende los coeficientes y el grado del polinomio, lo mismo que el valor de la variable, generalmente bajo la forma de números decimales.

• "Información programa", que comprende el esquema o estructura de información y el algoritmo o programa resultante.

El concepto de *lenguaje computacional* implica una *realidad* (información de hechos o demostraciones) representada y transformada por una máquina (información programa).

Así, se pueden ver ciertas relaciones entre las máquinas, el lenguaje y el mundo:

1. El lenguaje relacionado a las máquinas: cuya comunicación se realiza a través de las expresiones bien formadas, convenientemente codificadas en un lenguaje de computación y la máquina misma. Este es el objeto de la sintaxis del lenguaje.

2. El lenguaje relacionado con el "mundo real": la comunicación se realiza a partir del conocimiento de las condiciones y características de la realidad de los hechos y las expresiones del lenguaje. Este es el objeto de la semántica del lenguaje.

La relación lenguaje-máquinas, hace referencia al proceso realizado por un programador en la codificación de expresiones: primero en un *lenguaje* algorítmico y luego en su traducción a un lenguaje de programación y a los otros lenguajes compiladores, interpretadores, ensambladores y de máquina. Expresiones estructuradas en un procedimiento efectivamente calculable o algoritmo.

La relación lenguaje-mundo, hace referencia al proceso tanto de percepción y comprensión de la realidad (hecho, demostración), como la de producción de resultados, previstos en el algoritmo y que deben corresponder a las interpretaciones necesarias de los hechos del *mundo*.

A) *Estructuración del lenguaje computacional*

A partir de estos dos tipos de relaciones y tomando la base de un *lenguaje formal*, se presenta la estructura de este lenguaje como un sistema unificado por una *gramática*, a partir de la cual se desarrollan los conceptos y funcionamiento de sus dos elementos: la sintaxis y la semántica.

La gramática de un lenguaje de computación permite describir en su totalidad, por medio de un número restringido de reglas (formación y transformación) y de principios generales, las formaciones y relaciones sistemáticas y estructurales de los diferentes símbolos y expresiones del lenguaje.

Lo cual nos sugiere que estas reglas permiten determinar para el lenguaje:

- las cadenas de expresiones que son "expresiones correctas" del lenguaje;
- las cadenas de expresiones que son significantes de ese lenguaje;

• el significado de una expresión cualquiera de ese lenguaje.

En consecuencia, lo mismo que para los lenguajes formales, se puede hablar de dos niveles de la gramática: la sintaxis y la semántica. La *sintaxis* explicita las reglas que permiten determinar las cadenas de palabras que constituyen las expresiones correctas o expresiones bien formadas. La *semántica* explicita las reglas que permiten determinar, por una parte, las cadenas de palabras que tienen un significado y, por parte, el significado de cada frase o expresión del lenguaje.

En este sentido, la gramática utilizada se genera a partir de expresiones desarrolladas mediante símbolos o secuencias de símbolos que obedecen a reglas de formación estructural, tanto para la sintaxis como para la semántica.

B) *Reglas de los lenguajes computacionales*

Los lenguajes computacionales son lenguajes regulados, es decir, controlados por un sistema de reglas; de manera que la utilización de un símbolo sea determinado por ciertas reglas de formación y de transformación. Las características de estas reglas son las siguientes:

• constitutivas: están enfocadas para fines exclusivos y prácticos en la descripción de un algoritmo o programa;

• definidas: resumen la afirmación precisa de una acción en una palabra o expresión; por ejemplo, una instrucción como “mientras que”[9], afirma una *función iterativa* en el modelo algorítmico y no otra función;

[9] Lenguaje Algol W.

• explícitas: toda regla debe ser precisa, clara y sin ambigüedad, es decir, totalmente formalizada.

El lenguaje computacional se caracteriza por *reglas constitutivas*, con relación a las *reglas normativas*, intrínsecamente ligadas a las acciones o comportamientos que ellas regulan, portan sobre una actividad preexistente de manera que se pueda describir su comportamiento fuera de la existencia de esta regla; a la inversa, una *regla constitutiva* permite describir la acción o comportamiento que ella controla. En este caso, se tienen las expresiones o símbolos, por un lado, y las reglas, por el otro, y la acción se realiza únicamente a partir de las reglas. Es así como la existencia de un lenguaje computacional presupone siempre una "explicación" sobre este lenguaje y la forma de construcción de las expresiones a partir de reglas.

Esta concepción, así como en los lenguajes lógicos, nos lleva a la noción de "metalenguaje", en el cual es indispensable que se definan los símbolos, las reglas de formación, de transformación de las expresiones y las reglas semánticas que regulan el lenguaje computacional. Pero el hecho de presuponer otro lenguaje para describir otro lenguaje, presenta un problema de *regresión al infinito* o *circularidad*, se llama a otro lenguaje M para explicar, pero luego se llama a otro lenguaje para explicar $M(L)$, y así consecutivamente. De manera específica para los lenguajes computacionales, las gramáticas realizarían las funciones del "metalenguaje", las cuales describen y definen las diferentes reglas y los procedimientos que aseguran su efectividad, a partir del lenguaje mismo.

Igualmente, para asegurar estas funciones, las reglas deben ser rigurosamente definidas, explícitas, definidas por la gramática y efectivas por medio de procedimientos correspondientes, sin tener que recurrir a otro lenguaje para explicarlas, dando solución a los problemas de circularidad.

La estructuración de estos lenguajes en un conjunto de símbolos, por un lado, y en un conjunto de reglas, por el otro, es su base teórica esencial, porque el conjunto de símbolos puede ser el objeto de estudios lógico-matemáticos, independiente de cualquier contenido.

4. Sintaxis de los lenguajes computacionales

La característica de comunicación de un lenguaje computacional está asegurada por la base de una sintaxis que permite la formación de "expresiones bien formadas" o correctas, fuera de toda referencia a una interpretación. Es necesario estructurar los símbolos del lenguaje y definir las reglas correspondientes.

Así mismo, la *sintaxis* permite la clasificación de las expresiones del lenguaje en términos primitivos (simples) y derivados (compuestos), que pueden tomar el nombre de *categorías* y, por lo cual, la gramática de estos lenguajes se define como categorial[10]; así que a partir de la combinación regulada de los términos primitivos se producen los derivados.

La *sintaxis del lenguaje* computacional tiene como soporte un *alfabeto* que es el conjunto finito de símbolos de base bien definidos; se compone de letras, números, operadores, signos de puntuación,… Sobre estos símbolos y con la ayuda

[10] Las gramáticas de los lenguajes naturales u ordinarios utilizan las categorías gramaticales: nombres comunes, verbos, adjetivos, preposiciones, etc.

de reglas sintácticas se construyen las expresiones que, a su vez, constituyen los algoritmos o programas en el lenguaje de computación. Los símbolos de este alfabeto estructurado corresponden a las dos clases de *información* anotadas: los datos o información de un hecho real o demostración y las expresiones correspondientes al lenguaje mismo.

A) *Estructuración de la sintaxis*

Para ilustrar esta estructuración, se toma como base un lenguaje de programación, de fácil descripción, el Algol W. Lenguaje compuesto de unidades de base, llamadas "palabras reservadas", cuyo conjunto integra lo que llamamos "alfabeto". La sintaxis cuenta con un conjunto de reglas de formación, transformación y funcionamiento de las *expresiones bien formadas*.

La descripción de las reglas y definiciones de la *sintaxis* utiliza una *simbolización rigurosa* y *formalismo* bastante generalizado; así, por ejemplo, signos como < > (designa el término que se va a definir), ::= (designa la identificación), / (separa las diferentes posibilidades de definiciones del término).

Así mismo, para relacionar los términos singulares, pertenecientes a categorías compuestas, se designa el sistema de reglas sintácticas, por ejemplo:

<*instrucción ir-a*> ::= ir-a <*identificador de etiqueta*>

Aquí se dispone de dos términos relacionados: "ir-a" e "identificador de etiqueta" que en su conjunto y en el orden propuesto determinan la regla de utilización y la definición de la instrucción presentada. Es decir, que la instrucción "ir-a", se presenta sintácticamente con el nombre-acción "ir-a", seguido de un número de etiqueta que designa la dirección de memoria donde está almacenado un dato.

Para definir una lista de elementos de la misma naturaleza, los elementos se designan por <*elemento*>; la manera de expresar formalmente esta definición sintáctica es la siguiente:

<*lista de elementos*> ::= <*elemento*> / <*lista de elementos*>, <*elemento*>

Donde la *barra* separa las diferentes posibilidades de definición para la noción que se quiere definir.

La aplicación de esta regla, requiere de una segunda regla que define los diferentes elementos posibles. Muestra, a la vez, cómo construir una lista y cómo reconocer si se tiene una lista sintácticamente correcta:

<*elemento*> ::= *A*/*B*/*C*

De acuerdo con esta regla, se definen como *listas correctas*:

A

A, *B*

A, *A*, *A*

C, *B*, *B*, *A*

Los caracteres tipográficos < > ::= / no forman parte del lenguaje, pero hacen posible su descripción formal.

B) *Descripción de la sintaxis*

Teniendo como base las definiciones expuestas, se presentan los símbolos de la sintaxis, su estructuración, las reglas de formación y las instrucciones.

- *Alfabeto: símbolos de base*

  <símbolo de base> ::= A/B/C/D/E/F/G/H/…
  0/1/2/3/4/…
  verdadero/falso/nulo/…
  cadena/bits/lógica/entero/…
  =/-/…
  empiezo/fin/si/entonces/sino/ira/…
- *Identificadores*

  <letra> ::= A/B/…/Z
  <cifra> ::= 0/1/2/3/4/5…/9
  <identificador> ::= <letra> / <identificador> <letra> / <identificador> <cifra>
- *Tipos de objetos manipulables*

— *Números*

<número complejo> ::= <número real> / <número entero>

<número real> ::= <número decimal> / <número entero> <factor>

<número decimal> ::= <número entero>. <número entero> /. <número entero> / <número entero>

<número entero> ::= <cifra> / <número entero> <cifra>

<signo> ::= +/-

— *Valores lógicos*

<valor lógico> ::= verdadero/falso

— *Valores binarios*

<valor binario> ::= # <cifra hexadecimal> / <valor binario> <cifra hexadecimal>

<cifra hexadecimal> ::= 0/1/2/3/…/A/B/C/…

— *Declaraciones*

<declaración> ::= <declaración variable simple> / <declaración de tabla> / <declaración de procedimiento>

— *Expresiones*

<expresión> ::= <expresión simple> / <expresión condicional> / <expresión de selección>

— *Variables*

<variable de tipo T> ::= <variable de tipo T> (*T* representa un tipo cualquiera de variable)

— *Instrucciones*

<instrucción> ::= <instrucción simple> / <instrucción iterativa> / <instrucción de selección> / <instrucción condicional>

Anotaciones. Los lenguajes computacionales son *lenguajes formales*, entre los cuales se presentan similitudes y diferencias, con respecto a los símbolos. Más precisamente, los lenguajes formales de primer orden se identifican a los lenguajes computacionales, según las variables cuyos valores podrían ser asignados, teniendo la capacidad de ser predicados o funciones. Igualmente, las operaciones de base realizadas para la asignación de valores a las variables son similares, en los dos tipos de lenguajes. Sin embargo, las diferencias residen, en el caso de los lenguajes computacionales, porque estos tienen la necesidad de organizar las expresiones del lenguaje dentro de una estructura de información como son las rutinas, subrutinas, bloques,...

a) Estas estructuras son entidades complejas, pero concebidas de forma esquemática, de talla reducida, en particular para resolver un problema, suficientemente probadas y en lo posible reutilizables en otros programas. Tienden, entonces, a ser creadas con componentes jerárquicos, para su pertinencia y facilidad. Este tipo de estructura no existe en el lenguaje formal.

b) Los lenguajes computacionales utilizan variables de diferentes tipos (enteros, reales, objetos,…) y, en consecuencia, los valores que toman las variables son de tipos que les corresponden igualmente; por el contrario, los lenguajes formales no utilizan estas variables de tipos. La razón de esta característica es práctica, porque muchos de los errores de programas se detectan automáticamente, por medio de pruebas según tipos de entidades utilizadas; de esta manera, los datos serán estructurados y almacenados según ese tipo.

Estas características propias de los lenguajes computacionales, tanto la estructuración por tipos como las "estructuras reducidas", producen "efectos secundarios" en el funcionamiento de los programas como son la entrada y salida de datos, lo que agrega complicaciones a las tareas ulteriores de la sintaxis y semántica de los lenguajes computacionales.

C) *Aspectos formales de la sintaxis*

La sintaxis de un lenguaje computacional, basada en una *sintaxis formal*, ha seguido una evolución en su *formalización*, a partir de los lineamientos teóricos de NOAM CHOMSKY, que aporta la precisión, concreción, estructuración y formalización de la *sintaxis de un lenguaje natural*, incluyendo la noción de *gramática*. Así, el resultado de CHOMSKY y MILLER de 1959 (CHOMSKY y MILLER, 1958, págs. 91-112), enuncia que el lenguaje aceptado por un *autómata* se engendra por una gramática de tipo 3, según la jerarquía de CHOMSKY.

En 1957, el objetivo de CHOMSKY (1969) es construir una teoría sobre las estructuras del lenguaje, mediante una gramática cuyo núcleo es la sintaxis, por su nivel autónomo con respecto a la semántica. La *gram*ática generativa permite la capacidad constructiva del lenguaje: crea expresiones nuevas a partir de expresiones existentes. Así que por medio de un modelo sistemático, riguroso y operativo de la gramática sintáctica, se engendran y describen las cadenas gramaticalmente correctas; su origen es CARNAP (1937), en su libro *Sintaxis l*ógica *del lenguaje*[11].

CHOMSKY define una gramática formal, estructurada según categorías sintácticas; retoma de la lógica los medios de formalizar los métodos lingüísticos y dispone de un *método mecánico* para determinar si una expresión es un enunciado del lenguaje y si cómo este enunciado se articula en constituyentes. Así, si una expresión se representa por una anotación "*quasi*-aritmética" (se aplica también a las palabras), es posible verificarla mecánicamente, con la aplicación de reglas de derivación. Lo cual tiene una semejanza a los algoritmos utilizados por los lógicos para los lenguajes formales. Esta gramática "categorial" es la primera formulación matemática rigurosa de una gramática y se presenta como un conjunto organizado coherentemente, cuyas relaciones con los autómatas se ha formalizado.

[11] CARNAP plantea la "sustitución parcial". Si antes, los lógicos plantearon el problema de las categorías sintácticas, no considerando sino la sustitución total de las palabras (es decir, en la totalidad de los enunciados), desde CARNAP se dispone de teoremas sobre las categorías sintácticas del tipo: el reemplazo de una palabra "u" por una palabra "v" en una expresión bien formada, ¿presupone una expresión bien formada?

a) Gramáticas formales. CHOMSKY (1956) propuso, en primera instancia, tres tipos de gramáticas formales, constituyendo los conceptos de gramática generativa:

1. *Gramática a número finito de estados* (proceso markoviano[12]): engendrada para algunos tipos de estructuras sintácticas, mas no para todas, pero deficiente para las estructuras condicionales (si… entonces… si no…) o para operar jerarquizaciones que permitan estructurar funciones. Procedimiento a número finito de estados, donde a cada transición de estado (realizado por reglas restrictivas) produce un cierto símbolo.

2. *Gramática sintagmática*: las expresiones del lenguaje se generan por intermedio de reglas formales en número finito (de la misma manera que se estructura el análisis gramatical tradicional), lo mismo que un subconjunto de constituyentes, ellos mismos constituidos de componentes y así sucesivamente, hasta llegar a constituyentes no analizables. Es una construcción jerarquizada de elementos encajonados: sintagma nominal, sintagma verbal,…, en los cuales los constituyentes se clasifican por categorías (nominal, verbal,…).

Por ejemplo, un árbol, cuya sima (raíz) se presenta como un "axioma" (el símbolo F, como frase), un vocabulario común (las categorías gramaticales como grupo nominal: GN, grupo verbal: GV; nombre: N; verbo: V…, llamado "vocabulario auxiliar"); las unidades lexicales que son el "vocabulario terminal" y el conjunto de reglas (llamadas "sintagmáticas")

[12] MARKOV se refiere a un "procedimiento de cálculo que se aplica según una descripción precisa sobre objetos dados, que se puede hacer variar, según el resultado buscado".

permiten engendrar las expresiones, y su descripción. Al introducir en la gramática las reglas llamadas "recursivas", es posible generar un conjunto general de expresiones.

En el ejemplo, se puede presentar una gramática muy simple:

F(rase) → GN + GV

GN → artículo + N

GV → V + GN

V → lleva

N → señor, perro

Artículo → el

Las reglas desarrollan o reescriben la parte izquierda de la flecha, bajo la forma de lo que se encuentra a la derecha de la flecha. El signo "+" significa que los dos constituyentes son obligatorios, la coma (,) significa al contrario, que el símbolo a la izquierda de la flecha puede reescribirse según selección por cualquiera de los elementos separados por la coma (,).

La representación en árbol sería:

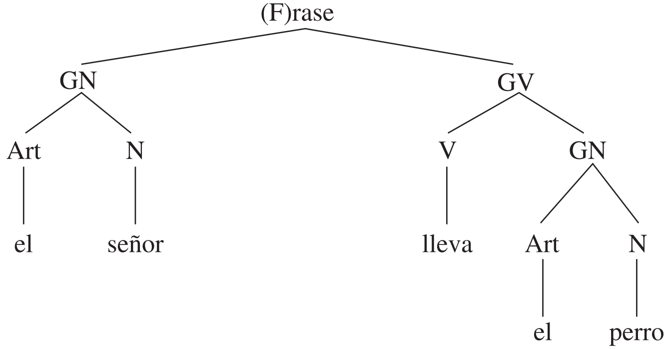

Figura 2. Gramática representada en árbol.

CHOMSKY estructura las reglas, les da un orden y son entonces aplicadas secuencialmente, volviendo a la primera, cuando se aplica la última de la lista, simplifica, así, la gramática sin perder la capacidad generativa. Por tanto, este modelo de gramática se presenta como un algoritmo y es el modelo de gramáticas sintácticas.

3. *La gram*ática transformacional: las gramáticas propuestas, engendran estructuras que pueden comprender unidades lexicales y elementos del vocabulario auxiliar (verbo, nombre,…), que producen luego una cantidad de expresiones entre las cuales persiste bastante ambigüedad. En este sentido, es necesario establecer reglas de transformación que conviertan estas estructuras en nuevas estructuras ("derivadas") y que producen "cadenas terminales" o expresiones más precisas. Así, una *gramática transformacional* permite seleccionar y estructurar ciertas ambigüedades de la gramática generativa.

Se tienen, por tanto, dos tipos de transformaciones:

a) las obligatorias, que deben necesariamente ser aplicadas, teniendo como resultado la *frase-núcleo* o *primitiva* (afirmativa o activa);

b) las facultativas, que engendran las frases derivadas, que no tienen las características de frases-núcleo, como las negativas, pasivas o interrogativas.

Estos tres tipos de gramática constituyen la primera base de estructuración de una gramática sintáctica que va a completarse con la *jerarquía de gramáticas*, propuesta por CHOMSKY, posteriormente.

b*) Jerarquía de gramáticas sintácticas.* En el artículo de 1959 (CHOMSKY, 1959), él presenta una profundización del modelo de estados finitos, donde sus propiedades matemáticas se revelan adaptadas a los desarrollos en el campo de las comunicaciones. Estos se han interesado en el cálculo del número de frases de una longitud dada, lo mismo que a la representación de la generación de un lenguaje por un árbol o un grafo de un autómata finito, asegurando así la equivalencia de una gramática con un algoritmo y materialmente a un dispositivo finito.

CHOMSKY define tres condiciones restrictivas que son aplicadas a la gramática desarrollada, que, a su vez, producen cuatro tipos diferentes de gramáticas a las tres presentadas. Son las gramáticas de tipo 1, 2, 3 y 0, respectivamente, si la gramática satisface las condiciones 1, 2 y 3. La gramática de tipo 0 no tiene ninguna condición.

En general, las restricciones no son independientes, así las gramáticas de los tipos $i >= 1$ son un caso particular del tipo $i - 1$[13], es decir, que son de intensidad creciente; con estas restricciones se fundan las relaciones entre estos nuevos tipos de gramáticas y las máquinas (informática).

Las tres condiciones restrictivas son:

• Restricción 1: las reglas autorizan solo la reescritura de un símbolo a la vez, así que el lenguaje de tipo 1 es decidible. Cualquier derivación de una frase del lenguaje son listas de longitud inferior o igual a aquella de la frase;

• Restricción 2: las reglas permiten la descripción de la estructura sintáctica de los lenguajes naturales y de la estructura de ciertos lenguajes formales, especialmente aquellos de las máquinas;

• restricción 3: se consideran solo los sistemas a estados finitos. Se interpretan los símbolos no terminales, designan-

[13] Por ejemplo: una gramática de tipo 2 comporta aquella de tipo 1.

do tanto los estados y los símbolos terminales que aquellos producidos durante las transiciones. Una gramática de tipo 3 equivale a un autómata de KLEENE, un lenguaje de tipo 3 equivale a los "eventos regulares".

Se puede decir que las perspectivas teóricas de los niveles 0 y 3, y de otra parte los niveles 1 y 2, presentan intereses diferentes:

• el primer grupo (0 y 3) no responde a los modelos de lenguajes naturales, sino a las máquinas de TURING, por un lado, y a los autómatas finitos y expresiones regulares (de KLEENE), por otro;

• el segundo grupo (1 y 2), parte de las modelizaciones de los lenguajes naturales y establece el elemento motor para completar la teoría de autómatas. Las gramáticas tipo 1 y 2 permiten la construcción de árboles generativos, es decir, asociar términos sintagmáticos a las derivaciones; ellas precisan la noción de gramática sintagmática y formalizan el análisis tradicional y las gramáticas de constituyentes.

c*) Gramáticas contextuales y fuera de contexto.* A partir de la gramática sintagmática, CHOMSKY plantea la concepción de otros dos tipos de gramáticas: "fuera de contexto" y "contextuales".

a*') Gramáticas "fuera de contexto".* En la gramática de tipo 2, se dispone de una capacidad generativa que queda intacta a pesar de las restricciones impuestas para este tipo. Se obtiene un resultado de normalización de la forma A → a o A → BC, de manera que sea posible suponer que cada sintagma posee máximo dos constituyentes, como producto de los árboles generativos diádicos. Así, se enuncia la condición bajo la cual una gramática "fuera de contexto" lleva a una gramática de tipo 3 (llamadas "regulares" o de KLEENE), porque este tipo de gramáticas se estructura en los sintagmas cuyo miembro derecho de las reglas es uno solo (A → a). CHOMSKY demuestra que la superioridad generativa de esta gramática sobre las gramáticas de tipo 3 (regulares), se debe a la presencia de un elemento de "autoencajonamiento", es decir, un símbolo no terminal A.

Los importantes resultados de estas gramáticas se encuentran en el desarrollo de los lenguajes formales y luego en los lenguajes computacionales.

b*') Gramáticas "contextuales".* Estas son más difíciles de tratar que las de "fuera de contexto", generan lenguajes que forman una clase muy grande, porque su capacidad generativa autoriza a diversos tipos de derivación, lo que hace imposible una descripción estructural que satisfaga las expresiones o frases.

Las gramáticas de tipo 1, no permiten engendrar sino lenguajes recursivos; con el fin de reducir la vasta clase del lenguaje contextual, es necesario una condición de "buena formación" de la gramática que requiere un algoritmo para determinar si una palabra dada aparece como parte de una derivación de una frase del lenguaje. Este algoritmo no existe, lo que implica que una gramática contextual es indecidible y, en consecuencia, el modelo contextual es inaceptable como gramática para un lenguaje informático o computacional.

d*) Formalización de las gramáticas.* A partir de las gramáticas fuera de contexto y de la jerarquía chomskyana, se busca una formalización más importante, con la investigación de procedimientos operativos, algoritmos o autómatas capaces de engendrar diferentes lenguajes.

La formalización de los lenguajes “fuera de contexto”, se realiza a partir de un procedimiento efectivamente calculable: *el autómata a pila*. Este se utiliza en diversos problemas relativos a la sintaxis de lenguajes, en vista, entre otros, de la traducción automática. La idea de predicción es la base de este enfoque, llamado “análisis predictivo”, en la cual los morfemas que aparecen al descifrar una frase, permiten realizar prospecciones sobre la función gramatical de aquellos que aparecen ulteriormente.

Alrededor del concepto clásico de *autómata a pila* se desarrolla una nueva parte de la teoría de autómatas, en la cual se elaboran diversas variantes del modelo principal y donde se comparan las posibilidades y propiedades. De esta manera, se puede desarrollar el enfoque algebraico de los problemas de la lingüística de CHOMSKY, cuyo interés se manifestó, con mayor importancia, en programación. Con los autómatas a pila, se llegó a unificar una pequeña parte de la teoría de autómatas o máquinas abstractas.

Hacia 1962, se dispone simultáneamente de una caracterización en términos de tipos según la jerarquía chomskyana de gramáticas y en cuanto a autómatas u otras estructuraciones importantes, las cuales consolidan la teorización de una sintaxis formalizada, así:

• las expresiones regulares de la gramática de tipo 3;

• las gramáticas categoriales de la gramática de tipo 2;

• las equivalencias variadas de la enumerabilidad recursiva de la gramática de tipo 0;

• los lenguajes recursivos de la gramática de tipo 1 (para los lenguajes contextuales), que son mal conocidos;

• los autómatas a pila, de la gramática de tipo 2 (para los lenguajes “fuera de contexto”).

Anotaciones. A pesar de las suposiciones de Chomsky y Miller de 1959, en las cuales el lenguaje aceptado por un autómata es engendrado por una gramática de tipo 3, este tipo de gramática no parece llegar a resultados afirmativos para las gramáticas y lenguajes computacionales. Por el contrario, la gramática de tipo 2, más cercana a los lenguajes naturales, creó las posibilidades del desarrollo de gramáticas y lenguajes de programación a partir de las propiedades de los lenguajes naturales. Así mismo, si las gramáticas contextuales aportan las primeras bases con el fin de realizar procedimientos operativos o autómatas, ha sido necesario profundizar las gramáticas "fuera de contexto" para ampliar las gramáticas computacionales.

Además, la solidez de una sintaxis de un lenguaje computacional se profundiza por medio de una fase experimental, con la creación de los computadores.

D) *Vía práctica: producción de lenguajes de programación*

Desde los años cincuenta del siglo xx, los investigadores de la teoría informática formularon la necesidad de contar con lenguajes más accesibles a los usuarios: "lenguajes intermediarios".

El proyecto sobre "lenguajes intermediarios" exigía un lenguaje más rico y de fácil empleo; no podía ser una simple lista de elementos o direcciones, era necesario que se combinaran los elementos en nuevas unidades de sentido, esta combinación se regula por la sintaxis. La traducción de las frases necesita un análisis sintáctico que no se reduce solo a la consulta de un diccionario, era necesario que estos lenguajes de programación fueran independientes de las máquinas y capaces de ser utilizados por una gran variedad de máquinas: los lenguajes universales, definición más extensa y universal de un lenguaje de programación. Para ello, se pusieron en funcionamiento procedimientos para la manipulación de símbolos antes de procedimientos para el cálculo ordinario, aproximándose el enfoque lingüístico y el sintáctico, así:

• Los lenguajes universales se formalizan, provistos de reglas sintácticas de formación de configuraciones aceptables de símbolos.

• Los procedimientos que manipulan los símbolos constituyen la noción de compilación, que en realidad es una *traducción*; es, por tanto, la teoría chomskyana de gramáticas y de autómatas, que presenta las bases para esta traducción.

Por consiguiente, la definición de un lenguaje de programación en términos de *gramáticas fuera de contexto*, le da un uso más seguro y neto de las características formales necesarias en los programas. El papel más importante de estas gramáticas junto con los *autómatas a pila*, se desarrolla en la segunda fase de la compilación, es el análisis sintáctico ("parsing")[14], que consiste en asociar a la cadena

[14] La primera fase de la compilación es el *análisis lexical*, que consiste en separar la cadena de caracteres en cadenas que pertenecen a las categorías sintácticas determinadas; esta tarea es similar al *análisis gramatical* de la gramática de cualquier lengua natural (MOSCONI, JEAN, 1989, pág. 761). Por ejemplo, el análisis lexical indica que una cadena de caracteres puede ser considerada de la forma:

(*<identificadorr>1* + *<identificador>2*)

Según las reglas de la sintaxis del Algol, un identificador se constituye de una cadena finita de letras, seguida de una cadena finita de letras o de “palabras” resultado del analizador lexical, un indicador sintagmático, revelando su estructura sintáctica o árbol generativo; se muestra, por tanto, que si la construcción es imposible, indica que las reglas de la sintaxis no han sido contempladas, y, por consecuencia, la cadena propuesta se deja de lado.

Así, los especialistas de la programación llegaron independientemente a ideas vecinas de aquellas formuladas por Chomsky. Un lenguaje como el Algol, tiene una sintaxis idéntica a una *gramática fuera de context*o de Chomsky. Las fórmulas de definición son análogas a las reglas de reescritura *fuera de contexto* (los símbolos no terminales); igualmente, la técnica de descripción sintáctica permite un estudio matemático de las propiedades de posibilidad del lenguaje. De esta manera, el desarrollo ulterior de los lenguajes de programación basado en las gramáticas sintácticas, desarrolla un lenguaje de máquina cercano al cálculo numérico (recursivo, binario) y un lenguaje de manipulación de símbolos.

5. Semántica

Las expresiones codificadas y sintácticamente estructuradas en una gramática formalizada de un lenguaje computacional, no tienen valor si ulteriormente no se benefician de una interpretación. Esta interpretación se realiza bajo la forma de un *modelo* (véase cap. iv, sec. 2, apdo. D, lit. b: “Método del modelo”), donde para una clase de las apli-

---

cifras. Para las unidades lexicales, de la mayoría de los lenguajes de programación se pueden expresar por una expresión regular, así:

(<*letra*> <*letra v cifra*>)

caciones dada, existe una aproximación que conviene para todas las operaciones; y es sobre esta aproximación única que se expresan todas las aplicaciones o “conocimientos” del sistema, lo que significa que las representaciones que se enuncian se interpretan sobre este modelo dado.

La construcción semántica del lenguaje computacional, a partir del *modelo*, tiene como objeto desarrollar un núcleo formal de significado del lenguaje dado, por medio de la relación de un significado y cada palabra o expresión del lenguaje. Esta semántica presenta las reglas necesarias para la construcción de la interpretación del lenguaje computacional o para elaborar la clasificación de los diferentes lenguajes computacionales. La semántica está ligada estrechamente con la práctica de la programación, pero no se puede desarrollar fuera de las nociones *a priori* de la informática[15].

Esta semántica se puede ver desde dos niveles: un primer nivel enfocado a determinar el significado de las expresiones de un lenguaje computacional y un segundo nivel que determina el comportamiento de los programas (procedimientos) a partir de los valores semánticos.

A) *Semántica de primer nivel*

Determinar el significado de las expresiones de un lenguaje computacional, exige un principio general que puede ser formulado independientemente de toda concepción particular de la naturaleza del significado. Se trata del principio “composicional”, según el cual el significado de una *expresión compuesta* se determina por la estructura de la

[15] Por ejemplo, el concepto de *transparencia referencial* de QUINE, que se muestra en la semántica de segundo nivel.

expresión y por el significado de las expresiones simples que la componen, lo que nos lleva al principio de *analicidad*, presentado para los lenguajes formales. Así, el componente semántico de la gramática está enfocado a formular las reglas que permiten determinar cuál es el significado de una expresión compuesta, teniendo como dado el significado de las expresiones simples que la componen y la manera como están combinadas, es decir, la parte sintáctica de la expresión.

Se trata, por tanto, de caracterizar una cierta relación de dependencia entre los significados de diferentes tipos de expresiones. Existe, así, una relación entre la interpretación de la estructura y aquella de los componentes: se llama *Int* (x) a la interpretación de una estructura x, y se supone que x es ella misma compuesta por un símbolo funcional f aplicado a los símbolos de los componentes a y b; el caso ideal es aquel donde *Int* (x) se define como la aplicación de *Int* (f), *Int* (a) e *Int* (b).

Se tienen, por consiguiente, tres diferentes formas de interpretación para los lenguajes computacionales: denotacional, inferencial y verificacionista.

a*) Interpretación denotacional o referencial.* En una estructura arborescente de la sintaxis de un lenguaje informático, los elementos sin sucesores (las hojas del árbol) se pueden interpretar como "objetos reales" del modelo y que deben ser identificados claramente y en ciertos casos físicamente (registro en la memoria de la máquina). En este caso, se dice que la estructura "denota" o designa los objetos, por lo cual se puede hablar de interpretación denotacional.

b) *Interpretación inferencial.* La interpretación inferencial complementa la interpretación denotacional, porque las expresiones representadas en términos de símbolos denotan objetos que no son directamente fáciles de utilizar o procesar, con el fin de llegar al final del procedimiento. Entonces, es la aplicación de las reglas de inferencia aplicadas a representaciones simbólicas, lo que permite llegar a expresiones interpretadas. La interpretación inferencial se revela indispensable para establecer este primer nivel de la semántica.

c) *Interpretación verificacionista.* El método de verificación o adecuación "lenguaje-realidad" para un lenguaje computacional, es utilizado en dos momentos de su realización:

• Por una parte, la verificación de las expresiones e instrucciones de un algoritmo-programa se efectúa, en la mayoría de los casos, automáticamente, según parámetros presentados por las reglas sintácticas y semánticas del lenguaje; es decir, que siguiendo a TARSKI, según un "modelo de expresiones de base", es un lenguaje *compilador* o *interpretador* que verifica la validez de las expresiones del programa.

• Por otra parte, los resultados de los programas son objeto de verificación, porque ellos pueden corresponder o no a las características o propiedades que debe realizar el programa. Estos resultados, que son la traducción o interpretación de las expresiones o instrucciones del programa, se verifican por el programador (humano): verificación diferente de aquella realizada automáticamente por la máquina.

B) *Semántica de segundo nivel*

Este nivel de semántica comporta la estructuración de los programas y la clasificación de los lenguajes computacionales[16]. Esta semántica se basa en el principio de *susti-*

[16] Los primeros estudios de la semántica de estos lenguajes se realizaron por el grupo asociado a CHRISTOPHER STRACHEY, PETER LANDIN,

tucionalidad, que significa la posibilidad de reemplazar un elemento por otro. Así, en el caso de un programa, se habla de reemplazar los constituyentes de los programas por otro que les son equivalentes, como son los procedimientos, rutinas, bloques,…

La *sustitucionalidad* conlleva la noción de "equivalencia observacional", en la cual, por ejemplo, para dos programas, P y Q, cuyas entradas son las mismas y por consecuencia sus salidas son equivalentes, se pueden sustituir las expresiones de un programa (P) por otras expresiones estructuradas en procedimientos de otro programa, por lo cual los dos programas son equivalentes. De la misma manera, si los constituyentes de entrada de los programas P y Q son diferentes, se puede concluir que los programas son diferentes. Pero esta conclusión nos lleva a los problemas de identificación y de diferencia entre programas[17]; lo que nos llevaría a hacer un análisis minucioso de sus constituyentes de entrada y salidas, que producen una *equivalencia observacional*. En conclusión, la *sustitucionalidad* con la ayuda de estas "equivalencias observacionales", representa una propiedad extremadamente polivalente y muy importante para la eficiencia de los programas.

Igualmente, el principio de *sustitucionalidad* ayuda a analizar y señalar con seguridad las diferencias y semejanzas

Dana Scott y otros, hacia 1960. Graham White, "The philosophy of computer languages", pág. 239, en *The blackwell guide to the philosophy of computing and information*, Ed. by Luciano Floridi, Blackwell Publishing Ltd., 2004.

[17] Una respuesta trivial consiste en la identificación del código fuente de la expresión; pero esto no es interesante porque los programas pueden variar su forma de expresiones sin cambiar la acción que realizan, entonces, son los mismos programas con códigos fuentes diferentes.

de los lenguajes computacionales, con el fin de realizar una taxonomía de lenguajes.

Más concretamente, con el fin de aplicar el principio de *sustitucionalidad*, es necesario describir las operaciones, por las cuales las expresiones codificadas del programa o entidades sintácticas de un lenguaje de programación se asocian a objetos (interpretación *denotacional*). De manera que los valores se asignan a las diversas entidades: variables, constantes… tanto a la totalidad del programa como a los diferentes constituyentes (procedimientos, bloques, rutinas,…). Sabemos, en consecuencia, que una de las bases de este tipo de semántica y de la *sustitucionalidad* es de ser *denotacional*, entonces los programas tienen valores semánticos, en todos los niveles de los constituyentes. Su objetivo es el de facilitar las tareas del programador, teniendo la buena definición de los valores semánticos durante el desarrollo del programa. Este requisito significa que los valores semánticos asignados a los diferentes constituyentes (procedimientos, bloques, subrutinas) forman un sistema completamente coherente.

Pero este sistema no puede conservar, continuamente, su consistencia; así, al considerar, por una parte, un caso particular de entidad de programación: el procedimiento o rutina que se elaboran a partir de ciertas expresiones pueden ser invocados con valores particulares, sobre los cuales, a su vez, se pueden realizar múltiples acciones. Tal *rutina* puede ser pensada como una clase de función: sus argumentos son valores semánticos de sus parámetros y su valor es el valor semántico de la expresión a la cual reenvía.

Por otra parte, generalmente, sabemos que el desarrollo de la semántica es un proceso de elaboración progresiva de los valores semánticos; así, por ejemplo: las variables tienen un significado a partir de los valores, las *rutinas*, ellas lo son por medio de parámetros de funciones que reenvían a los valores. En esta concepción de *rutina*, es imposible prever la *sustitucionalidad*, porque si los valores se asignan a los programas y a sus constituyentes en un primer momento, estos valores semánticos se han cambiado poco a poco de la semántica original, produciendo una "opacidad referencial", cuestión que ha sido señalada por QUINE[18].

Con el fin de cumplir con los objetivos y con buenos niveles de consistencia, la semántica de segundo nivel utiliza herramientas como la lógica de alto rango (escuela intuicionista)[19] y de la teoría matemática de categorías (los lingüistas utilizan la lógica de primer orden y la teoría de conjuntos). Igualmente, la semántica utiliza la teoría de pruebas, porque un programa corresponde, en cierto sentido, a las pruebas de expresiones o proposiciones. En ese caso, se pueden tener varios tipos de pruebas para una proposición, que conducen a realizar una equivalencia de pruebas[20].

La utilización de una semántica de segundo nivel es parcialmente utilizada en el desarrollo de la programación,

[18] WILLARD VAN ORMOND QUINE, "Speaking of objects", en *Ontological relativity and others essays*, 1969. Citado por GRAHAM WHITE, *op. cit.*, pág. 243.

[19] Por ejemplo, la "Floyd-Hoare Logics", ligado al razonamiento alrededor de los programas y los valores semánticos. Esta lógica ayuda a mostrar las diferencias de razonamientos y de valores semánticos de los programas. GRAHAM WHITE, *op. cit.*, pág. 241.

[20] JEAN-IVES GIRARD, "A new constructive logic: Classical logic", *Mathematical structures in computer science*, 1(3): 255-296, 1991. Citado por GRAHAM WHITE, *op. cit.*, pág. 245.

debido al grado de indecidibilidad (inconsistencia) existente todavía en sus desarrollos, pero se utiliza con éxito en la parte metateórica en informática. Al contrario, la semántica de primer nivel es enteramente utilizada en todos los procesos realizados por un computador.

Anotaciones. La estructura de la *semántica* de un lenguaje computacional, está basada en la estructura de los lenguajes formales, de manera que la noción *denotacional*, el tipo de relaciones entre las expresiones, el sistema *inferencial* de reglas, la validación y verificación de las expresiones toman como soporte las características de estos lenguajes, igualmente (véase cap. II, sec. 4, apdo. A: "Ejes principales del lenguaje formalizado").

Se dispone así, para esta semántica, en su aspecto *composicional y denotacional*, la base lógica en el eje analítico y proposicional del lenguaje; de manera que la exigencia de una denotación para los nombres, hace posible el aspecto *denotacional* y la exigencia de la *elementaridad* asegura la *composicionalidad* del lenguaje computacional. Se encuentra, así, las posibilidades de una semántica enteramente formal, donde aquello que es de exigencia lógica se determina por la realización de las expresiones del lenguaje.

C) *Asimilación de la sintaxis y semántica lógica en los lenguajes computacionales*

Por lo general, los lenguajes computacionales, en la semántica de primer nivel, se estructuran según el principio de la *composicionalidad*. El significado de una expresión compuesta se determina por la estructura de cada expresión, así como por el significado de las expresiones que la componen. Este principio, es puesto en funcionamiento gracias a la propiedad de *analicidad* de la proposición, articulada según un *sentido* (posibilidades de lo verdadero y falso) en la cual las expresiones del lenguaje pueden ser presentadas según sus diferentes elementos. De la misma manera, el resultado de esta *analicidad* debe ser precisado por la vía de un procedimiento de decisión o mecánico, como son las *tablas de verdad*; para el caso de los lenguajes computacionales, como vimos anteriormente, el procedimiento es *efectivamente calculable*, cuyos exponentes son, por ejemplo, los árboles[21].

Igualmente, un lenguaje computacional debe corresponder imperativamente a un principio *denotacional*, y es por esto la importancia de la distinción entre el *sentido* y la *denotación* o *referencia* (véase cap. IV, sec. 2: "Nociones generales de la semántica"); por un lado, como nivel abstracto y formalizado que representa el *sentido* o condiciones de verdad en un programa o algoritmo y, por otro, un nivel del objeto como tal que denota, designa o referencia la dirección del lugar en la memoria de la máquina, en la cual se encuentra el objeto o información (datos, programa,…).

Por tanto, esta información (objeto) representa la base para la interpretación denotacional de una semántica del lenguaje computacional. Este lenguaje define los elementos llamados "palabras propias del lenguaje" (constantes y operadores lógicos) que permiten la producción de las expresiones compuestas. Estas *palabras* no se relacionan con ningún *objeto* y son puramente operativas, articulan las

[21] Este procedimiento es útil cuando se trabaja con más de tres variables.

proposiciones, razón por la cual las expresiones del lenguaje computacional pueden estructurarse y calcularse según la configuración del algoritmo o programa.

Para que el lenguaje computacional pueda producir resultados o hechos, es necesario disponer del principio *inferencial*; así las expresiones compuestas, que no son verificadas inmediatamente, son el resultado de operaciones de cálculo o de combinaciones cuya *efectuación* es independiente de la verificación de lo *real*, efectuación realizada a partir de un sistema de reglas y operaciones.

El conjunto de las expresiones de este lenguaje es una articulación y configuración precisa de variables, constantes y operadores lógicos. Los nombres (variables sustituidas) presentan los valores; es decir, que los valores designan el significado tanto de las expresiones, como de la *información dada* o dirección de la memoria, donde se registra esta información. Así, es por este valor "concreto" de las variables que un programa lleva a los resultados y a su "parada".

Según el propósito de WITTGENSTEIN, estas expresiones tienen un *sentido* porque ellas representan un "estado de cosas"; son configuraciones[22] de variables, constantes y

[22] TURING fue uno de los primeros en presentar la idea de configuración; se refería a las posiciones mayúscula y minúscula de una máquina de escribir: una de las dos posiciones presentaba la "configuración" de la máquina. Sin embargo, él pensaba en máquinas que podían ser posicionadas en un número superior a dos, pero finito, de configuraciones posibles. Él incorpora la idea de una máquina de escribir en la cual la tecla que golpea se desplazaría con respecto a la página, así la acción de la máquina debía ser totalmente independiente de su posición. De la misma manera, pensaba en máquinas capaces no solo para escribir símbolos, sino también para escribir con más rapidez; de ahí la idea de una banda de papel ilimitado (permitiendo olvidar las márgenes y las

operadores lógicos que pueden o no ser realizados (es decir, que forman parte del mundo de los hechos, tal como él es). Igualmente, los lenguajes computacionales necesitan postular que existe una posibilidad de adecuación entre el lenguaje y lo real; esta última debe ser "materializable" por valores, datos, direcciones, "cosas simples", que dan la condición de posibilidad del lenguaje computacional.

En informática, se pueden evocar dos niveles de "real": aquel referido al lugar de la memoria de la máquina en la que se encuentran registrados, datos, constantes…; y el real como resultado final o producto de la ejecución de un programa. Los dos niveles de real son indispensables en el funcionamiento del programa y su realización.

A manera de ilustración, vamos a examinar atentamente estas nociones de la semántica computacional, a partir del siguiente ejemplo:

Disponemos de un programa, en Algol, cuyo objetivo es adicionar 10 números (1, 3, 5, 6, 8, 10, 15, 17, 20, 25) que están almacenados en un archivo, así:

```
Principio
    Mientras ARCHIVO:= < 10
        Hacer
        s : = s + x
        x : = x + 1
Fin
```

---

interlíneas), dividido en "cajas" con fines de precisión suplementaria. Para ello, era indispensable también presentar el problema de la decidibilidad de HILBERT descifrando toda aserción matemática para indicar si es demostrable o no, decisión que debía ser tomada sin la menor interferencia de la inteligencia.

Igualmente, disponemos de un lenguaje compilador del Algol, que tiene un componente sintáctico con reglas sintácticas para la realización del análisis lexical y sintáctico, así que un componente semántico con reglas semánticas. Estas últimas tienen la forma de "expresiones de base". El resultado previsto, luego de la adición de los 10 números, realizado por el programa es 107.

A partir de este algoritmo, se pueden ilustrar ciertos aspectos sobre la sintaxis y semántica del lenguaje computacional:

• Este algoritmo es un enunciado, que tiene un *sentido* (nombres articulados) y representa un "estado de cosas" o una configuración de acciones que se pueden realizar o no, según si el programa llega a su objetivo y tiene la "parada". De esta manera, las acciones (instrucciones) van a producir "hechos" (la adición de los 10 números, con un valor de 107, en el ejemplo).

• Cada término del enunciado está estructurado según un orden de la configuración del algoritmo, que marca las relaciones con los otros términos y son inmodificables, porque cambiaría el sentido del enunciado y del algoritmo mismo; por ejemplo: si en lugar de plantear esta *configuración*, se hace una lista de términos, no se podrá nunca llegar a la realización del programa. Así, uno de los elementos fundamentales en la construcción del algoritmo es su *configuración*, la cual no corresponde a los objetos o entidades mismas, sino a la estructuración sintáctica y semántica que debe tener cada enunciado, expresión o instrucción del algoritmo, para que pueda ser realizado (condición de verdad). Esta estructura presupone un "hecho" (es decir, que el enunciado y el conjunto de enunciados se realicen y que el programa llegue al fin previsto).

• Este enunciado es compuesto, con la ayuda de constantes lógicas y de variables, cuyo valor de verdad se determina por los enunciados simples. En nuestro caso, se tienen las siguientes constantes y operadores lógicos: “principio… fin”, “mientras… hacer”, “+”, “:=”, “<”, que no representan nada en la realidad de los “hechos” (en el resultado de la adición) y, sin embargo, son utilizados en todos los algoritmos y programas y que pertenecen al lenguaje. Estas constantes y operadores permiten el dinamismo del programa, por su carácter operativo. Igualmente, se disponen de las variables siguientes: “s”, “x” y “ARCHIVO” que son sustituidas por valores de la información dada, según las reglas del lenguaje en general y del programa en particular.

• Los resultados de los cálculos y operaciones determinadas por las constantes lógicas y la sustitución de variables, según las reglas presentadas por el lenguaje de programación y correspondientes a la configuración del algoritmo o programa, son las proposiciones elementales.

De esta forma, tenemos los siguientes enunciados simples: “ARCHIVO := 1” y “s := 1 + 2”. Las cifras (números) “1”, “2”, denotan los valores últimos de la expresión.

• Para un lenguaje computacional, en un programa o algoritmo, los nombres que denotan las “cosas simples”, no son ni indestructibles, ni datos sensibles, sino datos definidos como nombres y variables sustituidas para el conjunto del programa, poseen un carácter finito y determinado en el programa en cuestión.

Así que por su definición precisa, sus valores, su localización en la memoria y su duración definida, estos datos sintetizan la condición para la realización, interpretación, validez y decidibilidad (problema de la “parada”) del programa o algoritmo; es decir, la condición de posibilidad del programa y, por consiguiente, del lenguaje computacional.

Anotaciones. En consecuencia, los aspectos sintáctico-semánticos de un lenguaje formal, nos presentan las nociones de la sintaxis y semántica de primer nivel de un lenguaje computacional. Sabemos que esta semántica comprende nociones suplementarias como la verificación, lo mismo que nociones como la sustitución y la de la “significancia” (*significance*). Así pues, el principio de representación nos presenta ciertos criterios para la relación lenguaje-real; será necesario indicar las modalidades con el fin de realizar la correspondencia como la “interpretabilidad”, la traducción de diferentes lenguajes y la concretización de la sustitución y de la “significancia”.

D) *La verificación de los lenguajes computacionales*

Hemos presentado el proceso de verificación de un lenguaje computacional que se realiza durante dos momentos de la ejecución de un programa: en el primero: las expresiones o instrucciones del algoritmo o programa constituidos a partir de las reglas del lenguaje, son el producto de una aplicación específica. En este caso, las instrucciones son verificadas de manera automática. En el segundo momento, los productos resultantes de la ejecución de los programas, las expresiones de los algoritmos o programas siguen una transformación (pasando por un proceso de *traducción* entre varios niveles de lenguajes) hasta la presentación de los resultados, que forman parte de los *hechos de la experiencia*; es decir, que el resultado de un programa puede ser catalogado como un "objeto de la vida cotidiana", observable, como textos, gráficos, facturas,…

En el primer caso, la verificación es automática y se considera que el significado de estas expresiones se presenta por medio de *enunciados de base*; es decir, que un enunciado significativo es sinónimo o pertenece a una categoría de un *enunciado de base* o *núcleo de expresiones*, creado a partir de reglas semánticas del lenguaje computacional y, en consecuencia, la verificación se realiza por la máquina, de manera que las condiciones de verdad de una expresión corresponden unívocamente con las condiciones de *asertabilidad.*

En este caso, las condiciones epistémicas o condiciones de un usuario del lenguaje, son generales para el conjunto de las aplicaciones del lenguaje. Es, por tanto, "la máquina que decide", a partir de las reglas preestablecidas; al contrario de un lenguaje ordinario cuya aplicación conlleva ambigüedades o formas particulares de justificación.

Por tanto, si las condiciones de la significación de los enunciados se definen "objetivamente" como un método o procedimiento con el fin de verificarla o rechazarla, es entonces posible presentar el significado de una expresión del lenguaje computacional como su método de verificación o rechazo.

En el segundo caso, la verificación por el programador, no mecánica, en donde las expresiones de los algoritmos o programas siguen una transformación, cuyos resultados tocan la experiencia, los resultados son objeto de verificación, porque pueden corresponder o no a las características acotadas por el programa; se verifica, por tanto, el procedimiento, el método, el algoritmo y sus resultados.

Anotaciones. Podemos pensar que el objetivo final del lenguaje computacional es relacionar el *lenguaje* y la *realidad*, por lo cual el desarrollo de la semántica se efectúa a través de diversos niveles de formalización; partiendo de una “forma como imagen”, siguiendo con una adecuación estructural y, finalmente, la realización de la traducción, en donde se muestra una unidad del desarrollo a partir de la presentación de las expresiones intuitiva (extralingüístico), formalizada y abstracta (lenguajes computacionales) y, por último, un nivel intuitivo en la verificación. Así tenemos:

• La imagen: la proposición misma (lenguaje formal) y las expresiones (lenguajes computacionales) que son, a su vez, una imagen del “hecho posible”, pero que paradójicamente no tiene una similitud material, pertenece al orden extralingüístico, fuera de los lenguajes formales y computacionales.

• La adecuación estructural: a partir del análisis formal, se pasa de lo real (hecho), de la imagen de este real (enunciado), a un “plano abstracto” y formalizado, de manera que la expresión de la imagen presenta una estructura formal correspondiente, adecuación estructural que es posible por el método de la proyección (según WITTGENSTEIN). En este sentido, esta adecuación implica una transformación regulada y una dimensión operativa, que estaría soportada por el programa.

• La interpretación o traducción: aquí las operaciones aseguran el paso entre las realidades diferentes. Dar un *sentido*, es traducir: presentar signos en lugar de otros signos. La unidad del lenguaje está en lo operativo y no en lo *referencial* o *denotacional*, de manera que la transformación regulada producida por la proyección tiene como fin la *traducción*.

Así, el proceso de ejecución de un programa computacional es una traducción operativa entre varios lenguajes; interpretador, compilador y de máquina; traducciones que tienen una adecuación estructural con las expresiones primeras, como imagen, que realiza la relación entre el lenguaje y lo real, cuya última *traducción* es la producción de resultados del programa y que en ese caso toca el mundo de lo concreto-real y de los hechos.

CONCLUSIONES GENERALES

La relación de la *lógica contemporánea* o *matemática* y la *informática*, tiene dos papeles importantes: uno, que es externo y teórico y reproduce la relación que se tiene con los *fundamentos de las matemáticas* y, el otro, es interno y de tipo práctico, porque la lógica ha encontrado un terreno donde sus métodos son aplicados y experimentados.

En el primer caso, vemos el desarrollo de la noción de calculabilidad, los resultados de la indecidibilidad (teoremas de Church, Godel, Turing,…) y la clasificación de los problemas según su complejidad, que van a colocar a la lógica como una *metainformática*.

En el segundo caso, interno y práctico, se puede pensar en el *formalismo hilbertiano*, trasladado a un programa de computador, que calcula una función numérica ϕ como construcción abstracta; si se le presenta un argumento 10 a ϕ, se dispone de una descripción abstracta de un entero $\phi(10)$, y la pregunta es de saber si *sí* o *no* este entero puede escribirse directamente en numeración decimal. Si se hace la analogía *demostración* = *programa*, *demostración real* = *valor numérico*, la ejecución del programa $\phi(n)$; es decir, la búsqueda del valor numérico es muy cercano a aquello propuesto por Hilbert. Se trata, por tanto, de objetivar las manipulaciones lógicas, lo que ha dado grandes frutos para la informática. La eliminación del *infinito* que procuraba

Hilbert, se verifica a través de diversas generalizaciones y el resultado es la gran dinamización de los procesos que dan como resultado: *finito* + *dinámico*, que es el resultado de las aplicaciones en informática.

En informática se utilizan *objetos finitos*, como números, programas, archivos; la lógica utiliza términos, expresiones bien formadas o fórmulas y pruebas. En ambos casos, son las *palabras* (v. gr., la cadena de caracteres) que son empleadas para trabajar con los objetos matemáticos o informáticos; cada una de ellas respeta un lenguaje: formal, por un lado, y computacional, por el otro, compuestas de una sintaxis y una semántica concreta que se escoge según las aplicaciones.

La *sintaxis* en ambos casos nos lleva a plantear las *reglas de formación* y a las *expresiones bien formadas*, con las propiedades consecuentes. La *semántica* cuyo objetivo es la interpretación de estas expresiones en otras teorías; es decir, analizar sus diferentes sentidos o significados: el valor de una expresión en programación y el valor de verdad en una fórmula de la lógica.

Además, la aplicación del lenguaje mismo en la programación, se presenta como una actividad lógica; la especificación de un programa es siempre una fórmula de un lenguaje lógico; concebir un programa es probar una proposición de manera constructiva; ejecutar un programa es normalizar una prueba. Interpretar u optimizar un programa es construir un modelo más o manos estándar, como si se tratara de evaluar la verdad o la consistencia de una proposición lógica.

En consecuencia, es el desarrollo de la *formalización* en concordancia con la simbolización, que han conducido el desarrollo de cada uno de estos lenguajes y marcan las características de estos lenguajes:

• En el *lenguaje formal*, es la objetivación del pensamiento representada por las expresiones bien formadas y sus reglas, en donde se presenta una liberación frente a las variaciones subjetivas individuales; por ejemplo, las expresiones: "me parece…", "es evidente…", no se tienen en cuenta.

• En el *lenguaje computacional*, además de la característica anterior, se encuentra el "seguimiento a la regla", se actúa "mecánicamente", de manera rutinaria.

Así, un lenguaje perfectamente formalizado y simbolizado, como el computacional, se convierte en una regla de transformación de palabras y normalizado en la estructura de un algoritmo o programa. El procedimiento simbolizado, por su forma "sintética", utiliza un espacio reducido que facilita una visión clara. Si el símbolo es controlado y su forma de construirlo es conocida, se pueden examinar las cadenas de prueba o deducción y es posible identificar las proposiciones o expresiones como un solo conjunto o sistema formal.

Este procedimiento, una vez estructurado, se realiza por medio de la aplicación repetida de una *operación*. Entonces, es la pareja *procedimiento-mecánico* y *operación-repetición* que permite la dinamización del lenguaje computacional, lo que hace la diferencia con el *lenguaje formal*.

BIBLIOGRAFÍA

AHO, ALFRED V.; HOPCROFT, JOHN E., and ULLMAN, JEFFREY D.: *Data structures and algorithms*, Addison-Wesley Publishing Company, 1983.

BARBIN, EVELYN: "Les deux faces du théorèmes de Kleene et la question des machines", dans: *Calculs et formes*, Paris, Ellipses Editions Marketing, 2003, págs. 24-51.

BARWISE, J.: (éd.). *Handbook of mathematical logic*, Amsterdam-London, North Holland Publishing Company, 1977.

BERNAYS, PAUL: "La philosophie des mathématiques et la théorie de la démonstration de Hilbert" (1930), dans: *Philosophie des mathématiques*. Introduction et traduction de Hourya Sinaceur. Paris, Librarie Philosophique J. Vrin, 2003 (Mathesis: Michel Blay- Hourya Sinaceur), págs. 80-81.

BLANCHÉ, ROBERT: *L'axiomatique*, Paris, Quadrige puf, 1990 (1ère édition 1955).

BLANCHÉ, ROBERT et DUBUCS, JACQUES: *La logique et son histoire*, Paris, Armand Colin, 1996. (Collection "U", série Philosophie).

BONIFACE, JACQUELINE: *Calculs et formes*, Ouvrage collectif coordonné par J. Boniface, Paris, Ellipses Editions Marketing, 2003.

— *Hilbert et la notion d'existence en mathématiques*, Paris, Librairie Philosophique J. Vrin, 2004 (Mathesis: Michel Blay- Hourya Sinaceur).

BOOLOS, GEORGE S., and JEFFREY, RICHARD: *Computability and logic*, 3ème éd., Cambridge University Press, 1994.

BOUVERESSE, JACQUES: *Pays des possibles: Wittgenstein les mathématiques et le monde r*éel, Paris, Les Eds. Minuit, 1988. Collection Critique.

— *La force de la r*ègle, Paris, Les Eds. Minuit, 1988. Collection Critique.

BRETON, PHILIPPE: *Une histoire de l'informatique*, Paris, Eds. La Découverte, 1990.

CANTOR, GEORG: "Beitrage zur Begrundung der transfiniten", *Mathematische Annalen*, *46* (1895), 481-512; *49* (1897); reimpreso en 1932. Traducido al francés: *Sur les fondements de la théorie des ensembles transfinis,* por F. Marotte. In Mémoires de la Société des Sciences Physiques et Naturelles de Bordeaux, reeditado por Gabay, Paris, 2000.

CARNAP, RUDOLF: *Formalization of logic*, Cambridge, Massachusetts, Harvard University Press, 1943.

— *Introduction to semantics*, Cambridge, Massachusetts, Harvard University Press, 1948.

— *Logical syntax of language*, First published, Kegan Paul, Trench, Trubner & Co. Ltd. Trad. Anglaise, Londres & New York, 1937. Reprinted 2000, Routledge, London. Reprinted 2001, Routledge, London.

— "Foundations of logic and mathematics", Chicago, University of Chicago Press, May 1939; 4th ed., 1947. En: *International Encyclopedia of Unified Science*, vol. 1, Number 3. págs. 1-71.

CAVEING, MAURICE: *Le problème des objets dans la pensée mathématique*. Ouvrage publié avec le concours du Centre National du Livre, Paris, Lib. Philosophique J. Vrin, 2004.

CHAUVIRE, CHRISTIANE: *Ludwig Wittgenstein*, Paris, Seuil, 1989.

CHION, JEAN S. et CLEEMANN, EDOUARD F.: *Le langage Algol W. Initiation aux algorithmes*, Grenoble, Presses Universitaires de Grenoble, 1973.

CHOMSKY, NOAM et MILLER: “Finite state langage”, dans: *Information and control*, núm. 2, 1958, págs. 91-112.

CHOMSKY, NOAM: “On certain formal properties of grammars”, dans *Information and control*, vol. 2; reproduit dans “Readings in mathematical psychology”, Luce, Bush & Galanter Eds.

— *Syntactic structures*, Mouton, 1957. Traduction française: *Structures Syntaxiques*, Paris, Editions Le Seuil, 1969.

— “Three models for the description of language” (in: *IRE “Transactions on information theory”*), 1956. Traducción francesa: “Trois modèles de description du langage” (in: *Langages*, 9, 1968).

COURTIN, JACQUES et VOIRON, JACQUES: *Introduction à l’algorithmique & aux structures de données*, Université de Grenoble- IUT, 1974.

DAHAN-DALMEDICO, AMY et PEIFER, JEANNE: *Une histoire des mathématiques: Routes et dédales*, Paris, Seuil, 1986.

DAVIS, MARTIN: “Mathematical logic and the origin of modern computers”, dans: *The universal Turing machine. A half-century survey*, ed. par Rolf Herken, 1988, págs. 149-174.

DELAHAYE, JEAN PAUL: *Information, complexité et hasard*, 2nd éd. revue, Paris, Hermes, 1999.

DUMMETT, MICHAEL: *Philosophie de la logique*, Paris, Eds. Minuit, 1991.

FREGE, GOOTLOB: *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*, Halle, Nebert, 1879; traduction française: *Idéographie*. Traduction, préface, notes et index par Corine Besson. Postface de J. Barnes. Paris, Librairie Philosophique J. Vrin, 1999. Bibliothèque des Textes Philosophiques.

— *Ecrits logiques et philosophiques*, traduction et introduction de Claude Imbert, Paris, Editions du Seuil, 1971.

GANDY, ROBIN O.: "The confluence of ideas in 1936", dans: *The universal Turing machine. A half-century survey*, 1998.

GARDIES, JEAN LOUIS: *Du mode d'existence des objets de la mathématique*, Paris, Vrin, 2004.

GAUTHIER, YON: *Logique et fondements des mathématiques*, Diderot Eds. Arts & Science, 1997.

GIRARD, JEAN-IVES: "A new constructive logic: Classical logic", *Mathematical Structures in Computer Science*, 1(3): 255-296, 1991.

GIRARD, JEAN YVES: *Le théorème de Gödel*, traductions de l'anglais et de l'allemand par Jean Baptiste Scherrer, Paris, Editions du Seuil, 1989.

GOCHET, PAUL et GRIBOMONT, PASCAL: *Logique*, Paris, Hermès, 1990.

GÖDEL, KURT: "Sur les propositions formellement indécidables des Principia mathematica et des systèmes apparentés I", 1931, dans: NAGEL, ERNEST; NEWMAN, JAMES R.; GÖDEL, KURT.

HILBERT, DAVID: *Grundlagen der Geometric*, traduction française: *Les fondements de g*éométrie, Laugel, éd. critique de Paul Rossier, Paris, Dunond, 1971.

HILBERT, DAVID et BERNAYS, P.: *Fondements des mathématiques 1*, traduction de l'ouvrage *Grundlagen der Mathematik 1* (Springer), 2ème éd. (1968) avec les passages parallèles de la 1ère éd. (1934). Traduction de l'allemand par F. Gaillard et M. Guillaume. Ed. L'Harmattan, 2002, 2 vols.

HILBERT, DAVID: *Sur l'infini*, traduit par André Weil; "Über das unendliche: conférence prononcée le 4 Juin 1925 à l'occasion d'un congrès des mathématiciens organisé à Munster i. w. par la Société Mathématique de Wesphalie en l'honneur de la mémoire de Weierstrass". L'original de cette traduction a paru en allemand dans les Math. Ann. t 95.

HOTTOIS, GILBERT: *Penser la Logique: une introduction technique et théorique à la philosophie de la logique et langage*, 2ème ed., Bruxelles, De Boeck Université, 2002.

KLEENE, STEPHEN C.: *Introduction to metamathematics*, Amsterdam North Holland, 1952.

— *Logique mathématique*, traduction de Jean Largeault, Paris, Lib. Armand Collin, 1971.

LADRIERE, JEAN: *Les limitations internes des formalismes: Etude sur la signification du théorème de Gödel et des théorèmes apparentés dans la théorie des fondements des mathématiques*, Paris, Gauthier-Villars Ed., 1957. "Collection de Logique Mathématique", série B. Monographies réunies par M. R. Feys (Louvain).

LALEMENT, RENÉ: *Logique, r*éduction, *r*ésolution, Paris, Masson, 1990.

LARGEULT, JEAN: *La logique*, 2ème éd. corrigée, Paris, PUF, 1993.

LAURIER, DANIEL: *Introduction à la philosophie du langage*, Liège, Mardaga, 1993. *Philosophie et Langage*.

LEROUX, JEAN: *Introduction à la logique*, Diderot Multimedia, 1998. "Bibliothèque des Sciences".

LONERGAN, BERNARD J. F.: *Collected work of Bernard Lonergan. Phenomenology and logic the Boston College lectures on mathematical logic and existentialism*, Edited by Philip J. Mc Shane. Published for Lonergan Research Institute by University of Toronto Press, Toronto, Buffalo, London, 2001.

MARION, MATHIEU: *Ludwig Wittgenstein: introduction au Tractatus logico-philosophicus*, Paris, PUF, 2004.

MARTIN, ROGER: *Logique contemporaine et formalisation*, Paris, PUF, 1964.

MOSCONI, JEAN: "Calculabilité et formalisation", págs. 47-65, dans: *Le formalisme en question: le tournant des années trente*, Edité par Frédéric Nef et Denis Vernant, Paris, Vrin,

1998. (Problèmes et controverses: directeur Jean François Courtine).

— *La Constitution de la théorie des automates*. Thèse de Doctorat Histoire et Philosophie des Sciences, préparée sous la direction de Suzanne Bachelard, 1989, 2 vol.

Nagel, Ernest; Newman, James R.; Gödel, Kurt et Girard, Jean Yves: *Le théorème de Gödel*, traductions de l'anglais et de l'allemand par Jean Baptiste Scherrer, Paris, Editions du Seuil, 1989.

Piguet, Christian et Hugli, Heinz: *Du zéro à l´ordinateur. Une brève histoire du calcul*, Lausanne, Presses polytechniques et universitaires romandes, 2004.

Pradilla Rueda, Magdalena: *Vers une epistémologie de la théorie informatique*. Thèse pour obtenir le titre de Docteur en Philosophie, Paris, Université Paris I Panthéon-Sorbonne, 2008.

Poincaré, Henri: *La science et l'hypothèse*, Paris, Flammarion, 1968.

Putnam, Hilary: *Minds and machines*, Dimensions of mind: A symposium, Sydney Hook, ed., New York, University Press, New York, 1960. Trad. française, *Pensée et machine*, Anderson A. R., éd., Editions du Champ Vallon, Seyssel, 1983, págs. 110-134.

Quine, Willard van Orman: *Le mot et la chose*, traduit de l'américain par les professeurs Joseph Dopp et Paul Gochet, Paris, Flammarion, 1977.

Rivenc, François: *Introduction à la logique*, Paris, Payot, 1989.

Rivenc, François et Rouilhan, Philippe de: *Logique et fondements des mathématiques. Anthologie (1850-1914)*, Paris, Payot, 1992.

Rouilhan, Philippe de: *Frege les paradoxes de la représentation*, Paris, puf, 1996.

— *Russell et le cercle des paradoxes*, Paris, Les Editions de Minuit, 1988.

RUSSELL, BERTRAND and WHITEHEAD, ALFRED NORTH: *Principia mathematica*, 3 vols., Cambridge, University Press, 1910-1913; 2ª ed., 1925-1927.

SACKUR, JÉRÔME: *Opération et description. La critique par Wittgenstein des théories de la proposition de Russell*. Thèse pour obtenir le grade de Docteur de l'Université Paris 1, 2000.

SCHOLZ, HEINRICH: *Esquisse d'une histoire de la logique*, traduit de l'allemand par E. Coumet, Fr. de Laur, J. Sebestik, Paris, Aubier-Montagne, 1968.

SINACEUR, HOURYA et BOURGUINON, JEAN PIERRE: *David Hilbert et les mathématiques du XX* ième siècle, dans: *La Recherche*, 257, Sep. 1993, vol. 24, traduction en espagnol dans: *Mundo Cientifico*, 3(140): 936-943.

SOULEZ, ANTONIA: *Wittgenstein et le tournant grammatical*, Paris, PUF, 2004.

STERN, JACQUES: *Fondements mathématiques de l'informatique*, Paris, Ed. Science Internationale, 1990.

TARSKI, ALFRED: *Introduction à la logique*, trad. française de J. Tramblay, Paris, Gauthier-Villiers, 1971 (1ère éd. en polonais 1936).

— *Le concept de vérité dans les langages formalisés*, 1931; traduction française G. G. Granger, *Logique, sémantique, métamathématique*, Paris, Colin, 1972.

TURING, ALAN M.: "Computing machinery and intelligence", dans: *Mind*, vol. 59, núm. 236, 1950. Trad. française dans: *Sciences cognitives. Textes fondateurs (1943-1950)*, Paris, PUF, 1995.

— "On computable numbers, with application to the Entscheidungsproblem", dans: *Proceedings of the London Mathematical Society*, 1937. Traduction française: *Théo-*

rie des nombres calculables, suivie d'une application au problème de la décision. Traduit de l'anglais et annoté par Julien Basch, dans: *La machine de Turing*, Paris, Editions du Seuil, 1995.

VERNANT, DENIS: *Introduction à la philosophie de la logique*, Bruxelles, Pierre Mardaga Editeur, 1986.

WAGNER, PIERRE: *La machine en logique*, Paris, PUF, 1998 (*Science histoire et société*).

WATIER, GUILLAUME: *Le calcul confié aux machines*, Paris, Ellipses Editions Marketing.

WITTGENSTEIN, LUDWIG: *Grammaire philosophique*, edition posthume due aux soins de Rush Rhees. Traduit de l'allemand et présenté par Marie-Anne Lescourrent, Paris, Gallimard, 1980.

— *Remarques sur les fondements des mathématiques*, Paris, Gallimard, 1983.

— *Tractatus logico-philosophicus*, trad. de G. G. Granger, Paris, Gallimard, 1993. *Tractatus logico-philosophicus, suivi de investigations philosophiques*, trad. de l'allemand par Pierre Klossowski, Paris, Gallimard, 1961.

ANOTACIONES

ANOTACIONES

ANOTACIONES

ANOTACIONES

ANOTACIONES

Este libro se terminó de imprimir en los talleres de Nomos Impresores, el día veintiocho de julio de dos mil quince, aniversario del nacimiento de VICTOR MANUEL MAÚRTUA (n. 28, VII, 1867 y m. 26, V, 1937).

LABORE ET CONSTANTIA