

En el umbral de la computabilidad

Francisco Salto y José M. Méndez
Universidad de Salamanca
2000

Hay problemas cuya solución requiere manipular una cantidad infinita de información. Los procedimientos algorítmicos finitarios de solución de problemas están demostrablemente incapacitados, incluso en condiciones ideales, para realizar algunas de tales tareas. Por otra parte, los seres humanos disponemos de medios para producir demostraciones infinitarias en la práctica matemática informal. No se trata, por supuesto, de pruebas infinitamente largas, sino de secuencias finitas de (ocurrencias de) símbolos finitos capaces de *subrogar* o *representar* nociones o inferencias infinitarias. Del aparente conflicto entre las limitaciones metamatemáticas de las máquinas sintácticas recursivas y parte de la práctica matemática habitual surgen sendas actitudes opuestas respecto a la posibilidad de mecanizar el razonamiento matemático: o bien entendemos que el conflicto muestra que tenemos habilidades matemáticas que ninguna máquina sintáctica axiomática o algorítmica tiene, y por tanto la mecanización completa es imposible, o bien el conflicto muestra las limitaciones de las capacidades matemáticas humanas, con independencia del valor heurístico y explicativo (pero no demostrativo) de todas las prácticas matemáticas informales. El conflicto aducido refleja un conflicto aparente más general: la hipótesis que cualquier suceso físico es ya la computación de su propia conducta entra en conflicto con sucesos descritos abstracta pero exitosamente con funciones continuas infinitamente precisas, o descritos mediante sistemas dinámicos cuyas soluciones no son recursivas, incluso aunque sus condiciones iniciales lo sean.

En el umbral de la computabilidad están todos los algoritmos efectivos que no son codificables como funciones recursivas sobre números naturales. De acuerdo con el paradigma computacional imperante, tales procedimientos simplemente no cuentan como computaciones. Este trabajo argumenta nuestro rechazo a admitir esta limitación. Abogamos por una concepción *abierta* de la computabilidad, libre de la constricción de recursividad (en el sentido recién indicado). Los conflictos señalados son genuinos por cuanto no parece haber razones que nos aseguren que cualquier suceso real o cualquier acción inteligente haya de ser necesariamente reproducible sólo mediante algoritmos representables en algunas colecciones de números naturales. Pero entendemos que tales conflictos son perfectamente irrelevantes a la hora de refutar o fundamentar a priori modelos computacionales, por cuanto el propio fenómeno de la computabilidad admite la misma consideración. Los resultados metamatemáticos de indecidibilidad simplemente no son, necesariamente, resultados de incomputabilidad.

Una manera extensionalmente equivalente de expresar la aludida constricción de recursividad es afirmar que todo algoritmo computable es Turing-computable. La Turing-computación merece ser llamado paradigma de computación imperante por cuanto caracteriza en abstracto y desde un punto de vista lógico el tipo de funciones recursivas que se consideran computables, con independencia del grado de efectividad, rapidez y complejidad con la que distintos modelos, lenguajes y arquitecturas

computacionales las realizan. Máquinas abstractas deterministas o indeterministas, probabilistas, multiprocesadas, multiprogramadas y distintos tipos de redes computan, con distinta efectividad, como máximo, cualquier función computable por una máquina universal de Turing.

Aunque existen modelos computacionales abstractos «más allá» del paradigma ([2],[25],[31], entre otros), su sola existencia no avala la tesis que pretendemos establecer, como tampoco lo haría un oráculo omnisciente ni un intuidor categorial. Es necesario además mostrar que los procedimientos en cuestión son efectivos y finitamente realizables. Lamentablemente, hemos de reconocer que resultará tan vano *demostrar* que hay computaciones más allá del límite de Turing como vano ha resultado *demostrar* que toda computación es Turing-computable. Este es el caso debido a que las nociones básicas ‘finito’ y ‘eficiente’ tienen un significado controvertido. Por tanto, no hay una vía directa para establecer nuestra tesis, aunque sí varias estrategias indirectas alternativas. En [22] mostramos cómo representar funciones relevantemente recursivas (funciones que «realmente» dependen de sus argumentos) en sistemas formales no clásicos de modo análogo a como en sistemas formales clásicos se representan funciones recursivas en fragmentos de la aritmética de Peano. Aquí seguimos una estrategia diferente y complementaria, de carácter fundamentalmente conceptual. Si no podemos mostrar la *verdad* de paradigmas computacionales más flexibles, sí podemos mostrar su *conveniencia* teórica.

1. Antecedentes: el milagro de la recursividad

Es claro que inventar y esclarecer la noción de recursividad ha sido uno de los principales logros intelectuales del siglo que termina. Hace no tantos decenios a la noción de computabilidad no había asociado ningún contenido matemático preciso. Se trataba de una noción epistemológica informal que distintas líneas de investigación pretendían capturar con distintos aparatos teóricos. En particular, estos programas de trabajo pretendían definir matemáticamente las condiciones necesarias y suficientes para que una función total arbitraria sobre los números naturales fuese una función eficiente, esto es, calculable mecánicamente aplicando rutinariamente finitos algoritmos de complejidad finita. Las líneas de trabajo para definir tales funciones fueron diversas, empleándose técnicas tan diferentes en principio como sistemas de ecuaciones (Herbrand), sistemas formales axiomáticos (Gödel, Kleene), autómatas finitos (Turing, Wang), cálculo λ (Church), etc. Es un hecho histórico bien conocido cómo estas distintas líneas de investigación acabaron convergiendo y encontrando clases de funciones que resultaron ser globalmente equivalentes¹ entre sí y globalmente equivalentes a las funciones recursivas. Esta no fue, desde luego, una causalidad histórica, sino el resultado de las propiedades lógicas de la clase de las funciones o reglas recursivas, que permiten lo que Gödel llegó a considerar, literalmente, milagroso ([8]): preservarse en representaciones formales radicalmente diferentes. Sistemas y lenguajes diversos diseñados para campos entonces tan diversos como la teoría de números, la teoría de pruebas lógicas y la

¹ La equivalencia es global pero no local. Para cualquier función definida en algún formalismo adecuado (el de Herbrand, Gödel, Kleene, Turing, Church, Wang,...) hay alguna función equivalente en cualquier otro formalismo. Esto no quiere decir que cualquier función sea ella misma, dada en un formalismo dado, computable por cualquiera de los demás formalismos.

lingüística, preservaban una misma noción abstracta de algoritmo. Esto es, las funciones Turing-computables coinciden globalmente con las Gödel-computables, que coinciden con las λ -definibles y las Herbrand-computables, y todas ellas con las representables en cualquier extensión de la lógica clásica de primer orden con un fragmento de la aritmética formal. Este hecho otorgó a la noción de recursividad el papel extraordinario que aún hoy tiene entre todas las demás nociones lógicamente definibles. En palabras de Gödel [8] libremente traducidas:

«Con el concepto de computabilidad [recursividad] por primera vez se ha conseguido obtener una definición absoluta de una noción epistemológicamente importante, es decir, una definición que no depende del formalismo elegido. En todos los demás casos, como con los conceptos de demostrabilidad y definibilidad, sólo se han logrado definiciones relativas a un lenguaje dado, y está claro que, en cada lenguaje, la definición relativa no es lo que se buscaba. En cambio, esta situación es diferente con el concepto de computabilidad [recursividad], y eso que es sólo un tipo especial de demostrabilidad o definibilidad.»

El hecho que la misma clase de funciones se obtenga a partir de diferentes sistemas abstractos de reglas otorgó un contenido matemático preciso a la idea de que uno y el mismo algoritmo abstracto puede instanciarse en formalismos diferentes, y en consecuencia, realizarse físicamente en soportes diversos. El «descubrimiento» de la recursividad hizo posible el concepto –hoy acaso trivial, pero entonces en absoluto– de sistema inteligente abstracto realizable indiferentemente en fragmentos de ADN, en artilugios electrónicos o en cerebros. En suma, con el descubrimiento y definición de la clase de funciones recursivas, la comunidad científica podía dar por solucionado el problema de partida. La tesis de Church-Turing hace explícita esta convicción:

(tesis de Church-Turing) Un función (total) sobre números naturales es eficientemente computable si y sólo si es recursiva.

2. El milagro en sus límites

Los hechos que sorprendieron a los grandes lógicos de mediados de siglo no tienen por qué sorprendernos ahora a nosotros del mismo modo. Fue sorprendente que todos los sistemas formales consistentes clásicos suficientemente expresivos pudiesen capturar relaciones aritméticas que coinciden con las funciones recursivas. Ahora bien, hoy disponemos de sistemas formales no clásicos capaces de capturar formalmente relaciones modales e intensionales tales que, enriquecidos con axiomas aritméticos como los sistemas formales clásicos, sin embargo representan clases de funciones distintas de las representadas por éstos. Este es claramente el caso para sistemas formales cuyo operador de negación sea suficientemente distinto de la negación booleana. En general, la noción de recursividad es sólo relativamente estable desde un punto de vista lógico, una vez que entran en consideración aritméticas formales cuya lógica subyacente es suficientemente distinta de la lógica clásica.

Además de las razones profundas pero más o menos esotéricas recién aludidas, la noción de recursividad deja de ser estable desde el momento que se consideran funciones

sobre números reales y no sólo sobre números naturales. Desde un principio, Turing se planteó [27] estudiar las posibilidades de su formalismo para describir cómo calcular efectiva o mecánicamente funciones sobre los números reales. Al emplear expansiones decimales infinitas para expresar números reales, constató inmediatamente cómo, bajo tal notación, ni siquiera funciones como $x \mapsto 3 \cdot x$ resultaban computables. Por este motivo propuso en [28] un sistema de notación que no expresa unívocamente números reales, pero permite Turing-computar algunas funciones reales intuitivamente computables. Desde ese momento quedó claro que la computabilidad de funciones reales sí es dependiente de los sistemas de notación que cada formalismo: el *milagro* no alcanza a las funciones reales computables.

Es fácil extender la tesis de Church-Turing de modo que defina extensionalmente condiciones necesarias y suficientes para que cualquier conjunto enumerable sea computable. Sin embargo, dado el conocido hecho que hay enumerables algoritmos pero una cantidad no enumerable de números reales, necesariamente hay infinitos números reales que no son Turing-computables. A fortiori, infinitas funciones reales no son recursivas. Lo cual no implica que no sean computables por máquinas abstractas capaces de manipular infinita información. Más aún, el milagro que ocurrió con la clase de funciones aritméticas recursivas no se reproduce con las funciones reales, ya que distintas máquinas abstractas (como [2] y [32]) generan clases diferentes de funciones reales computables. En consecuencia, no existe para éstas un análogo de la tesis de Church-Turing de manera que, en último término, la noción de función real computable no es un concepto metamatemáticamente preciso.

Hemos parado a mostrar serias limitaciones en la estabilidad del concepto de computabilidad en dos contextos diferentes: las aritméticas no estándar y las funciones reales computables. De estas consideraciones se deduce que el milagro de la recursividad está mal comprendido cuandoquiera que se apela a él para avalar la siguiente versión de la tesis de Church-Turing:

(Tesis de Church-Turing) Un algoritmo es computable si y sólo si es recursivo

Lamentablemente, la confusión entre la tesis y la Tesis es ubicua en muchos cuarteles. La diferencia entre ambas es notable, dado que aquella no ha sido refutada (ni es previsible que lo sea), mientras que ésta es refutada por las consideraciones arriba expuestas. Es, por supuesto, un error histórico [3] atribuir a Church ni a Turing la paternidad de la Tesis. Esta proviene de un error que se ha extendido epidémicamente: entender que la estabilidad o irrelatividad de la clase de las funciones recursivas llega al extremo de poder capturar formalmente o dar precisión al concepto mismo de computabilidad.

3. Computabilidad sin milagros

No esperamos que la naturaleza se comporte de modo aproximado a como se comporta. En particular, no esperamos que sistemas físicos cuyo comportamiento está determinado por constantes reales (expresadas por números reales) sean adecuadamente descritos por aproximaciones finitas a tales números. Esperamos, por ejemplo, que los

planetas se muevan de acuerdo al valor exacto de la constante gravitacional, y no de acuerdo al valor efectivamente computado por los humanos (que son sólo dos dígitos de su expansión decimal, por lo que sabemos), y análogamente con otras constantes físicas como la carga del electrón, la constante de Planck, etc. El empleo de constantes permite obtener mediciones y predicciones precisas, por más que los términos que las designan no tengan un valor numérico conocido con exactitud. Pues bien, los modelos de computación más allá del límite de Turing explotan este modo de proceder introduciendo constantes como “puntos de referencia” para la computación de números reales (ver [11], [32]). El resultado es, respecto de una misma notación, computacionalmente equivalente a la extensión de una máquina de Turing con la capacidad para calcular secuencias infinitas de símbolos.

Este tipo de máquinas abstractas requiere, para realizarse², sistemas capaces de asignar valores arbitrariamente precisos a funciones reales, y requiere además la ausencia total de “ruido”, ya que éste reduce su poder computacional al de máquinas de Turing (como se demuestra en [13]). El punto conceptual que aquí nos importa es, sin embargo, que fenómenos continuos en la naturaleza pueden entenderse como computaciones infinitamente precisas de determinadas constantes físicas. Tales computaciones son recursivas en el sentido informal de ser recurrentes (autoaplicación de las funciones reales), pero no son recursivas en su sentido técnico, al no ser Turing-computables. De hecho, argumentos bien conocidos de Gödel y Turing demuestran recursivamente que son procedimientos no recursivos.

Llamemos irrecursiva a cualquier regla o función tal que sea recursivamente demostrable que no es recursiva. No es más misterioso que existan sistemas físicos que se comporten irrecursivamente que existan verdades matemáticas demostrables irrecursivamente. La conveniencia teórica de reconocer computaciones no recursivas reside en la capacidad para explicar de modo unificado y satisfactorio cómo la existencia de reglas irrecursivas no constituye necesariamente, en ambos casos, una limitación en los modelos computacionales tanto de fenómenos naturales como artificiales.

El paradigma recursivo imperante no ofrece una explicación análoga, dado que: (a) una consecuencia obvia del *milagro* es que las limitaciones lógicas de los sistemas o máquinas axiomáticas son también limitaciones Turing-computacionales (y a la inversa), y (b) de la Tesis de Church-Turing se sigue que tales limitaciones son a su vez todas y sólo las limitaciones abstractas de cualquier computación. Ambos supuestos han dado lugar a argumentos antimecanicistas que cuestionan que el fin último de la inteligencia artificial sea conseguible siquiera en relación a emular el razonamiento aritmético humano. Tales argumentos son, desde un punto de vista estrictamente lógico, en ocasiones un tanto torpes. De hecho, hay cierto consenso mayoritario en considerar que los teoremas de incompletud de los sistemas axiomáticos no son relevantes para decidir la cuestión de si poseemos habilidades aritméticas que ninguna máquinas recursiva puede poseer. Sin embargo, y sorprendentemente, no existe un cuerpo único y consistente de argumentos que avalen este consenso. Por el contrario, las razones aducidas son diversas y a menudo inconsistentes entre sí. Este hecho explica que a las sucesivas refutaciones de

² Una teoría de la computabilidad no puede considerarse completa con la elaboración de modelos abstractos de algoritmos; es necesario además explicar sus condiciones generales de realizabilidad o instanciación. Incluso para la propia Turing-computación, tal explicación no existe aún con la generalidad que sería deseable (a pesar de [...]). Los modelos más allá del límite de Turing plantean dificultades añadidas relativas a la realizabilidad física de tareas infinitas.

los argumentos antimecanicistas basados en teoremas de incompletud han seguido hasta este mismo momento y desde los años treinta, nuevos argumentos antimecanicistas. A continuación presentamos un diagnóstico de esta situación en las siguientes líneas generales: las refutaciones de los argumentos antimecanicistas en el marco del paradigma computacional imperante dejan claro que sus conclusiones no se *deducen*, no son *consecuencia lógica*, de la sola aplicación de tales teoremas a las putativas máquinas recursivas que emulen el razonamiento matemático informal. Sin embargo, no han logrado mostrar (unánimemente) que sus conclusiones tampoco se *inferen* razonablemente de tal aplicación. Existe un acuerdo (casi) general sobre los resultados, pero no sobre las razones que los avalan. En el entorno de un paradigma computacional más amplio, en cambio, tales argumentos son manifiestamente inconclusivos tanto deductiva como inferencialmente. Terminaremos ofreciendo una explicación alternativa de los supuestos erróneos en los argumentos antimecanicistas, independiente de la Tesis de Church-Turing.

4. Preliminares para el diagnóstico

Si una parte de la actividad mental humana puede ser emulada o incluso reproducida computacionalmente, cabe esperar que lo sea la actividad razonadora y en particular el razonamiento matemático. La demostración automática de resultados matemáticos (de los que [...] es un excelente compendio) asegura que fragmentos substantivos del razonamiento matemático son de hecho reproducibles recursivamente. Sin embargo, también es cierto que en la práctica matemática abundan pruebas informales no recursivas. Este hecho por sí sólo puede hacer más o menos plausibles modelos Turing-computacionales del razonamiento matemático informal, sin llegar a refutarlos o legitimarlos, puesto que queda abierta la posibilidad lógica de encontrar demostraciones recursivas de los mismos resultados. Lo que es más importante para nosotros es la presencia de pruebas tales que es recursivamente demostrable que no son recursivas. Llamamos, como ya estipulamos, *irrecursivas* a este tipo de pruebas no recursivas. Las pruebas *irrecursivas* no son, por definición, recursivamente emulables. Por lo tanto, la existencia de prácticas inferenciales *irrecursivas* muestra que no todo el acervo de razonamientos matemáticos informales es reproducible recursivamente. (Nótese que esta afirmación no es por sí sola un argumento antimecanicista respecto del paradigma ampliado ni respecto del paradigma recursivo, ya que para el último se necesitaría una prueba independiente de la validez de tales prácticas inferenciales, y una prueba axiomática-recursiva es precisamente lo que no es aducible al ser pruebas *irrecursivas*).

Antes de estudiar algunos ejemplos ilustrativos de pruebas *irrecursivas*, es al caso presentar unas consideraciones previas. Obsérvese que los sistemas o máquinas axiomáticas están diseñadas para construir demostraciones, y resultan particularmente poco aptas para producir lo que llamamos *indemostraciones*, esto es, para demostrar que fórmulas no son teoremas. Esto se explica fácilmente por ser una tarea infinitamente compleja la de recorrer todas las posibles secuencias finitas de objetos abstractos que pueden ser una prueba o demostración de una fórmula dada. Evidentemente, demostrar la *irrecursividad* de una prueba o un resultado es producir una *indemostración* y frutos del trabajo metamatemático más notable tanto en un entorno axiomático como teórico-recursivo han sido el logro de distintas *indemostraciones* de este tipo: los teoremas de

incompletud, la solución negativa al décimo problema de Hilbert y distintas verdades aritméticas indemostrables en la aritmética de Peano ([26], [10], [22]). Las indemostraciones más generales y mejor conocidas son los teoremas de incompletud de los sistemas axiomáticos que definen *milagrosamente* la clase de funciones recursivas.

Es deseable que cualquier sistema formal produzca indemostraciones, por cuanto un sistema capaz de demostrar cualquier fórmula es inútil. Llamamos trivial a un sistema axiomático cuando es capaz de demostrar cualquier fórmula. Diremos que un sistema formal es negación-consistente si y sólo si contiene como teoremas algún par de fórmulas $\langle A, \neg A \rangle$ tal que cada una de ellas es la negación booleana de la otra. Esta terminología no es la habitual, dado que la lógica clásica no puede distinguir entre las nociones de negación-consistencia y trivialidad³. Cualquiera de ellas define consistencia en un sistema clásico. Cuando la diferencia entre ambas no importe, o se trate de sistemas clásicos, hablaremos sencillamente de consistencia.

Puesto que demostrar la consistencia de un sistema es producir una indemostración, (de hecho, clásicamente, *alguna*, indemostración) no es sorprendente que los sistemas axiomáticos estén en general mal dotados para demostrar axiomáticamente su propia consistencia. Los teoremas de incompletud de Gödel son consecuencia de requerir consistencia (o negación-consistencia): un enunciado indemostrable es precisamente el que enuncia este hecho, esto es, la oración aritmética que afirma de sí misma que es indemostrable (primer teorema). Otra oración indemostrable en un sistema consistente es la que enuncia la consistencia misma del sistema (segundo teorema). Si el sistema formal es consistente en un sentido particularmente fuerte (ω -consistente), entonces la oración que aserta su propia indemostrabilidad (llamada oración-Gödel) tampoco es refutable en el sistema. Posteriormente, Rosser construyó una oración más compleja tal que basta la sola consistencia del sistema para que tanto esa oración como su negación sean indemostrables. En la literatura constan otras oraciones que muestran que los sistemas formales consistentes clásicos no pueden demostrar otros resultados acerca de sí mismos. Es importante destacar que todas estas pruebas de indemostrabilidad no son iredursivas, sino ellas mismas pruebas efectivas. Nos remitimos a [23] para una exposición completa de pruebas y resultados de incompletud. A continuación indicamos someramente las definiciones precisas de los conceptos recién indicados, que serán requeridos en la argumentación ulterior:

Para cada número natural n , sea \mathbf{n} el numeral que expresa n . Nótese que \mathbf{n} es un término del lenguaje, mientras que n no lo es necesariamente. Como es habitual, entenderemos \mathbf{n} como el término $S^n(\mathbf{0})$, donde S es la función de sucesor y $\mathbf{0}$ el numeral que expresa el número 0. Para cada fórmula A , sea “ A ” su número de Gödel y para cada fórmula $A(x)$ con una variable libre x , sea “ $A(\mathbf{x})$ ” el número de Gödel de la fórmula resultante de substituir en A la variable libre x por el numeral que x represente. Esta es, por cierto, una formalización de la función que asigna a n el valor “ $A(\mathbf{n})$ ”.

Sea T una teoría o conjunto recursivamente axiomatizable de oraciones del lenguaje formal de la aritmética. Las fórmulas de este lenguaje pueden clasificarse de acuerdo a su estructura cuantificacional. Para nuestros intereses basta señalar que las fórmulas Π_1 son aquellas en la forma $\forall x P x$ donde P expresa una propiedad recursiva de los números naturales, y la dual de esta clase de fórmulas es la clase Σ_1 de fórmulas en la

³ En general, ninguna lógica cuya negación sea al menos tan potente como la negación intuicionista podrá distinguir entre ambas nociones, debido a la presencia del teorema $A \rightarrow (\neg A \rightarrow B)$ (reductio).

forma $\exists xPx$. Para expresar que x es (el número de Gödel de) una prueba en T de (el número de Gödel de) y , escribamos $\text{Dem}_T(x,y)$. $\text{Dem}_T(y)$ es entonces un predicado de demostrabilidad para T , que abrevia la fórmula $\exists x\text{Dem}_T(x,y)$. Si \perp es la negación booleana de cualquier teorema de T , la oración $\text{Cons}[T]$ es la fórmula $\text{Dem}_T(\perp)$, que expresa formalmente la consistencia de T . Siendo A cualquier fórmula del lenguaje de T , la oración $\text{Triv}[T]$ es la fórmula $\text{Dem}_T(A)$, que expresa formalmente la trivialidad de T , es decir, una abreviatura de la fórmula $\text{Dem}_T(\forall x\text{Dem}_T(x))$.

La ω -consistencia de T no es formalizable como una fórmula particular del lenguaje, sino como el esquema:

$$\text{Dem}_T(\exists xA(x)) \rightarrow \exists x\neg\text{Dem}_T(\neg A(x))$$

(para toda fórmula $A(x)$ con la sola variable x libre), que abreviamos con la expresión $\omega\text{-CONS}[T]$. Si el esquema se restringe a fórmulas $A(x)$ que sean primitivo-recursivas, capturamos una propiedad interesante que llamaremos PRconsistencia, formalizada por dicha restricción y abreviada con la expresión $\text{PRCONS}[T]$.

El primer teorema de incompletud afirma que para toda teoría ω -consistente T hay una oración G equivalente a $\neg\text{Dem}_T(G)$ que resulta ser indecidible en T . La PRconsistencia de T (y por tanto su ω -consistencia) asegura que G es indemostrable y su consistencia que G es indemostrable. Esto es,

$$\begin{aligned} \text{(T1)} \quad & \omega\text{-CONS}[T] \vdash \neg\text{Dem}_T(G) \wedge \neg\text{Dem}_T(\neg G) \\ \text{(T1')} \quad & \text{PRCONS}[T] \vdash \neg\text{Dem}_T(G) \wedge \neg\text{Dem}_T(\neg G) \end{aligned}$$

El segundo teorema de incompletud muestra que G es lógicamente equivalente a $\text{Cons}[T]$, esto es,

$$\begin{aligned} \text{(T2)} \quad & \text{Si } \text{Cons}[T], \text{ entonces } \text{Cons}[T, \neg\text{Cons}[T]] \\ \text{(T2')} \quad & \text{PRCONS}[T] \vdash \text{Cons}[T, \text{Cons}[T]] \end{aligned}$$

Se comprueba fácilmente que G implica $\neg O$, si es que O es equivalente a:

$$\exists x (\text{Dem}_T(x, \neg O)) \wedge \forall y < x \neg\text{Dem}_T(y, O)$$

A la oración $\neg O$ se la llama oración-Rosser (para T). Por (T1), $\neg O$ tampoco es demostrable en T si T es consistente. Además, O también es indemostrable a partir de $\text{Cons}[T]$. Es decir,

$$\text{(T3)} \quad \text{Si } \text{Cons}[T], \text{ entonces } \text{Cons}[T, \neg O]$$

Los esquemas siguientes reflejan una misma idea y se llaman, respectivamente, principio de reflexión y principio de reflexión uniforme (para T):

$$\begin{aligned} \text{REF} \quad & \text{Dem}_T(A) \rightarrow A \\ \text{REFU} \quad & \forall x (\text{Dem}_T(A(x)) \rightarrow A(x)) \end{aligned}$$

Sean REF/\prod_n (REF/Σ_n), REFU/\prod_n (REFU/Σ_n) las restricciones de esos principios a las clases de fórmulas $\prod_n(\Sigma_n)$. La importancia de los principios de reflexión para nosotros proviene del hecho que la consistencia y PRconsistencia de T resultan ser lógicamente equivalentes a las siguientes restricciones de principios de reflexión:

- (T4) $\text{Cons}[T] \equiv \text{REF}/\prod_1$
 (T4') $\text{PRCONS}[T] \equiv \text{REF}/\Sigma_1$

Estos conceptos y resultados serán empleados asiduamente en lo que sigue.

5. Antimecanicismo

Los resultados limitativos de la capacidad demostrativa de sistemas formales estuvieron filosóficamente motivados desde un principio, y también desde entonces han tenido aplicaciones extrametamatemáticas. Como ha ocurrido con otros resultados formales o informales, sus aplicaciones filosóficas han sido a menudo inadecuadas y erróneas. Entre éstas no pueden contar, al menos trivialmente, aquellas que dependen de la equivalencia entre los sistemas formales y máquinas recursivas, y por lo tanto entre éstos y los modelos computacionales del razonamiento y aún de la mente humana en general. De los teoremas en cuestión se deduce que si tales modelos son correctos, entonces no lo son demostrablemente (de manera recursiva). Esta conclusión puede interpretarse de distintas maneras, y una de ellas, inaugurada por el propio Gödel, ha sido interpretarla mostrando que las capacidades matemáticas humanas sobrepasan a las de cualquier máquina recursiva.

Ilustraremos ahora los argumentos antimecanicistas a partir de un caso particular de prueba irecursiva, como es la demostración por Gentzen de la consistencia de la aritmética de Peano de primer orden (AP). Esta demostración usa inducción sobre los ordinales transfinitos hasta ϵ_0 (que es el límite de los ordinales ω , ω^ω , ω^{ω^ω} ,...). Esta estrategia de prueba parece absurda, al demostrar que nuestros supuestos sobre el comportamiento de los números naturales es consistente apelando a una colección mucho mayor que la de estos números, como es la de todos los ordinales menores que ϵ_0 .

La razón por la que no es una estrategia absurda es que el tipo de inducción que se realiza es una inducción particularmente restringida, mientras que la clave para obtener la consistencia de la aritmética está precisamente en el carácter irrestricto del principio de inducción. Buena parte de la prueba de Gentzen es aritmetizable recursivamente. Ya que cada ordinal menor que ϵ_0 puede expresarse mediante un símbolo finito⁴ y la relación de orden entre tales símbolos también es finitamente definible, podemos interpretar la prueba sobre esos símbolos como códigos de los ordinales. En definitiva, interpretar la prueba sobre números naturales. El enunciado que en el lenguaje de AP expresa la inducción transfinita hasta cualquier ordinal menor que ϵ_0 es él mismo un teorema de AP. Por tanto, gran parte de la prueba es realizable idealmente por una máquina de Turing. Sin embargo, por supuesto, el enunciado aritmético que expresa codificadamente la

⁴ Puede expresarse finitamente en la llamada forma normal de Cantor como una suma de potencias de ω .

inducción transfinita hasta el propio ϵ_0 no es un teorema de AP, pues si lo fuese AP demostraría su propia consistencia, contradiciendo el segundo teorema de Gödel.

Pues bien, miles de cerebros humanos han seguido y comprendido esta prueba, cuya estrategia y conclusión no es, demostrablemente, emulable por una máquina de Turing. ¿Significa esto que tales cerebros tienen habilidades matemáticas que ninguna máquina de Turing tiene? No necesariamente, ya que en la naturaleza de cada prueba está inferir su conclusión sólo respecto de sus premisas, y en particular la prueba de la consistencia de la aritmética demuestra la consistencia relativa de ésta respecto de la consistencia de una parte substantiva de la teoría de conjuntos, como es la teoría de los ordinales infinitos. Como es bien sabido, e igualmente se deduce de los teoremas de incompletud, tampoco puede haber una prueba axiomática de tal consistencia. Lo que la prueba de Gentzen muestra es que el enunciado aritmético que codifica la inducción hasta ϵ_0 no es considerado verdadero en virtud de nuestra comprensión de los números naturales estándar, sino porque consideramos verdaderos otros enunciados acerca de ordinales. Un argumento antimecanicista ideal sería aquel que, per impossibilem, aportase una prueba no relativa de la validez de alguna prueba irecursiva.

Esta forma ideal de proceder no es la que, en la práctica, han adoptado los argumentos antimecanicistas basados en la no aplicación al razonamiento matemático humano de los límites que los teoremas de incompletud imponen a cualquier máquina recursiva. En su forma más básica ([12]) el argumento es como sigue:

1. Todo sistema formal consistente para la aritmética contiene oraciones indecidibles, esto es, pares $\{G, \neg G\}$ de oraciones del lenguaje del sistema que no son demostrables. Si, aunque no es esencial, nos limitamos a considerar lenguajes que contengan un predicado de demostrabilidad, entonces una de las oraciones del par $\{G, \neg G\}$ será una oración-Gödel que, bajo las convenciones estándar, se interpreta asertando su propia indemostrabilidad.
2. Puesto que esa oración es indecidible, es verdadera. Más aún, nosotros podemos emplear el procedimiento de Gödel para encontrar efectivamente esa oración y, por (1), deducir que es verdadera.
3. Puesto que, por el milagro, cada máquina recursiva es equivalente a un sistema formal axiomático para un fragmento de la aritmética, (2) muestra que nosotros podemos alcanzar conclusiones metamatemáticas que no están al alcance de ninguna máquina recursiva.

Esta es, en bruto, la llamada “falacia de Lucas”, así calificada en el calor de lo que han sido (y siguen siendo) décadas de agria polémica. El calificativo no nos resulta ecuánime, ya que el argumento no contiene un craso error lógico, sino más bien carencias de carácter conceptual que afectan gravemente a la veracidad de su conclusión respecto de (1). Es más justo considerar que el argumento es un non-sequitur en la forma que se presenta, por las razones siguientes:

Sólo si conocemos la consistencia de la aritmética sabemos que una de las oraciones $\{G, \neg G\}$ es verdadera. Adviértase que no basta aquí con que los razonadores humanos puedan asumir tal consistencia. Los teoremas limitativos nada estipulan acerca de lo que sistemas formales o máquinas de Turing pudieran *asumir*, sino acerca de lo que pueden *demostrar recursivamente*. Es una demostración de la consistencia de la

aritmética lo que aportaría un genuino contraejemplo lógico al modelo computacional recursivo. Y por lo mismo, no una demostración informal cualquiera, sino precisamente una demostración recursiva. Para que una mente presente una actividad que demostrablemente realice lo que ninguna máquina recursiva o axiomática puede hacer, la mente debe realizar una computación efectiva no realizable por la máquina, y no cualquier otra actividad cognitiva más o menos sublime.

Penrose ([16],[17]) cree aportar una respuesta a esta objeción⁵. Ciertamente, un argumento antimecanicista refinado debe distinguir entre las capacidades demostrativas de un sistema formal aritmético Γ_i y las capacidades del sistema que resulta de añadirle su oración-Gödel G_i como axioma adicional. Este es el equivalente formal de “asumir” que G_i es verdadera, esto es, más adecuadamente, el resultado con el que la máquina universal de Turing reproduce el argumento en (1) para comprobar que G_i es verdadera. Los teoremas de incompletud nos aseguran que añadir G_i como axioma no completa el sistema, sino que una nueva oración Gödel G_{i+1} es indecidible en el sistema $\Gamma_{i+1} = \{\Gamma_i, G_i\}$. La incompletud es, en palabras de Putnam [20] “esencial” en el sentido que el simple añadido de nuevas fórmulas al sistema no altera su incompletud. Esta operación puede iterarse al infinito, de manera que el argumento antimecanicista refinado debiera constatar que la máquina es incapaz de reproducir esta operación para algún ordinal transfinito, y por tanto de “seguir” a la mente en la secuencia: $\Gamma_i, \Gamma_{i+1}, \dots, \Gamma_\omega, \Gamma_{\omega+1}, \dots$

De este modo el problema original se convierte en un problema acerca de las progresiones infinitas de sistemas formales. Puesto que no existe un procedimiento recursivo para acceder a todos los ordinales, el alcance de la máquina está limitado a ordinales infinitos recursivos. Feferman y Spector [4] demostraron que la progresión sigue siendo incompleta por más que se añadan recursivos infinitos enunciados de consistencia. Por tanto *esa* carrera entre la máquina y la mente es una carrera a ninguna parte.

Lo que es más, las limitaciones de la máquina en *esa* progresión no son, respecto de la consistencia, mayores que las de la mente, dado T4 y la progresión recursivo-infinita considerada por Turing en [28], que es completa respecto de las oraciones \prod_1 . Una progresión diferente es la progresión infinita (no sólo recursivo-infinita) que itera no oraciones Gödel, sino el principio uniforme de reflexión (REFU, ver p. 12). En [4] también se demuestra que una progresión semejante contiene todas las oraciones aritméticas verdaderas. Por supuesto, que la mente pueda imaginarse realizando tal progresión no asegura su consistencia y está sujeto a las mismas consideraciones que mereció la prueba de Gentzen.

De todo ello se sigue que el argumento antimecanicista no resiste la objeción. De hecho, que el razonamiento matemático humano tenga habilidades axiomáticas no emulables recursivamente no es consecuencia lógica de los teoremas de incompletud y de la existencia de verdades aritméticas no demostrables en AP .

6. Consistencia y reflexión

Lo cual no significa que la conclusión del argumento antimecanicista sea falsa, ni tan siquiera que sea irrazonable. Una refutación filosóficamente fructífera del argumento

⁵ Desarrollando una idea bosquejada ya por Lucas en [12].

no debe contentarse con objeciones a su validez deductiva. Debe, en definitiva, también explicar o acomodar las intuiciones que lo motivan.

La clave de éstas está, en nuestra opinión, en las razones creamos tener para *saber* que nuestra comprensión de los números naturales es consistente. Es en esta pieza de “conocimiento” en la que descansa lógica y conceptualmente el argumento antimecanicista, y lo hace además necesariamente (pues de otro modo la premisa (2) del argumento carece de fundamento). La sola *confianza* en la teoría elemental de los números naturales estándar no nos regala gratuitamente su consistencia. La “falacia” atribuida a Lucas reside en pasar por alto la riqueza informacional y lógica que implica no sólo demostrar finitariamente consistencia, sino incluso simplemente suponerla (como tal, la “falacia” no es sólo ni especialmente cometida por antimecanicistas). Estas afirmaciones se argumentan a continuación.

Sea T una teoría axiomática lo suficientemente rica como para representar las funciones primitivo recursivas. Podemos demostrar finitamente por medios puramente axiomático recursivos que T satisface propiedades análogas a la consistencia, lo cual es mucho más de lo que suele reconocerse ante teoremas de incompletud. En particular, y sorprendentemente, puede demostrarse por tales medios finitos que T no es trivial, si sus axiomas aritméticos son los convencionales y sus axiomas lógicos los de distintas lógicas no clásicas. Esto es, demostrar que la fórmula $\neg\text{Triv}[T]$ sí es un teorema de T (cfr.[14],[22]). El precio de este valioso resultado axiomático es, desde luego, la renuncia a considerar válidas formas de inferencia por reducción al absurdo que son clásicamente válidas como condicionales materiales (pero no como implicaciones). Naturalmente, al ser T tan expresiva como aritméticas clásicas, es imposible demostrar recursivamente que cualquier extensión de T sea negación-consistente: dado (T, T') , la fórmula $\text{Cons}[T]$ no es un teorema de T . No es obvio que la confianza en el cuerpo de prácticas aritméticas informales tenga que asegurar más su no trivialidad que su negación-consistencia y, manifiestamente, el argumento antimecanicista no es planteable siquiera en términos de no trivialidad.

Volvamos a la acepción convencional de consistencia para una teoría clásica T que sea un fragmento de AP capaz de representar las funciones (primitivo) recursivas. Se trata de aportar un segundo argumento a la idea de esta sección, motivado por la reflexión siguiente. Dado que no hay procedimientos sintácticos para demostrar la consistencia de T finitariamente, pudieramos, dada la confianza que T nos merece al formalizar la práctica aritmética, simplemente estipularla o suponerla semánticamente. Creeríamos, en esencia, que la aritmética formal es consistente al ser *verdadera* o, más propiamente, correcta o semánticamente consistente⁶. Veamos el alto precio de este supuesto.

La afirmación siguiente se sigue de las condiciones recién presentadas, asegurándonos que podemos, en tales condiciones, añadir el predicado de consistencia a fórmulas o clases de fórmulas que consideramos verdaderas:

$$(L5) \quad T, \text{Cons}[T] \vdash \text{Cons}[T, \text{Cons}[T]]$$

⁶ Penrose corrige en [17] su mal uso en [16] del concepto de corrección.

que se justifica razonando sencillamente: De la hipótesis a reducir: $\neg \text{Cons}[T, \text{Con}[T]]$, se deduce, dadas las propiedades de \vdash clásica: $T \vdash \neg \text{Cons}[T]$. Pero, siendo T verdadera, de ello se sigue la contradicción $\text{Cons}[T] \wedge \neg \text{Cons}[T]$.

Adviértase que el supuesto T es necesario, ya que en general la regla de inferencia:

$$\text{Cons}[T] \vdash \text{Cons}[T, \text{Cons}[T]]$$

no es consistente, en virtud de los teoremas de incompletud, con ninguna teoría T' suficientemente expresiva. En efecto, sea 'T cualquier teoría tal que $T' \vdash \text{Cons}['T]$, y sea $T = \{ 'T, \neg \text{Cons}['T] \}$. T es consistente por los teoremas de Gödel, y sin embargo $\{ T, \text{Cons}[T] \}$ no lo es, ya que T demuestra que un fragmento de T que no es consistente.

Ahora bien, la consistencia es lógicamente equivalente al principio de reflexión señalado en (T4). Pero (L5) supone, por definición de T y el tipo de fórmulas involucradas, al menos, el principio REF/ Σ_1 (ver p.12 de nuevo). Y éste, (T4'), es equivalente a la PR-consistencia del sistema. Por lo tanto, se está suponiendo un principio más fuerte que la propia consistencia.

Esto muestra cierta curiosa falta de monotonicidad en el razonamiento puramente matemático: suponer la consistencia de verdades ya supuestas comporta información y estructura lógica adicional que puede llegar a ser intolerable.

REFERENCIAS

- [1] Anderson, Belnap y Dunn, *Entailment. The Logic of Relevance and Necessity*, vol. II. Princeton University Press, 1992
- [2] Blum, Shub y Smale, *On a theory of computation and complexity over the real numbers*, Bull. AMS 21 (1989), 1-46
- [3] Copeland, *Turing's O-machines, Searle, Penrose and the Brain*, Analysis 58 (1998), 128-38
- [4] Feferman y Spector, *Transfinite recursive progressions of axiomatic theories*, Journal of Symbolic Logic 27 (1962), 259-316
- [5] Feferman, *Gödel's program for new axioms: Why, where, how and what?* En Hájek (ed.) *Gödel 96*, Springer, 1996
- [6] Feferman, *Penrose's gödelian argument*, Psyche 2 (7), 1995
- [7] Gödel, *Collected Works I*. Oxford University Press, 1986
- [8] Gödel, *Collected Works II*. Oxford University Press, 1990
- [9] Gödel, *Collected Works III*. Oxford University Press, 1995
- [10] Kirby y Paris, *Accesible independence results for Peano Arithmetic*, Proc. London Mathem. Soc. 14 (1982), 285-93
- [11] Kreisel, *A notion of mechanistic theory*, Synthese 29 (1974), 11-26
- [12] Lucas, *Minds, machines and Gödel*, Philosophy 36 (1961), 112-27
- [13] Maas y Orponen, *On the effect of analog noise in discrete time*, Manuscrito 1997
- [14] Meyer, *Relevant arithmetic*, Bulletin of the Section of Logic 5 (1976), 5.
- [15] Meyer, *$\supset E$ is admissible in "true" relevant arithmetic*, Manuscrito 1994, publicado Journal of Philosophical Logic, (1998)

- [16] Penrose, *Shadows of the mind: a search for the missing science of consciousness*, Oxford University Press, 1994
- [17] Penrose, *Beyond the doubting of a shadow*, Psyche, 2, 23
- [18] Pour-El y Richards, *Computability in analysis and physics*, Springer 1989
- [19] Pour-El y Zhong, *The wave equation with computable initial data whose unique solution is nowhere computable*, CAA 97, Springer 1997
- [20] Putnam, *Incompleteness and essential incompleteness*, Journal of Symbolic Logic 22 (1957), 39-54
- [21] Putnam, *Review: Shadows of the mind*, Bull. A.M.S. 32 (1995), 360-73
- [22] Salto y Méndez, *Relevant recursiveness*, pendiente publicación
- [23] Salto y Méndez, *Un vademécum de incompletud*, en redacción
- [24] Siegelman y Sontag, *On computational power of neural networks*, Journal of Comput. Syst. Sci. 50 (1995), 132-150
- [25] Siegelman, *Computation beyond the Turing limit*, Science 268 (1995), 545-8
- [26] Smorynski, *The varieties of arboreal experience*, Mathematical Intelligencer 4 (1982), 182-9
- [27] Turing, *On Computable Numbers with an application to the Entscheidungsproblem*, Proc. London Math. Soc, 2,42 (1937),230-65
- [28] Turing, *On computable numbers with an application to the Entscheidungsproblem: a correction*, Proc.London Mathematical Soc., 2, 42 (1936-7), 230-65
- [29] Turing, *A system of logic based on ordinals*, Proc. London Mathematical Soc., 2, 45 (1939), 161-228
- [30] Vázquez, *La formalización de sistemas*. Tesis Universidad de Salamanca, 1990
- [31] Wegner, *Why interaction is more powerful than algorithms*, ACM 40, (1997), 80-91
- [32] Weihrauch, *A refined model of computation for continuous problems*, Journal of Complexity 14 (1998), 102-21
- [33] Wright, *Intuitionists are not Turing Machines*, Philosophia Mathematica 3 (1995), 86-102.