Lógica, Matemática, Deducción Automática

por

Manuel Ojeda Aciego

Presentamos una breve perspectiva histórica del desarrollo en paralelo y, a veces, entrelazado, de la Lógica y las Matemáticas, con el objetivo final de presentar la Lógica Computacional y, en particular, la Deducción Automática, como un área de investigación matemática de extraordinario potencial práctico, no en balde distintos autores de conocido prestigio afirman que la Lógica es a la Computación lo que el Cálculo Infinitesimal es a la Física.

1 Introducción

La importancia de la Lógica viene siendo reconocida desde la antigüedad, ya los griegos clásicos sabían que el razonamiento es un proceso sujeto a ciertos esquemas y que, al menos parcialmente, está gobernado por leyes perfectamente formulables. Pero su importancia en la actualidad se debe, sin duda, al destacado papel que ha tomado recientemente en los más diversos campos de la Informática (análisis, síntesis y verificación de programas, programación lógica, inteligencia artificial, control de procesos, robótica, etc) y todo ello no de forma completamente accidental ya que, como veremos, la Lógica nació como un intento de mecanizar los procesos intelectivos del razonamiento.

En el caso que nos ocupa se suele establecer, generalmente, que la Lógica moderna se desarrolló a partir de la confluencia de Matemáticas, Ingeniería y Lingüística. Cuando se concretan referencias a la Ingeniería y a la Lingüística se percibe un aroma con el que todo matemático aplicado se siente identificado. Comentaremos brevemente estas dos disciplinas para, en el resto de este trabajo, centrarnos fundamentalmente en la aportación de las Matemáticas.

Podríamos situar el comienzo de la aportación de la Ingeniería a la Lógica en 1938, cuando Claude E. Shannon (más tarde famoso por su Teoría de la Información) observó que las funciones realizadas por circuitos combinatorios, inicialmente construidos con relés, se podían representar con la notación simbólica del álgebra de Boole [52]. A mediados de la década de los 50, D.A. Huffman extendió este trabajo a los circuitos secuenciales, lo cual dio origen al desarrollo de la teoría de máquinas de estados finitos [32].

La contribución de la Lingüística llega a finales de los 50. Noam Chomsky, con su teoría de las gramáticas formales, establece las bases de la *lingüística matemática* e inicia el camino hacia la formalización en la descripción de los lenguajes naturales [12]. Al mismo tiempo, se estaba trabajando en la especificación de la sintaxis de lenguajes de programación de ordenadores: Backus adaptó algunos trabajos de E. Post [45] a tales especificaciones en [3], y obtuvo una notación que era una variante de las gramáticas libres de contexto de Chomsky. Por otra parte, el estudio de las clases de lenguajes generados por las gramáticas formales y el estudio de las máquinas de estados finitos llevó al establecimiento de una relación inmediata y sorprendente: los mismos fenómenos aparecían de forma independiente en ambos campos, de manera que se podían establecer *isomorfismos* entre ambos modelos.

La implicación de la Lógica Matemática en el nacimiento de la Informática, y de la Lógica Computacional en su desarrollo actual, hace que el estudio de esta disciplina para un docente e investigador en Matemática Aplicada sea doblemente atractivo: por una parte, es atrayente la juventud del campo de estudio frente otras ramas tradicionales de las Matemáticas, por otra parte, sus orígenes resultan especialmente interesantes desde el punto de vista histórico.

En lo que sigue nos centraremos especialmente en las interrelaciones entre la historia de la Lógica y la de las Matemáticas y su confluencia en la creación de la Lógica Matemática y en los fundamentos de la Informática. Para empezar, y de modo ciertamente informal, podemos decir que la Lógica Matemática no es en absoluto necesaria (en el sentido de ciencia) si se pretenden mecanizar tareas tales como:

- Cálculos basados en operaciones aritméticas (que un humano puede memorizar y aplicar sin necesidad de razonar);
- Búsqueda de datos (por simple comparación con un patrón dado);
- Clasificación u ordenación de datos (siguiendo un criterio establecido);

pero, si lo que se pretende es mecanizar tareas en las que interviene destacadamente la capacidad deductiva, que podemos calificar como «inteligentes», en las que se requiere:

- Tener conocimiento sobre el dominio del discurso;
- Razonar con tal conocimiento;
- Conocer cómo dirigir o guiar tal razonamiento;

entonces es preciso definir con claridad y precisión, así como analizar desde el punto de vista matemático, los procesos deductivos que el hombre ejercita de modo natural. Tal es el objetivo de la Lógica.

La estructura de este trabajo es la siguiente: en la próxima sección se describe un símil comúnmente usado entre las ventajas aportadas por la introducción del Cálculo Infinitesimal desde el punto de vista de la mecanización del estudio de fenómenos físicos y la introducción de la Lógica Computacional desde el punto de vista de la mecanización del estudio de gestión de la información; las secciones siguientes presentan distintas etapas históricas en el desarrollo de la Lógica, comenzando por la silogística aristotélica, haciendo hincapié precisamente en los avances obtenidos gracias a la interacción con las Matemáticas. Finalmente, se dedica una sección a presentar de modo informal el contenido matemático del área de investigación de la Lógica Computacional conocida como Deducción Automática.

2 De Euclides y Arquímedes a Newton y Leibniz

Reafirmando lo indicado en el resumen, algunos autores han afirmado que la Lógica es a la Computación como el Cálculo es a la Física. Por esta razón presentamos una brevísima visión de la importancia del Cálculo como herramienta para la mecanización del conocimiento de fenómenos físicos para, en las secciones siguientes, esbozar la historia de la Lógica Matemática y su aportación a la Informática actual.

En el desarrollo histórico de las ciencias matemáticas se pueden distinguir varias etapas aunque, sin duda, no es hasta la civilización griega cuando las Matemáticas aparecen como una disciplina «formal». En contraste con sus antecesores, los griegos tuvieron la originalidad de hacer un esfuerzo considerable para que sus demostraciones estuvieran fuera de toda duda respecto a su verosimilitud. El origen de la lógica formal puede centrarse, asimismo, en este periodo histórico.

Durante la época de esplendor de la Grecia Clásica, Platón (c. 428–348 a.C.) hizo de la Geometría un requisito imprescindible para entrar a su Academia. Es bien conocido el lema de su Academia «*Nadie pase sin saber Geometría*», pues de un experto en geometría se suponía, como el valor a un soldado, la habilidad para razonar con corrección y exactitud. No hay dudas de que Euclides de Alejandría (c.330–c.275 a.C.) fue el impulsor definitivo del método axiomático en Geometría: en sus *Elementos*, Euclides agrupó las derivaciones de la escuela Pitagórica y las de muchos otros en un todo unificado. Los *Elementos* proporcionaron un modelo para todos los subsiguientes trabajos matemáticos, y representan el principio de la matemática moderna (forman el primer sistema formal de la matemática), que ha estado en uso durante más de dos mil años.

Arquímedes (287–212 a.C.), entre otros, mostró cómo usar la geometría sintética para calcular áreas y volúmenes de muchas figuras y sólidos simples. También resolvió geométricamente muchos problemas de mecánica, hidrostática y óptica. En sus trabajos se puede observar cómo surgen problemas de índole matemática a partir de los esfuerzos científicos para extraer las leyes de la naturaleza. Por su parte, parece ser que la motivación principal de Euclides al desarrollar su geometría fue fundamentalmente artística, es decir, el placer estético. Se intuye, pues, la existencia de dos enfoques de las Matemáticas: uno encaminado a objetivos eminentemente prácticos (¿*matemática aplicada*?) y otro cuya motivación es de carácter puramente formal alejada de todo pragmatismo (¿*matemática pura*?). Veremos más adelante, sin embargo, que precisamente el estudio formal del método axiomático, y sus connotaciones respecto de los fundamentos de las Matemáticas, fue el detonante del estudio de la computabilidad y, finalmente, del desarrollo de la Informática.

En los veinte siglos que separan a Euclides y Arquímedes de Newton (1640–1722) y Leibniz (1640–1710), se resolvieron problemas de dificultad creciente en distintas disciplinas físicas, cada uno de los cuales necesitó de métodos desarrollados *ad hoc*. Cada avance en Física o Matemáticas conseguido con el método geométrico requería el extraordinario talento matemático de, por ejemplo, una figura de la talla de Galileo (1564–1642).

Las cosas cambiaron completamente después de que Descartes (1596–1650) descubriera que los problemas geométricos se podían traducir a problemas algebraicos. De este modo los métodos geométricos fueron reemplazados por cálculos algebraicos.

Ya había indicios de la aplicación de métodos proto-algebraicos de integración y diferenciación en los trabajos de Fermat (1601–1665), Barrow (1630–1677), que fue profesor de Newton, y Cavalieri (1598–1647), que fue predecesor de Leibniz. Los métodos simbólicos de diferenciación e integración descubiertos por Newton y Leibniz hicieron posible que las sucesivas generaciones usaran cálculos ordinarios para desarrollar la ciencia y la ingeniería sin necesidad de ideas (geométricas) felices. Estos métodos aún están en la base de la comprensión, modelado, simulado, diseño y desarrollo de sistemas físicos.

En otro orden de cosas, Leibniz fue el primero en afirmar la posible existencia de algo equivalente a una lógica formal completa para describir el razonamiento [42]. No estaba satisfecho con la lógica aristotélica y desarrolló sus propias ideas para mejorarla. Estaba convencido de que podría desarrollar un lenguaje para, y un cálculo de, los razonamientos que sería tan importante como el cálculo desarrollado por él mismo y Newton para las derivadas y las integrales.

Llamó *lingua characteristica* a este nuevo lenguaje y *calculus ratiocinator* al esperado cálculo, con el cual, la mente «será liberada de tener que pensar en las cosas en sí mismas, y aún así todo funcionará perfectamente».

Leibniz esperaba que estas nuevas ideas expandirían la capacidad de razonamiento mediante la reducción a un cálculo simbólico de mucha de la labor necesaria para descubrir cómo obtener determinada conclusión a partir de unas premisas dadas, y cómo comprobar la corrección de una deducción. Dicho de otro modo, esperaba la existencia de un *cálculo* análogo al cálculo infinitesimal, pero un cálculo de razonamientos para tratar las deducciones con proposiciones.

Pese al «sueño de Leibniz», la Lógica Matemática se iba gestando más lentamente; el desarrollo de las ideas, notación y formalismos adecuados para la obtención de conceptos similares al cálculo diferencial, en cuanto a potencia, para la Lógica necesitó varios siglos más. Podemos hacer un símil con la división habitual de periodos históricos, aunque es preciso notar que no coinciden temporalmente con los periodos homónimos de la Historia Universal, y así dividir el desarrollo de la Lógica en una Edad Antigua, que se corresponde con la Lógica Tradicional (500 a.C–1847); una Edad Media, con el desarrollo de la Lógica Simbólica (1847–1880); una Edad Moderna, en la que se introduce la Lógica Matemática de manera formal (1880–1960); y una Edad

Contemporánea, en la que surge la Lógica Computacional (desde 1960 a la actualidad).

3 La Edad Antigua de la Lógica

El embrión de la lógica moderna no es otro que la teoría silogística de Aristóteles (384–322 a.C.), que se ha enseñado como parte del *Trivium* (Gramática, Retórica y Dialéctica) desde la Edad Media hasta principios del siglo XX. La silogística de Aristóteles fue el primer cálculo de razonamientos con los cuantificadores «todos» y «algunos» que, usando terminología moderna, traducimos como los cuantificadores universal ($\forall$) y existencial ($\exists$), respectivamente.

En sentido aristotélico, el mundo consta de objetos c que pueden o no tener una propiedad dada P. Una interpretación formal de P se consigue mediante la especificación de un dominio no vacío C de objetos y un subconjunto de objetos que resultan denotados por P. De este modo, si x es una variable que se mueve en C entonces $P(x)$ es una fórmula lógica que se lee «x tiene la propiedad P».

La función principal de la silogística era comprobar que los cuantificadores «para todo» y «existe» se usan correctamente en una argumentación. La meta era la eliminación de argumentos incorrectos que usan principios que parecen lógicamente válidos pero no lo son.

Brevemente presentamos el objeto de estudio de la silogística como un conjunto de reglas de inferencia entre los siguientes cuatro tipos de proposiciones, llamadas *proposiciones categóricas*, cuyos nombres medievales fueron A, E, I y O[1].

(A)	Todo P es Q	(E)	Ningún P es Q
(I)	Algún P es Q	(O)	Algún P no es Q

La forma general de considerar un silogismo era la de tomar dos premisas con un término común (el *término medio*) que permitiera tender un puente entre ellas, para así poder deducir alguna consecuencia. De este modo obtenemos cuatro posibles *figuras* para un silogismo:

Premisa Mayor	M—P	P—M	M—P	P—M
Premisa Menor	S—M	S—M	M—S	M—S
Conclusión	S—P	S—P	S—P	S—P
	(Primera)	(Segunda)	(Tercera)	(Cuarta)

[1]Los nombres de las proposiciones categóricas son mnemotécnicos tomados de las palabras latinas *AffIrmo* y *nEgO* (las proposiciones de tipo A e I son afirmativas, mientras que las de tipo E y O son negativas). Es conveniente notar que el Diccionario de la Real Academia Española introduce estas acepciones para 'a', 'e', 'i' y 'o' desde su edición de 1899.

De cada figura podemos construir $4^3 = 64$ *modos* distintos, pues en cada premisa y en la conclusión podemos colocar una proposición de tipo A, E, I u O. De todos estos modos, obviamente, sólo algunos son válidos.

Un ejemplo de silogismo válido de la primera figura es el llamado «silogismo perfecto»

$$\frac{\begin{array}{c}\text{Todos los } Q \text{ son } R\\ \text{Todos los } P \text{ son } Q\end{array}}{\text{Todos los } P \text{ son } R}$$

En la notación lógica contemporánea escribimos $P(x)$ para representar «x es un P», $Q(x)$ para representar «x es un Q», $R(x)$ para representar «x es un R», $\forall x$ para «todo x», $\exists x$ para «existe x» y $\to$ para la relación de implicación. En notación moderna el modo anterior se convierte en

$$\frac{\begin{array}{c}\forall x(Q(x) \to R(x))\\ \forall x(P(x) \to Q(x))\end{array}}{\forall x(P(x) \to R(x))}$$

Aunque la silogística aristotélica fue útil para clarificar discusiones filosóficas, no tuvo demasiada influencia para los matemáticos, ya que éstos razonaban de modo perfectamente exacto antes de Aristóteles; de hecho, su trabajo era el modelo tradicional de razonamiento correcto. El problema estaba en que el método de razonamiento matemático habitual no estaba completamente descrito por la silogística. ¿Qué es lo que faltaba? Faltaba una forma de lógica que tomase enunciados que consideramos evidentemente ciertos y que construyese nuevos enunciados más complicados pero igualmente verdaderos, usando ciertas reglas bien definidas. Con la perspectiva actual, una respuesta concisa es: faltaba la noción de relación con varios argumentos, así como el resto de la lógica proposicional.

Relaciones de más de un argumento

Aristóteles no estudió la componente básica del lenguaje lógico de las relaciones binarias, ternarias, etc., pues sólo usó predicados monádicos del tipo $P(x)$ y, para codificar relaciones $S(x, y)$ como «x es el abuelo de y» escribía $S_y(x)$ para representar que «x tiene la propiedad de ser el abuelo de y». No había un defecto real en la teoría aristotélica de cuantificadores; simplemente sus fórmulas lógicas adolecían de conectivas proposicionales y de predicados con varios argumentos; nótese que desde el principio mismo de los *Elementos* de Euclides se comienza a trabajar con la relación de incidencia $R(x, y)$, que significa x es incidente con y donde x e y son puntos, rectas o planos. Esta laguna fue cubierta por autores bien conocidos de finales del siglo XIX como Schröder y Frege, de los que hablaremos más adelante.

Lógica proposicional

Por lo que se refiere a la lógica proposicional, Crisipo de Soli (c.281–206 a.C.) introdujo las conectivas de implicación, conjunción y disyunción exclusiva, y extrajo la propiedad característica de la lógica proposicional: el valor de verdad o falsedad de proposiciones compuesta a partir de estas conectivas está determinado por el conocimiento de la verdad o falsedad de sus partes. De esta forma llevó a los estoicos a su periodo de mayor relevancia a finales del siglo III a.C.

Los estoicos disponían de todos los ingredientes para acometer una teoría lógica de la demostración matemática. Sin embargo, con la muerte de Crisipo la escuela estoica pareció colapsarse, sus trabajos se perdieron, y la Lógica volvió a reducirse al estudio de la silogística aristotélica durante más de dos mil años.

Cuando finalmente fueron descubiertos algunos fragmentos de obras de los estoicos, fueron tratados con desdén. Algunos tratados de historia de la lógica antigua describen la tabla de verdad de la implicación como *excesivamente estúpida*. Sin embargo, como se puede leer en [39], «...de debajo del polvo de dos mil años, el genio de los estoicos volvió para fertilizar el nacimiento de la era de los ordenadores: sus excesivamente estúpidas tablas de verdad, embutidas en silicona, forman las mentes de nuestros, hoy por hoy, imprescindibles computadores».

4 La Edad Media de la Lógica

Decíamos que hubo que esperar más de dos mil años para que se volviesen a conocer las tablas de verdad. Más concretamente, el álgebra de clases y la lógica de proposiciones fueron descubiertas de nuevo y desarrolladas mucho más completamente a mediados del siglo XIX por Augustus de Morgan (1806–1871) y George Boole (1815–1864), véase [21, 10]. La contribución de A. de Morgan representó una extensión de la Silogística, introdujo conectivas proposicionales y sus leyes, así como una rudimentaria teoría de relaciones. Por su parte, G. Boole (re-)descubrió las tablas de verdad para las proposiciones y la forma normal disyuntiva (disyunción de conjunciones de literales), que introdujo con el nombre de la «ley de expansión». Fue él quien desarrolló un razonamiento sistemático de la lógica de las proposiciones basado en el álgebra pura, cuyo trabajo llevó más tarde a lo que se conoce como el *álgebra de la lógica*. Usando términos modernos, Boole interpretó las fórmulas lógicas con n proposiciones atómicas como funciones ${\mathbb{Z}_2}^n \longrightarrow \mathbb{Z}_2$. Las conectivas proposicionales fueron interpretadas como meras funciones binarias. Boole trabajó con tres conectivos $\wedge$, $\vee$ y $\neg$, cuya interpretación es la siguiente, para dos proposiciones cualesquiera p y q:

- $p \vee q$ es verdadera si y solo si al menos una p o q es verdadera.

- $p \wedge q$ es verdadera si y solo si ambas p y q son verdaderas.
- $\neg p$ es verdadera si y solo si p no es verdadera.

Formalmente, las funciones asociadas a los conectivos son las siguientes:

	$\neg$
0	1
1	0

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

De este modo, una proposición se puede interpretar como una función con rango en un conjunto binario (verdadero-falso, 0-1, etc.) en los valores de sus símbolos proposicionales, lo cual no es más que la idea de Crisipo en versión moderna.

En el álgebra usual sobre los números reales o complejos, el álgebra de los polinomios formales y el álgebra de las funciones polinómicas son virtualmente indistinguibles, con más propiedad diríamos *isomorfas*. Siguiendo la identificación indicada para los polinomios, es natural que Boole identificara el significado de una proposición con la función proposicional bivaluada inducida. Bajo esta identificación, las operaciones lógicas «y», «o», y «no» se corresponden con operaciones sobre las funciones proposicionales. Esto convierte el conjunto de proposiciones en una estructura algebraica donde dos proposiciones son iguales si denotan la misma función proposicional; por ejemplo, dadas dos proposiciones α y β, tenemos $\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$ (una de las leyes de de Morgan), que es una de las leyes de las álgebras de Boole.

Asimismo, Boole observó la relación entre los cuantificadores universal y existencial y las cotas superiores mínimas y las cotas inferiores máximas (respectivamente) e introdujo una notación algebraica para los mismos, conceptos que fueron formalizados y extensamente desarrollados por Schröder [50, 51], pero no tuvo una buena teoría de la cuantificación como tal. No mejoró el tratamiento aristotélico de los cuantificadores excepto para notar que también son operaciones sobre las funciones proposicionales.

5 La Edad Moderna de la Lógica

El siglo XIX fue testigo de un esfuerzo concertado para desarrollar una base firme sobre la que fundamentar las matemáticas, con definiciones precisas, axiomas y construcciones. La imprecisión de las definiciones provocaba confusión y controversia. Difícilmente se distinguía una función de su representación simbólica; la continuidad de la continuidad uniforme, etc. El mismo Cauchy, uno de los grandes defensores del rigor en el análisis, dio una «demostración» de que la suma de una serie infinita de funciones continuas es continua, a la que en 1826 Abel dio un contraejemplo.

Hemos de retrasar nuestro reloj unos dos mil quinientos años para recordar la elegancia del planteamiento de los *Elementos* de Euclides, la geometría sintética se tomó como el fundamento lógico de las matemáticas: cada campo de las matemáticas debía, pues, ser reducido a la geometría. Con la introducción en el siglo XVII de los sistemas coordenados, Descartes redujo la geometría sintética a la geometría analítica y, por lo tanto, todo quedó reducido al álgebra y a la aritmética. Este paso llevó al gran esfuerzo para definir cualquier estructura matemática compleja en términos de otras estructuras más simples. Finalmente, Dedekind [22] y Peano [43, 44] como bien sabemos, mostraron que incluso los naturales se pueden construir a partir únicamente de un conjunto unitario (el 0 ó el 1) y una función sucesor $S(x)$.

Para llevar a cabo de manera formal esta aritmetización se necesitaba completar una laguna que podemos dividir en dos frentes:

1. Si todas las propiedades de todos los sistemas se deducen mediante inferencias lógicas, primero hay que plantearse una definición formal del lenguaje lógico que se utilizará y luego hay que decidir sus conjuntos de axiomas y de reglas de inferencia.

2. Por otra parte, si cada sistema se define en términos de sistemas más simples mediante construcciones conjuntistas, junto con postulados de existencia de algunos conjuntos para poder empezar el trabajo, ¿qué conjuntos merecen existir por convenio?, ¿cuáles son las construcciones conjuntistas necesarias?, ¿cuáles son los axiomas de la teoría de conjuntos?

5.1 La crisis de los fundamentos

La introducción del concepto de lenguaje formal como objeto de estudio claramente diferenciado del lenguaje natural (o *metalenguaje*) usado para expresar sus propiedades fue adecuadamente tratado por Gottlob Frege (1848–1925). En [23] dio el primer tratamiento formal de la lógica de predicados incluyendo tanto los cuantificadores y las relaciones como las conectivas proposicionales; combinando el tratamiento aristotélico de los cuantificadores y el tratamiento booleano de las conectivas proposicionales y los símbolos de relación R de cualquier número de argumentos, definió la noción de fórmula lógica construida a partir de fórmulas atómicas de la forma $R(x_1, \ldots, x_n)$ mediante las conectivas proposicionales y los cuantificadores.

Frege dio, también por vez primera, un conjunto de axiomas y reglas de inferencia, así como la definición de *demostración* como una secuencia finita de fórmulas, cada una de las cuales es o un axioma o se deduce de las fórmulas anteriores tras la aplicación de una regla de inferencia. La introducción de la base actual de la semántica de la lógica de predicados fue fruto de Ernst Schröder (1841–1902)[2].

[2]Se presenta así un doble enfoque para el análisis de enunciados formales: un enfoque sintáctico (interno), en el que se intenta demostrar el enunciado usando los axiomas y reglas

Con la introducción de los lenguajes de primer orden, la lógica da lugar a expectativas sobre un gran deseo, ampliamente compartido, que solo David Hilbert (1862–1943) se atrevería a expresar explícitamente en 1900 en el Congreso Internacional de Matemáticos. En su lista de 23 problemas para los matemáticos del siglo veinte, propone como segundo problema el uso del sistema de Frege para expresar toda la matemática y probar su consistencia, es decir, que es imposible obtener contradicciones, como por ejemplo $0 = 1$ ó $0 \neq 0$.

Frege también presentó en sus *Grundgesetze der Arithmetik* [24] un fundamento formal común tanto para la lógica como para las matemáticas. La idea fundamental de Frege era que *toda* propiedad que pudiese ser formulada en el lenguaje formal sería aceptable como un símbolo de predicado P. Un conjunto de objetos A sería admisible también como un objeto, que podría ser a su vez miembro de otro conjunto. El sistema de Frege era seductivamente simple. No hay más que escribir una descripción de nuestro conjunto favorito para asegurar su existencia y unicidad (en base a los axiomas de comprensión y extensionalidad).

Pero he aquí que, mientras el segundo volumen de sus *Grundgesetze* estaba ya en imprenta, la paradoja de Russell hizo su aparición:

> Supongamos que $P(x)$ representa la propiedad $\neg(x \in x)$; y consideremos el conjunto $A = \{x \mid P(x)\}$. Al aplicar esta definición al propio A como elemento se obtiene que $A \in A$ si y solo si $A \notin A$, una flagrante contradicción[3].

Toda la estructura elaborada por Frege para la fundamentación lógica de las matemáticas se colapsó en una gran contradicción justo cuando la obra estaba terminada.

Por otra parte, tras la introducción por Georg Cantor (1845–1918) de la noción de conjunto como objeto semántico unificador [11], algunos matemáticos de renombre, entre ellos Hilbert, pensaron que iba a ser el concepto primitivo que permitiría expresar formalmente todas las Matemáticas. No todo el mundo matemático aceptó de buen grado las teorías de Cantor, especialmente sus números transfinitos, entre sus detractores destaca Leopold Kronecker (1823–1891), promotor del *contructivismo* en las demostraciones matemáticas, que argumentó especialmente la imposibilidad de «construir» conjuntos infinitos.

La aparición de paradojas en la teoría intuitiva de conjuntos y, muy especialmente, la paradoja de Russell en el sistema de Frege, proyectaron una oscura sombra de duda sobre los fundamentos de las Matemáticas, lo que repercutía en las dudas acerca de su consistencia.

---

de inferencia del sistema, y un enfoque semántico (externo) en el que se estudia la validez desde fuera del sistema. La coincidencia de ambos enfoques se formaliza mediante los conceptos de corrección y completitud.

[3] En la bibliografía se sigue usando el eufemismo *paradoja*, en lugar de antinomia o contradicción.

Estas paradojas condujeron a Bertrand Russell (1872–1970) y a Alfred N. Whitehead (1861–1947) a refinar el sistema de Frege mediante un lenguaje estratificado de objetos con tipos, y a proponer el desarrollo de la formalización de las matemáticas en sus *Principia Mathematica* [57]. Este enfoque se denominó *corriente logicista* de los fundamentos matemáticos.

Por su parte, Hilbert dio el primer conjunto completo de axiomas para la geometría euclídea y, en la década de 1920, aplicó dicha filosofía a la mismísima lógica de las matemáticas, introduciendo la *corriente formalista* que, fundamentalmente, pretendía eliminar la posible interpretación de los símbolos matemáticos usados, y de este modo evitar el uso de la intuición en el repertorio de herramientas matemáticas.

Ahora bien, si los axiomas elegidos no están basados en interpretaciones sensibles, difícilmente se puede alegar su carácter de «evidentes», de modo que surge la necesidad de demostrar que el conjunto de axiomas con el que se trabaja es consistente. En su obra conjunta con Wilhem Ackermann [31], la lógica de predicados fue separada como objeto propio de estudio y se hizo hincapié especialmente en las demostraciones de consistencia y de decidibilidad, es decir, la existencia de algoritmos que permitiesen decidir de manera automática la validez de las fórmulas de la lógica de predicados[4]. Esta empresa dio lugar a la introducción de la *metamatemática*, el estudio matemático de los métodos y razonamientos de las Matemáticas. Esta rama es conocida en la actualidad como *teoría de la demostración*.

Lamentablemente, el programa de Hilbert terminó de una forma completamente inesperada, ya que Kurt Gödel (1906–1978) lo refutaría definitivamente demostrando su teorema de incompletitud: *en todo sistema formal consistente capaz de contener la aritmética existen fórmulas verdaderas que no pueden ser demostradas en el sistema* [27]. Como consecuencia se obtiene que si el sistema de las Matemáticas es consistente (ni siquiera pensemos las consecuencias de su inconsistencia) entonces existen enunciados verdaderos que no admiten demostración.

Por otro lado, el propio Gödel demostró la completitud de sistemas de menor potencia expresiva tales como la lógica de predicados de primer orden; específicamente, toda fórmula válida desde el punto de vista externo admite asimismo una demostración en el sistema. Puesto que una parte importante de las Matemáticas puede ser expresada en lógica de primer orden, el *Entscheidungsproblem* de Hilbert seguía teniendo vigencia (aunque de modo parcial) tras el teorema de completitud, de modo que ¿qué se puede decir acerca de la posibilidad de automatización del razonamiento en la lógica de primer orden? La búsqueda de respuestas a esta pregunta pasa necesariamente por un estudio detallado del concepto de algoritmo o, lo que es lo mismo, de función computable.

[4]Este problema se conoce con su nombre original alemán, *Entscheidungsproblem*

Antes de pasar a la formalización de la computabilidad, y por completar la presentación de las distintas corrientes de pensamiento surgidas durante la crisis de los fundamentos, es de justicia dedicar al menos un párrafo a la escuela de los *intuicionistas*. Éstos consideraban que el formalismo podría implicar una mecanización excesiva del trabajo matemático; en palabras de Poincaré según el formalismo «se podría imaginar una máquina en la que se pondrían axiomas por una parte, y por la otra se recogerían los teoremas que se siguen de los axiomas». El principal impulsor del intuicionismo fue L.E.J. Brouwer (1881–1966), para el que las Matemáticas existen como intuición en la conciencia de los matemáticos, y no en el lenguaje usado para transmitirlas. Respecto de la existencia de los objetos matemáticos, Brouwer afirmó « ... *la existencia* en Matemáticas significa: poder ser construido por la intuición; y la pregunta de si cierto lenguaje es consistente, no solo no es banal en sí misma, sino que no sirve como test para la existencia matemática». Esta filosofía de las Matemáticas se encuentra en oposición frontal tanto con el formalismo y como con el logicismo, estando más en la línea del constructivismo de Kronecker.

5.2 La formalización del concepto de computabilidad

Al hablar de algoritmo nos solemos referir a un conjunto de instrucciones efectivo y explícito que permita hallar la solución de un problema. Pero ésta es una definición informal. La formalización de esta noción intuitiva es un requisito imprescindible para que se puedan plantear y resolver matemáticamente cuestiones básicas como qué es un problema insoluble algorítmicamente. Cuando hablamos de problemas nos referimos a problemas abstractos o generales, tales como «sumar dos números» frente a problemas concretos tales como «sumar 3 y 15». Tales problemas abstractos pueden considerarse como clases de problemas concretos.

Un problema abstracto es resoluble algorítmicamente si es posible construir un algoritmo que resuelva cada problema concreto de la clase. Los problemas abstractos se clasifican en decidibles e indecidibles (resp. resolubles e irresolubles algorítmicamente). Para probar la decidibilidad de un problema basta con dar un algoritmo, el que sea, que lo resuelva; para la indecidibilidad se requiere una especificación formal de la noción de algoritmo, de manera que se demuestre que ningún algoritmo, construido o no, la podría resolver. La reducción del concepto de algoritmo a la noción matemática de recursividad o calculabilidad ha sido fundamental para abordar esta cuestión.

El análisis matemático de la recursividad pasa por una investigación con profundidad de la noción de función. En la década de 1920, los matemáticos ya habían dedicado esfuerzos a entender la noción de función. La visión tradicional de una función como conjunto ordenado de pares con dominio y rango fijo no describe adecuadamente el comportamiento de las funciones. La idea clave, ya presente en los trabajos de Frege, fue que era suficiente estudiar funciones de un solo argumento: Toda función de la forma $f : A \times B \longrightarrow C$ podía ser reemplazada por una función $f^* : A \longrightarrow (B \longrightarrow C)$ que, dado su primer argumento, produce otra función a la espera de aceptar el segundo argumento. No ha lugar pues a distinción entre funciones, funciones de orden superior, etc. Surge así el λ-cálculo, introducido por Alonzo Church (1903–1995), como una teoría en la que las funciones vuelven a ser consideradas reglas, en lugar de seguir la formalización tradicional de Dedekind que considera las funciones como tipos particulares de grafos.

Church y sus seguidores establecieron las propiedades teóricas básicas de este formalismo en los años treinta. Conceptos como *variable, sustitución, ocurrencias libres y ligadas* de una variable en una expresión, *renombramiento*, etc, tan alegremente usadas en los formalismos matemáticos, se convierten en protagonistas de elite sobre los que recae la responsabilidad de cualquier progreso esencial en matemáticas [16].

El λ-cálculo presentó un enfoque de la formalización de «lo calculable» ciertamente prometedor, en el que ciertas propiedades como la terminación y la confluencia tienen un papel comparable a los principios fundamentales de la Física. Lamentablemente, apenas existe una conexión entre la forma en que Church estableció sus ideas y cualquier cosa que pudiéramos llamar «mecánica». La idea clave subyacente en el λ-cálculo es esencialmente abstracta: de hecho, la clave del mismo es una operación matemática que el propio Church llama *abstracción*. Stephen Kleene, discípulo de Church, comprobó que todas las funciones numéricas que podía «calcular» eran λ-definibles, es decir, definibles usando el λ-cálculo. Esto animó a Church a considerar la λ-definibilidad como una buena noción de *calculabilidad*.

No todo el entorno científico compartía en enfoque de Church y Kleene basado en el λ-cálculo. En particular, Gödel trabajaba, sobre una sugerencia de Jacques Herbrand (1908–1931), para introducir una definición bastante más manejable de computabilidad: su trabajo inicial culminó con la definición de las funciones *recursivas generales*.

Pese a que aparentemente podrían parecer nociones diferentes, Kleene demostró en 1936 que las definiciones de Church y Herbrand-Gödel son equivalentes. Por su parte, Church demostró en ese mismo año que ninguna función recursiva podía decidir la validez de fórmulas de primer orden, demostrando, en consecuencia, la inexistencia de algoritmos para decidir la validez de las fórmulas de la lógica de primer orden.

Desde un punto de vista mecánico, ni las funciones λ-definibles, ni las funciones recursivas generales constituyen modelos intrínsecos «convincentes» de la calculabilidad. Para tal fin Alan Turing (1912–1954) propuso un nuevo modelo de calculabilidad: sus máquinas abstractas, actualmente llamadas *máquinas de Turing*, le permitieron introducir la noción de función computable por una máquina abstracta. Turing fue capaz de probar en 1936, independientemente de Church, que no podía existir un algoritmo para examinar la validez de inferencias en la lógica de primer orden. Posteriormente demostró que las funciones recursivas coincidían con las calculables en sus máquinas abstractas, lo cual vino a corroborar la siguiente afirmación:

> Todo proceso algorítmico define una función matemática caracterizada por las siguientes condiciones equivalentes:
>
> 1. Recursividad general.
> 2. λ-definibilidad.
> 3. Calculabilidad en una máquina de Turing.

En la práctica, no sólo se trabaja con la equivalencia de estas condiciones, sino que se conjetura que cualquier noción de calculabilidad que se pueda construir será equivalente a cualquiera de las anteriores. Esta conjetura se conoce como la *Tesis de Church*.

5.3 El germen de la deducción automática

El desarrollo de los métodos estrictamente lógicos siguió su curso en paralelo a las convulsiones producidas en los fundamentos de las matemáticas.

Siguiendo los *Principia Mathematica* de Whitehead y Russell, Emil Post en [45] dio una definición de demostrabilidad formal para la lógica proposicional, junto con los teoremas de corrección y completitud. Por su parte, el desarrollo del cálculo de predicados hasta un punto semejante necesitó algo más de tiempo.

Thoralf Skolem observó que toda fórmula es equisatisfacible[5] con una clase especial de forma normal prenexa (aquella que tiene todos los cuantificadores al principio de la fórmula y ninguno de ellos es $\exists$). Su idea es bastante común en el quehacer habitual de un matemático: supongamos que tenemos una fórmula lógica F cuya expresión prenexa es de la forma $\forall x \exists y \phi(x, y)$. Si F es verdadera en un dominio D entonces para cada $x \in D$ existe un elemento y tal que se verifica la relación expresada por $\phi(x, y)$. Este procedimiento nos permite definir una función, llamémosla f, tal que $y = f(x)$ para todo x. De este modo, podemos eliminar el cuantificador existencial y considerar simplemente $\forall x \phi(x, f(x))$. Este procedimiento hace hincapié en la utilidad de considerar símbolos de función y nombres de elementos individuales de un dominio.

El crédito de haber dado la primera prueba de la completitud (toda fórmula verdadera admite una demostración en el sistema) para la lógica de predicados lo tiene Gödel [26] según hemos indicado con anterioridad. Por su parte, Jacques Herbrand tuvo a su alcance todos los ingredientes para construir una demostración del teorema de completitud [30]. Asoció a cada fórmula de la lógica de predicados $\neg\phi$ una sucesión ψ_n de fórmulas de la lógica *proposicional* y demostró que $\neg\phi$ es demostrable si y solo si existe un n tal que $\psi_1 \vee \cdots \vee \psi_n$ es una fórmula válida. Por lo tanto, si $\neg\psi$ no es demostrable entonces existe, para cada n, un modelo proposicional para $\neg\psi_1 \wedge \cdots \wedge \neg\psi_n$. Su trabajo también

[5]Dos fórmulas son equisatisfacibles si una tiene un modelo si y solo si la otra también lo tiene (no necesariamente el mismo).

demostraba cómo obtener un modelo de $\neg\phi$ a partir de estos modelos proposicionales, y así poder demostrar el teorema de completitud. Lo que Herbrand se negó a hacer fue usar un argumento no constructivo (como el teorema de compacidad) para producir, a partir de los modelos individuales de cada conjunción, un modelo común para todas ellas y para $\neg\phi$, con lo que probar que ϕ no es válida.

La extrema insistencia de Herbrand en el constructivismo fue, sin embargo, la causa de la importancia de su trabajo. El teorema de Herbrand es el resultado más importante de la historia de la deducción automática en lógica clásica, al establecer un puente entre el cálculo de predicados y el cálculo proposicional. Su interés radica en que se puede comprobar la validez de una fórmula cuantificada haciéndolo sucesivamente en distintas instancias proposicionales.

6 La Edad Contemporánea de la Lógica

Con trabajos de grandes matemáticos, la Lógica tomó cuerpo como potente herramienta matemática. Sus trabajos dieron lugar a una disciplina referida de forma habitual como *Lógica Computacional*, en cuya concepción han intervenido, como hemos visto, al menos tres grupos distintos con muy diferentes motivaciones.

Sin lugar a dudas, la investigación acerca de las propiedades computacionales de la Lógica es la que, en mayor medida, goza de las características más atractivas para un matemático, ya que considera los aspectos más profundos del concepto de computación, permite plantearse problemas de la más diversa dificultad, toda herramienta matemática le es útil, admite multitud de visiones mutuamente enriquecedoras y, además, exige el constante ir y venir de la abstracción a la realidad.

En la actualidad la Lógica Computacional se puede dividir en distintas subespecialidades, cada una de ellas con importantes retos tanto desde el punto de vista de sus fundamentos matemáticos, como desde el de las aplicaciones prácticas que conllevan. Una de estas especialidades es la deducción automática, cuyo desarrollo y metas describiremos con detalle en la próxima sección; describimos someramente a continuación algunos otros tópicos de interés dentro de la Lógica Computacional.

Creación de nuevas lógicas y métodos formales. Conforme la Informática ha ido evolucionando, en paralelo ha ido creciendo la necesidad de considerar (o crear) lógicas no clásicas que permitan modelizar distintos aspectos del razonamiento humano o distintas aplicaciones; por ejemplo, el control de procesos, que involucra modalidades como tiempo, conocimiento y creencia, es hoy una de las líneas de mayor interés y que reclama mayores esfuerzos de investigación.

LENGUAJES LÓGICOS DE PROGRAMACIÓN. En este punto se incluye el estudio de las relaciones entre la lógica y los lenguajes de programación, incluyendo los fundamentos lógico-matemáticos de los lenguajes de programación (como el uso de la lógica como base para definir un lenguaje de programación, o para proporcionar la semántica de los programas), y la lógica como herramienta matemática para expresar características de los programas, como en el paradigma de programación lógica con restricciones.

LÓGICA PARA EL PROCESAMIENTO DE INFORMACIÓN. En esta área se pueden destacar varias ramas en las que interviene la lógica, por ejemplo el desarrollo de sistemas de conocimiento y bases de datos, los sistemas multi-agentes, y el procesado del lenguaje natural.

En el desarrollo de sistemas lógicos de conocimiento, se incluye el uso de estructuras lógicas en las bases de datos, la revisión de creencias, la integración de información y la modelización lógica del aprendizaje. Por su parte, la interacción con los sistemas multi-agentes incluye el desarrollo de lógicas del tiempo y la acción, razonamiento abductivo (que se puede interpretar como búsqueda de explicaciones a hechos observados) o razonamiento no monótono (en el que fórmulas deducidas previamente pueden no ser válidas con un conjunto mayor de hipótesis)[6]. Por último, respecto del procesado del lenguaje natural, cabe destacar el desarrollo de semánticas lógicas para el lenguaje, la comprensión del lenguaje como procesos de inferencia, y los formalismos basados en la lógica para el análisis de las gramáticas.

APLICACIONES DE LA LÓGICA A LA INGENIERÍA. Los métodos lógicos están desempeñando un papel cada vez más importante en distintos campos de la ingeniería, especialmente en ingeniería eléctrica e ingeniería mecánica. Las técnicas más relevantes hacen uso de la lógica difusa y de instrumentos de control lógico para, por ejemplo, el desarrollo de controladores difusos para la industria, o el uso de herramientas lógicas para el control y la verificación de instrumentos mecánicos.

7 DEDUCCIÓN AUTOMÁTICA

El término deducción automática se vino a acuñar a partir de la introducción de mecanismos efectivos de deducción. Sin embargo, las inquietudes teóricas en el campo de la automatización del razonamiento distan mucho temporalmente de las realizaciones. La idea de crear un lenguaje simbólico aparece en la Edad Media con Raimundo Lulio [9], que centró su objetivo en encontrar

[6]El razonamiento no monótono puede resultar extraño desde un punto de vista matemático, sin embargo recuérdese que las teorías físicas se asumen válidas mientras corroboren los hechos observados, y dejan de ser admitidas al encontrarse un contraejemplo.

un «método» para demostrar si una afirmación es verdadera o falsa. Este objetivo fue más precisamente formulado con la máxima ambición por Hilbert, cuyo *Entscheidungsproblem* consistía en disponer de un algoritmo general que permitiera resolver cualquier problema matemáticamente bien definido.

Hasta bien entrado el siglo XX, casi todos los esfuerzos para automatizar las matemáticas estuvieron dirigidos hacia la mecanización de la aritmética, en lugar de la mecanización del proceso deductivo mismo; aunque contamos con la excepción de W.S. Jevons, que en 1870 diseñó un artilugio consistente en un teclado con un sistema de varillas que permitía la deducción automática de conclusiones a partir del par de premisas de un silogismo [34].

Una vez que los primeros ordenadores aparecieron en la década de 1950, un gran número de investigadores intentó automatizar la demostración de teoremas en lógica de predicados. Desde una perspectiva histórica, podemos identificar tres tendencias principales dentro de la deducción automática: (a) simular el proceso de razonamiento humano, (b) centrarse en la obtención de resultados de manera automática sin usar necesariamente métodos humanos de razonamiento, y por último (c) utilizar enfoques interactivos, en los que el proceso de deducción mecanizada está guiado, en mayor o menor medida, por el usuario.

7.1 Los primeros demostradores

El crédito de haber sido el primer programa de deducción automática se suele conceder a la *Logic Theory Machine*, desarrollada por A. Newell y H. Simon, e implementada por J.C. Shaw en 1955, para demostrar los teoremas de la lógica proposicional de los *Principia Matematica*. La *Logic Theory Machine* encontró demostraciones para 38 de los 52 teoremas presentes en el capítulo 2 de los *Principia* y, en particular, para uno de ellos encontró una demostración mucho más simple que la propuesta por Russell y Whitehead. Newell y Simon no estaban interesados en obtener un buen programa de deducción automática, sino explicar cómo la mente humana desarrolla el proceso de deducción y usar los heurísticos obtenidos para la implementación de un *General Problem Solver* [41]. Sin embargo, esta meta no era compartida por otros investigadores, que concebían al ordenador como «un afanado y persistente realizador de tareas rutinarias», y en 1960 existían al menos tres implementaciones para la lógica de primer orden, basadas en mayor o menor medida en el teorema de Herbrand, debidas a P. Gilmore [25], H. Wang [56] y D. Prawitz [47].

El éxito de tales enfoques estuvo limitado tanto por los procedimientos utilizados como por la capacidad de las máquinas disponibles, mucho menos potentes que el más barato ordenador personal del mercado en la actualidad. Por otra parte, M. Davis y H. Putnam [18] entre otros, hicieron algunos avances en la automatización «eficiente» de la demostración de teoremas en lógica de predicados, aunque no presentaron ninguna implementación.

7.2 MÉTODOS BASADOS EN LA RESOLUCIÓN

Inspirado en los trabajos de Prawitz [46] y Davis y Putnam [18], J.A. Robinson (1930–) presentó una implementación diferente del teorema de Herbrand [48] (también aparece en [53]). En ella Robinson introduce su método de resolución como la única regla necesaria para el cálculo proposicional, para el caso de lógica de predicados la introducción del conocido algoritmo de unificación es fundamental, ya que permite evitar la enumeración de los elementos de la base de Herbrand. La introducción del método de resolución permitió el uso más eficiente del razonamiento en sentido lógico, a pesar de que no podía trabajar con fórmulas arbitrarias de la lógica clásica, sino que debían estar en forma normal conjuntiva (como conjunción de disyunciones de literales).

Los limitados recursos, tanto en velocidad como en memoria, de los ordenadores de la época, hicieron del método de resolución el preferido de los investigadores en deducción automática respecto de otros enfoques ya existentes, como el de las tablas semánticas que comentaremos más adelante. Entre finales de los sesenta y principios de los setenta se produjo un *boom* de variaciones del método de resolución: en 1978, D. Loveland incluyó hasta 25 variaciones diferentes del método de resolución en su libro sobre demostración automática de teoremas [38].

A principios de los 1970 Robert Kowalski [35] introdujo la idea, sobre el trabajo de Alan Colmerauer *et al.* [14], de mecanizar la deducción para solo una cierta clase de fórmulas llamadas cláusulas de Horn[7]. Ya era conocido que la deducción con cláusulas de Horn permite calcular todas las funciones recursivas; sin embargo, ni siquiera trabajando en este pequeño fragmento de la lógica de predicados se conseguía una eficiencia suficiente. Ésta se consiguió con la introducción del lenguaje de programación PROLOG, cuyo procedimiento de demostración para cláusulas de Horn es conocido como resolución SLD; aunque su método de deducción es lógicamente *incompleto*[8], es suficientemente potente como para computar todas las funciones recursivas.

En la segunda mitad de la década de los setenta, en el Argonne National Laboratory (ANL) en el que J.A. Robinson descubrió el método de resolución, L. Wos y G. Robinson desarrollaron nuevos métodos de prueba, en particular la *demodulación* y la *paramodulación*, a la vez que R. Overbeek y E. Lusk mejoraban distintos aspectos de las implementaciones. A principios de los años ochenta, comenzaron a aplicar sus sistemas (Aura y Otter) no solo a teoremas cuya demostración era conocida, sino también a la búsqueda de soluciones de problemas abiertos de determinadas áreas de la matemática [59, 58, 60].

A pesar de todos los avances realizados, la eficiencia de las implementaciones existentes dejaban bastante que desear. Por otra parte, algunos resulta-

[7] Son fórmulas de la forma $\neg R_1 \vee \cdots \vee \neg R_n \vee T$ donde las R_i y T son fórmulas atómicas, y que corresponden a reglas del tipo $(R_1 \wedge \cdots \wedge R_n) \to T$

[8] Esta *incompletitud* significa que el algoritmo no es capaz de demostrar todas las fórmulas demostrables.

dos acerca de complejidad algorítmica indicaban que incluso la comprobación de la validez de una fórmula de la lógica proposicional es un problema NP-completo [15]. Estos hechos hicieron que gran parte de los investigadores en deducción automática se desvincularan del área, con la excepción de la gente de ANL, o bien se planteasen la utilización de estrategias no basadas en la regla de resolución.

7.3 Métodos no basados en resolución

Uno de los impulsores de la *deducción automática no basada en resolución* de los años setenta fue W.W. Bledsoe [5, 6]. Sus ideas estaban inspiradas en la «naturalidad» del proceso deductivo, que no debía encorsetarse en los formalismos de los métodos lógicos de demostración existentes. En particular, no se sintió obligado a buscar métodos *completos* de demostración, le bastaba que fueran buenos métodos en dominios específicos, y que las demostraciones que proporcionaran fueran naturales[9]. Sus nuevas ideas, junto con algunos heurísticos para el paso al límite específicamente diseñados como ayuda en las demostraciones en cálculo infinitesimal, le permitieron desarrollar un sistema capaz de demostrar muchos teoremas del cálculo infinitesimal elemental, un área en la cual los demostradores existentes hasta la fecha habían tenido poco o ningún éxito [7].

Dentro de los métodos no basados en resolución, no todo fueron nuevas ideas: también hubo que echar mano de procedimientos olvidados, como el método de tablas semánticas. Históricamente este método fue introducido por Beth [4] a finales de la década de los años 50, popularizado por Jeffrey [33] y Smullyan [54], y perfeccionado y extendido a algunas lógicas no clásicas por Fitting a principios de los setenta, aunque solo despertó un interés generalizado hasta principios de los noventa.

El método de tablas semánticas (o *tableaux*) está basado en refutación: se supone lo contrario de lo que queremos demostrar $\neg A$, y se comienza una búsqueda exhaustiva de contramodelos, si no se puede construir ningún contramodelo, entonces hemos demostrado la fórmula A. La búsqueda sistemática se realiza de manera analítica, y creando un árbol de posibilidades, cuyas reglas de extensión dependen de la estructura de la fórmula inicial. Existen un total de siete posibilidades de extensión para conectivos proposicionales, agrupados en dos tipos de reglas (según tengan comportamiento disyuntivo o conjuntivo), mientras que para la lógica de primer orden existen otras cuatro posibilidades para la gestión de los cuantificadores $\forall$ y $\exists$ agrupadas en dos nuevos tipos de reglas (según tengan comportamiento universal o existencial).

La introducción de ordenadores cada vez más potentes desde principio de los noventa, invitó al estudio de problemas en los que son necesarios contextos

[9]El uso de métodos incompletos es razonable ya que, después de todo, de un matemático no se espera que sepa (o pueda) demostrar todos los teoremas de las Matemáticas.

temporales o en los que se tenga que trabajar con incertidumbre, por ejemplo considerando lógicas multivaluadas (en las que las fórmulas pueden admitir más de dos valores de verdad). En el intento por desarrollar versiones de resolución para lógicas temporales o multivaluadas se observó la dificultad de extender el método de un modo más o menos natural, especialmente debido a la dificultad de encontrar análogos a la forma normal conjuntiva. Esta restricción no es aplicable al método de tablas, que además se puede generalizar de manera natural a varios tipos de lógicas no clásicas, lo cual sirvió como acicate para incorporarlo como método eficiente de deducción automática.

Desde 1992 se celebran congresos de carácter anual dedicados específicamente al estudio teórico y a las aplicaciones del método de tablas, que se han mostrado como un formalismo conveniente para la deducción automática tanto en lógicas no clásicas como en la propia lógica clásica. El avance producido en el estudio de los métodos de tablas ha llevado a la conveniencia de publicar tratados monográficos en los que se presenta de manera sistematizada el desarrollo y los distintos aspectos de este enfoque [17].

Hemos de citar en este punto un nuevo enfoque para la deducción automática, llamado TAS desarrollado por nuestro grupo de investigación que, en cierto punto, puede considerarse como una alternativa a los métodos de resolución y de tablas semánticas.

El punto clave para obtener algoritmos de satisfacibilidad eficientes es la capacidad de controlar el proceso de ramificación, y el enfoque de la metodología TAS (por Transformaciones de Árboles Sintácticos), introducida en la tesis doctoral de F. Sanz [49] en 1992, está basado en la reducción de la fórmula tanto como sea posible (obviamente con una complejidad razonable tanto en tiempo como en espacio) antes de aplicar cualquier proceso de ramificación.

Para determinar la satisfacibilidad de una fórmula dada, en primer lugar se intenta reducir el tamaño de la misma mediante la aplicación sucesiva de transformaciones que conservan la satisfacibilidad (no necesariamente el significado), entonces se selecciona una variable respecto de la que ramificar, y se aplica recursivamente el proceso en cada una de las tareas generadas.

Por último, es conveniente resaltar que TAS intenta obtener características sintácticas distintivas de los teoremas, es decir, aquello que los hace válidos; considera que lo más útil es disponer de condiciones suficientes, eficientemente implementables, que aseguren la falsedad o veracidad de una fórmula. Estas condiciones se suelen formular como teoremas de reducción en términos de implicantes e implicados unitarios de las fórmulas[10].

Esta misma idea puede extenderse a lógicas no clásicas en las que no existe una forma normal conjuntiva, asimismo es suficientemente general porque se puede aplicar a diferentes tipos de lógicas, y también es flexible, porque las modificaciones necesarias para cambiar de lógica se pueden realizar de manera

[10]Un implicante (resp. implicado) unitario de la fórmula A es un literal ℓ tal que $\ell \models A$, esto es, siempre que ℓ es verdadero también lo es A, (resp. $A \models \ell$).

modular sin necesidad de rediseñar todo el demostrador. De hecho, aparte de la versión proposicional clásica [1], existen métodos TAS para Lógicas Temporales [19] y Lógicas Multi-Valuadas [20].

7.4 Métodos interactivos de deducción automática

Otra línea de investigación en deducción automática ha consistido en dirigir el interés hacia sistemas semi-automáticos interactivos que permitan la ayuda externa del usuario en la búsqueda de las demostraciones.

J.R. Guard *et al.* desarrollaron a mediados de los sesenta una serie de sistemas llamados SAM (*Semi-Automated Mathematics*) [29] en los que se persigue una combinación de rutinas lógicas automáticas con la intervención humana en tareas de control y de guía.

El desarrollo de sistemas interactivos de deducción automática fue, asimismo, guiado por otra meta de interés para las Matemáticas: la validación automática de pruebas. Podemos citar en este punto el desarrollo del sistema *Automath*, junto con la comprobación en 1977, por parte de L.S. van Benthem Jutting, de todas las demostraciones de los *Grundlagen der Analysis* de Landau. Pronto se observó que el uso de este tipo de sistemas resultaría especialmente interesante en la *verificación de programas*, más que en la comprobación de pruebas matemáticas.

Entre los sistemas interactivos más conocidos desarrollados hasta la fecha, podemos destacar el NQTHM de Boyer y Moore. Su principal característica es que está fuertemente basado en la inducción matemática, que proporciona una importante técnica de demostración cuando, por ejemplo, se desean demostrar propiedades acerca de estructuras definidas como clausuras inductivas libremente generadas del tipo de los números naturales, o las listas finitas de elementos, o los árboles de ramificación finita, etc.

Por una parte, es claro que el principio de inducción no puede expresarse directamente en lógica de primer orden, puesto que se necesita cuantificar sobre todos los predicados (el principio de inducción se enuncia ≪para cualquier propiedad P≫) y, por lo tanto, no suele ser usado en demostradores basados en resolución. Por otra parte, la dificultad principal de la automatización de pruebas por inducción es la elección adecuada del parámetro sobre el cual realizar la inducción, así como la elección de la hipótesis de inducción más conveniente. La solución propuesta consiste en la prohibición expresa del uso de los cuantificadores en el lenguaje, usándose para tal fin funciones recursivas. Este hecho tiene como consecuencia que el usuario implícitamente indica al sistema cómo probar conjeturas: ≪se intenta aplicar inducción reflejando las definiciones de las funciones recursivas usadas en la especificación≫. Entre los logros más significativos conseguidos con este sistema figura una demostración automática del teorema de incompletitud de Gödel realizada en 1986 por N. Shankar. Un descendiente directo de NQTHM es ACL2 (*A Computational Logic for Applicative Common Lisp*) que está formado por un lenguaje lógico junto con un demostrador automático basado en NQTHM. El sistema tiene diferentes facetas, pues puede usarse, por ejemplo, como lenguaje de programación, como una lógica matemática formal, o como un demostrador semiautomático. En relación con la deducción automática, es posible usar ACL2 para «razonar» acerca de demostradores automáticos; por ejemplo, en [40] se presenta una implementación en ACL2 de un marco general para la deducción automática en lógica proposicional que, usando los mecanismos del sistema, se demuestra que tiene las propiedades de terminación, corrección y completitud a nivel general. Especificando el marco abstracto, es posible obtener versiones ejecutables y formalmente verificadas de, por ejemplo, el método de Davis y Putnam y del método de tablas semánticas.

7.5 Deducción Automática . . . y Matemáticas

El desarrollo formal de la demostración automática ha necesitado de herramientas puramente matemáticas bien conocidas, en particular de matemática discreta (se hace uso abundante de clausuras inductivas libremente generadas, teorías de árboles y de grafos, etc.). Por su parte, ya hemos visto que algunos problemas abiertos en Matemáticas han sido demostrados mediante el uso de sistemas de deducción automática. Quizá al lector se le venga a la memoria la demostración por Appel y Haken en 1977 del teorema de los cuatro colores [2]. En dicha prueba se usó la potencia *computacional* de un ordenador para «comprobar» la reducibilidad de un conjunto de miles de configuraciones básicas inevitables. La obtención de dicho teorema no tiene nada que ver con el uso de la capacidad deductiva del sistema usado, sino la aplicación directa de la fuerza bruta.

Dentro de los teoremas demostrados usando sistemas de deducción automática, destaca la prueba en 1996 de una conjetura de H. Robbins hecha en los años treinta, según la cual todas las álgebras conmutativas y asociativas que satisfacen, además, la ecuación de Robbins $n(n(x+y)+n(x+n(y))) = x$ (donde n representa el complemento) son álgebras de Boole. La demostración se realizó usando un sistema de deducción automática con igualdad en modo completamente automático. El sistema, llamado EQP, está basado en Otter y está dotado de AC-unificación, esto es, unificación módulo conmutatividad y asociatividad.

En este contexto también podemos destacar los trabajos de Kennet Kunen [36, 37], publicados en una revista de prestigio «tradicional» como es el *Journal of Algebra*, en los que ha usado Otter para obtener demostraciones de algunos resultados relativos a quasi-grupos.

Por otra parte, es asimismo destacable el demostrador para la Geometría desarrollado por Chou, Gao y Zhang [13], que ha sido usado para obtener nuevos resultados en geometrías no euclídeas.

Dentro de la verificación formal de resultados matemáticos es conveniente destacar el proyecto Mizar (`http://www.mizar.org`), que desde 1973 tiene como objetivo la construir una matemática formalmente certificada, mediante el desarrollo de un lenguaje formal fácilmente legible por los matemáticos y, simultáneamente, suficientemente riguroso como para que pueda ser procesado y verificado por computador. En conjunción con el *J. of Formalized Mathematics*, se está construyendo una base de datos de la «matemática formalizada» que en la actualidad contiene más de dos mil definiciones y treinta mil teoremas.

Para terminar esta sección, es conveniente comentar aunque sea brevemente los intentos de aunar la potencia deductiva y la potencia computacional mediante la integración de sistemas de deducción automática con sistemas de álgebra computacional que se están realizando dentro del grupo Calculemus (`http://www.calculemus.net`).

8 Epílogo

El cálculo se convirtió en una herramienta muy extendida una vez desarrollados sus algoritmos algebraicos y simbólicos. Con la llegada de los actuales computadores de alto rendimiento, ahora se usa para problemas de una talla y dificultad sencillamente inimaginables hace solo unos pocos años: problemas que quedaban fuera de las posibilidades computacionales de los sistemas anteriores hoy día se pueden resolver bien automáticamente o bien de modo interactivo.

Podemos hacer una extrapolación semejante para la deducción automática en un futuro próximo, puesto que las implementaciones se van adaptando rápidamente a los nuevos modelos de computadores. Este extraordinario desarrollo ha permitido que los sistemas de deducción automática hayan ampliado su horizonte de aplicaciones en áreas tales como el razonamiento sobre programas, razonamiento sobre agentes, y planificación; actualmente existen «asistentes matemáticos» que usan e incrementan conjuntos de lemas, demostraciones, tácticas y estrategias.

Cada vez se espera más de estos sistemas, de modo que sepan reaccionar de modo «inteligente» en función de las distintas peticiones al sistema, deduciendo ciertos datos automáticamente. Es esperable que, finalmente, la deducción automática sea, simplemente, tan útil como es el cálculo en la actualidad y con el mismo amplio campo de aplicaciones. Sencillamente, una extensión moderna del sueño de Leibniz.

Referencias

[1] G. Aguilera, I. P. de Guzmán, M. Ojeda-Aciego, A. Valverde. Reductions for non-clausal theorem proving. *Theoretical Computer Science*, **266** (2001) nº 1/2, 81–112.

[2] K. Appel, W. Haken. Every planar map is four colorable. *Illinois J. Mathematics*, **21** (1977) 429–567.

[3] J.W. Backus. The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM Conference. *Proc. of the Intl Conf on Information Processing*, IFIP'59, pp. 125–131, 1959

[4] E.W. Beth. *The foundations of mathematics.* North-Holland, 1959.

[5] W.W. Bledsoe. Splitting and reduction heuristics in automatic theorem proving. *Artificial Intelligence*, **2** (1971) 55–77.

[6] W.W. Bledsoe. Non-resolution theorem proving. *Artificial Intelligence*, **9** (1977) 1–35.

[7] W.W. Bledsoe, R.S. Boyer, W.H. Henneman. Computer proofs of limit theorems. *Artificial Intelligence*, **3** (1972) 27–60.

[8] W.W. Bledsoe, D.W. Loveland, editors. *Automated Theorem Proving: after 25 years*, volume 29 of *Contemporary Mathematics.* American Mathematical Society, 1984.

[9] I.M. Bocheński. *Historia de la Lógica Formal.* Gredos, 1976.

[10] G. Boole. *An investigation of the laws of thought, on which are founded the mathematical theories of logic and probability.* Macmillan, 1854.

[11] G. Cantor. Über unendlich lineare punktmannigfaltigkeiten. *Math. Annalen*, **17** (1880) 355–358.

[12] N. Chomsky. On certain formal properties of grammars. *Information and Control*, **2** (1959) nº2, 137–167.

[13] S.C. Chou, X.S. Gao, J.Z. Zhang. *Machine Proofs in Geometry.* World Scientific, 1994.

[14] A. Colmerauer et al. *Un systeme de communication homme-machine.* Technical report, Gr. de Recherche en Intelligence Artificielle, Univ. d'Aix-Marseille, 1973.

[15] S.A. Cook. The complexity of theorem proving procedures. In *3rd Annual ACM Symposium on the Theory of Computing*, 1971.

[16] H.B. Curry, R. Feys. *Combinatory Logic.* North-Holland, 1958.

[17] M. D'Agostino, D.M. Gabbay, R. Hähnle, J. Posegga (eds.) *Handbook of Tableau Methods* Kluwer Academic Publishers, 1999.

[18] M. Davis, H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, **7** (1960) 201–216. Reimpreso en [53].

[19] I. P. de Guzmán, M. Ojeda-Aciego, A. Valverde. Implicates and reduction techniques for temporal logics. *Annals of Mathematics and Artificial Intelligence*, **27** (1999) 2–23.

[20] I. P. de Guzmán, M. Ojeda-Aciego, A. Valverde. Reducing signed propositional formulas. *Soft Computing*, **2** (1999) nº4, 157–166.

[21] A. de Morgan. *Formal logic, or, the calculus of inference, necessary and probable.* Taylor and Walton, 1847.

[22] R. Dedekind. *Essays on the theory of numbers.* Dover, 1963.

[23] G. Frege. *Begriffsschrift eine der arithmetischen nachgebildete Formelsprache des reinen Denkens.* Halle, 1879. Reimpreso en [55] como "*Begriffsschrift*, a formula language, modeled upon that of arithmetic, for pure thought".

[24] G. Frege. *Grundgesetze der Arithmetik, begriffsschriftlich abgeleitet I.* Verlag Hermann Pohl, 1893. Volumen II en 1903.

[25] P.C. Gilmore. A proof method for quantification theory; its justification and realization. *IBM J. Res. Develop.*, **4** (1960) 28–35.

[26] K. Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Mon. für Math. und Physik*, **37** (1930) 349–360. Reimpreso en [28, 55] como "The completeness of the axioms of the functional calculus of logic".

[27] K. Gödel. Über formal unentscheidbare sätze der Principia Mathematica und verwandte systeme. *Monat. Math. Phys.*, **38** (1931) 173–198.

[28] K. Gödel. *Collected works.* Oxford University Press, 1986.

[29] J.R. Guard et al. Semi-automated mathematics. *Journal of the ACM*, **16** (1969) 49–62.

[30] J. Herbrand. *Recherches sur la Théorie de la Démonstration.* PhD thesis, Université de Paris, 1930. Traducido parcialmente en [55] como "Investigations in proof theory: the properties of true propositions".

[31] D. Hilbert and W. Ackermann. *Grundzüge der theoretischen Logik.* Springer, 1928. Traducción inglesa de la edición de 1938 publicada por Chelsea, New York, en 1950.

[32] D.A. Huffman. The synthesis of sequential switching circuits. *Journal of the Franklin Institute*, **257** (1954) 161–190.

[33] R.C. Jeffrey. *Formal logic: its scope and limits.* McGraw-Hill, 1967.

[34] W.S. Jevons. On the mechanical performance of logical inference. *Philosophical transactions of the Royal Society*, **160** (1870) 497–518.

[35] R.A. Kowalski. Predicate logic as a programming language. In *Information processing'74.* North-Holland, 1974.

[36] K. Kunen. Moufang quasigroups. *Journal of Algebra*, **183** (1996) 231–234.

[37] K. Kunen. Quasigroups, loops, and associative laws. *Journal of Algebra*, **185** (1996) 194–204.

[38] D.W. Loveland. *Automated Theorem Proving: a logical basis.* North Holland, 1978.

[39] A. Macintosh Wilson. *The infinite in the finite.* Oxford University Press, 1995.

[40] F.J. Martín-Mateos, J.A. Alonso, M.J. Hidalgo, J.L. Ruiz-Reina. Verification in ACL2 of a Generic Framework to Synthesize SAT-Provers. *Lecture Notes in Computer Science* 2664:182–198, 2003

[41] A. Newell, H. Simon. GPS: a program that simulates human thought. In E.A. Feigenbaum and J. Feldman, editors, *Computers and Thought*. Mc Graw Hill, 1963.

[42] G. Parkinson. *Logic and Reality in Leibniz's metaphysics*. Clarendon Press, 1965.

[43] G. Peano. *Formulario mathematico.* Bocca Freres, 1894–1908.

[44] G. Peano. *Arbeiten zur analysis und zur mathematischen Logik.* Teubner, 1990.

[45] E.L. Post. Introduction to a general theory of elementary propositions. In Jean van Heijenoort, editor, *From Frege to Gödel. A Source Book in Mathematical Logic, 1879–1931*, 264–283. Harvard University Press, Cambridge MA, 1921.

[46] D. Prawitz. An improved proof procedure. *Theoria*, **26** (1960) 102–139.

[47] D. Prawitz, H. Prawitz, N. Voghera. A mechanical proof procedure and its realization in an electronic computer. *Journal of the ACM*, **7** (1960) 102–128.

[48] J.A. Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, **12** (1965) 23–41. Reimpreso en [53].

[49] F. Sanz Jiménez. *TAS-D^{+}: un nuevo ATP dirigido por la sintaxis. Hacia una alternativa a la resolución.* PhD thesis, Universidad de Málaga, España, 1992.

[50] E. Schröder. *Der Operationskreis des Logikkalkulus.* Teubner, 1877.

[51] E. Schröder. *Vorlesungen über die Algebra der Logik.* Teubner, 1890–1905. Tres volúmenes.

[52] C.E. Shannon. A symbolic analysis of relay and switching circuits. *Transactions of the AIEE*, **57** (1938) 713–723.

[53] J. Siekmann, G. Wrightson, editors. *Automation of reasoning. Classical papers on computational logic*, volume 1. Springer-Verlag, 1983.

[54] R.M. Smullyan. *First-Order Logic.* Springer-Verlag, 1968. Reimpreso por Dover en 1995.

[55] J. van Heijenoort, editor. *From Frege to Gödel: a source book in mathematical logic.* Harvard University Press, 1967.

[56] H. Wang. Toward mechanical mathematics. *IBM J. Research and Development*, **4** (1960) 2–21.

[57] A.N. Whitehead, B. Russell. *Principia Mathematica.* Cambridge University Press, 1910-1913.

[58] S. Winker. Generation and verification of finite models and counterexamples using an automated theorem prover answering two open question. *Journal of the ACM*, **29** (1982) n°2, 273–284.

[59] S. Winker, L. Wos, E.L. Lusk. Semigroups, antiautomorphism, and involutions: A computer solution to an open problem. *Mathematics of Computation*, **37** (1981) 533–545.

[60] L. Wos, S. Winker. Open questions solved with the assistance of AURA. In Bledsoe and Loveland [8], pages 73–88.

Manuel Ojeda Aciego
Dept. Matemática Aplicada
Universidad de Málaga
Correo electrónico: aciego@ctima.uma.es