Informática Educativa Vol. 6 Nº 2, 1993
Proyecto SIIE, Colombia pp.121-132

NATURALEZA Y LÍMITES DE LA CALCULABILIDAD: ¿POR QUÉ CARECIENDO DE INTELIGENCIA, SON TAN EFECTIVOS LOS COMPUTADORES?

Jaime BOHÓRQUEZ

RESUMEN

Se presenta una perspectiva histórica del arte de calcular, ilustrando cómo en su naturaleza mecánica radica tanto su efectividad como sus limitaciones. Sin la interpretación última de un observador (humano) esta actividad se reduce simplemente a la realización de manipulaciones simbólicas carentes de significado.

INTRODUCCIÓN

El hombre ha calculado desde tiempos inmemoriales. El calcular presupone, por un lado, un acto de abstracción (simbolización) de algún concepto de interés (v.gr., la noción de cantidad, de espacio, los razonamientos lógicos, el funcionamiento de un sistema, los métodos mismos de cálculo, etc.). Por otra parte, también presupone métodos mecánicos de manipulación de los símbolos producto de la abstracción, que permiten obtener respuestas a preguntas referentes al concepto de interés, sin que necesariamente seamos conscientes todo el tiempo del significado de los símbolos manipulados.

Por ejemplo, el concepto de número 3 proviene de la abstracción de la propiedad común a todos los conjuntos que poseen exactamente tres miembros. Entre los sistemas de representación de números naturales, tal vez el más sencillo conceptualmente (aunque no necesariamente el más práctico) es el llamado sistema monario, donde el número

uno se representa con	|	(un palote)
el número dos con	||	(dos palotes)
el número tres con	|||	(tres palotes)

etc.

Con este sistema de representación símbólica, el algoritmo para sumar dos números naturales consiste simplemente en la yuxtaposición de los sumandos:

```
||||  +  |||  =  |||||||
```

Varios siglos le tomó a la humanidad desarrollar el sistema actual de representación posicional de los números. Considere de nuevo el algoritmo para sumar dos números naturales, pero esta vez usando la representación decimal usual.

Por ejemplo

```
    3 2 9
+   4 4 3
  -------
    1 2
```

```
  7 6
 ---
 7 7 2
```

El algoritmo para esta operación nos fue enseñado de forma brutal (!), es decir, de manera mecánica y sin consideración de su significado y mucho menos de su corrección. No obstante, para explicar su significado y corrección, habría que referirse a propiedades (algebráicas) tanto de la adición como de la multiplicación (como la asociatividad y la distributividad) de los números naturales. Veamos:

$$
\begin{array}{rrcrcrcr}
 & 3x10^2 & + & 2x10 & + & 9 & = & 329 \\
+ & 4x10^2 & + & 4x10 & + & 3 & = & 443 \\
\hline
 & & + & 1x10 & + & 2 & & \\
 & 7x10^2 & + & 6x10 & & & & \\
\hline
 & 7x10^2 & + & 7x10 & + & 2 & = & 772
\end{array}
$$

Sin embargo, cuando cuadramos nuestra chequera, ya sea a mano o con la ayuda de una calculadora, estamos contentos con no tener esta carga adicional para nuestro cerebro.

La noción de significado en matemáticas (semántica) es siempre relativa: explicar un concepto consiste en, de alguna manera, representarlo en términos de otro que se considera más sencillo o familiar. El proceso de comprender o entender corre por cuenta del observador (ser inteligente) que realiza los cálculos. El definir una semántica (significado) para un sistema formal o de cálculo da seguridad al que realiza los cálculos sobre el sentido (buena definición) o propósito de dicho sistema. Esta noción parece ser exclusiva de la mente humana, pues no tiene ningún sentido hablar de que un proceso o mecanismo simbólico necesite conocer el significado de las manipulaciones simbólicas que efectúa.

LO ABSTRACTO Y LO CONCRETO

Los números y su correspondiente notación decimal son para la mayoría de nosotros una y la misma cosa. Pensamos en ellos no como abstracciones, sino como cosas concretas. Sin embargo, de algo tan concreto como la notación posicional de los números naturales, v.gr.,

$$
\begin{array}{rrcrcrcr}
 & 329 & = & 3x10^2 & + & 2x10 & + & 9 \\
+ & 443 & = & 4x10^2 & + & 4x10 & + & 3 \\
\hline
 & 772 & = & 7x10^2 & + & 6x10 & + & 12
\end{array}
$$

podemos pasar (abstrayendo la base decimal) a entes llamados polinomios como

$$
\begin{array}{rcrcr}
3x^2 + & & 2x & + & 9 \\
4x^2 + & & 4x & + & 3 \\
\hline
7x^2 + & & 6x & + & 12
\end{array}
$$

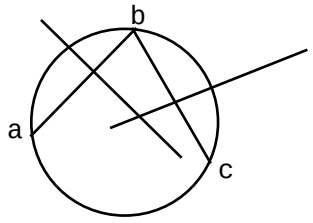
donde "x" es un símbolo que representa en principio cualquier valor numérico.

Las operaciones entre polinomios, tales como la adición, nos fueron enseñadas también de manera mecánica en la escuela secundaria. Sólo después de algún tiempo, necesario para familiarizarnos con estos conceptos, comprendimos la utilidad de estos entes para resolver problemas mediante la proposición de ecuaciones.

LA NOCIÓN DE ALGORITMO

Los métodos de cálculo mismos han logrado estudiarse abstractamente mediante entes llamados algoritmos. La palabra *algoritmo* proviene del nombre del matemático persa Abu Jafar Mohammed ibn Mûsâ *al-Khouârizm* quien escribió un texto de matemáticas alrededor del año 825 DC, llamado "Kitab al jabr w'al-muqabala" título del cual proviene la palabra álgebra. Los algoritmos, no obstante, se conocían mucho antes de la aparición del texto de al-Khouârizm.

Calcular no necesariamente implica manipular números o cantidades. Los griegos, por ejemplo, realizaban cálculos relativos al concepto de espacio mediante construcciones geométricas de regla y compás.Un problema típico para esta clase de cálculos es el siguiente: dados tres puntos no colineales en el plano, es posible construir, con regla y compás, una circunferencia que pase por dichos puntos:



De hecho, aun para efectuar cálculos aritméticos, los griegos realizaban construcciones geométricas. El llamado algoritmo de Euclides para hallar el máximo común divisor de dos números naturales se explicaba en estos términos.

Otra área de interés, susceptible de ser estudiada abstractamente para obtener un mecanismo simbólico (léase: sistema de cálculo), es la *lógica*, en lo que a sistemas de deducción se refiere. Las correspondientes abstracciones son denominadas sistemas formales. Entre estos se destacan los *sistemas de producción de Post*:

El sistema `MIU` (D. R. Hofstadter)

```
Axioma    :                MI
Regla I   :    xI      →   xIU
Regla II  :    Mx      →   Mxx
Regla III :    xIIIy   →   xUy
Regla IV  :    xUUy    →   xy
```

Derivación del teorema `MUIIU`

```
(1)  MI          , Axioma
(2)  MII         , por Regla II
(3)  MIIII       , por Regla II
(4)  MIIIIU      , por Regla I
(5)  MUIU        , por Regla III
(6)  MUIUUIU     , por Regla II
(7)  MUIIU       , por Regla IV
```

Una secuencia bien definida de operaciones se conoce con el nombre de *algoritmo*. La formulación precisa del concepto general de algoritmo data solamente de este siglo. Varias descripciones alternativas de este concepto fueron desarrollados en la década de los años treinta:

- las máquinas de Turing
- los sistemas de producción de Post
- las funciones recurrentes generales (Hilbert y Ackermann)
- el cálculo Lambda (Church y Kleene)
- los algoritmos de Markov

Todos estos esquemas de cálculo fueron propuestos independientemente, y más o menos simultáneamente, con el ánimo de capturar matemáticamente la noción intuitiva de cálculo o procedimiento mecánico. Todos estos esquemas fueron demostrados completamente equivalentes.

TESIS DE CHURCH-TURING

> El concepto de máquina de Turing (o cualquiera de sus equivalentes) define matemáticamente la noción intuitiva de procedimiento algorítmico (o efectivo, o recurrente o mecánico).

En la actualidad casi nadie cuestiona esta tesis.

EL PROGRAMA DE HILBERT

El concepto de máquina de Turing fue desarrollado por el matemático inglés Alan Turing en los años 1935-36 para atacar un problema planteado por el gran matemático David Hilbert en 1928, *(Entscheidungsproblem)* que pedía un procedimiento algorítmico general para resolver, sistemáticamente, todos los problemas de la matemática (pertenecientes a una clase adecuadamente bien definida); o mejor, una respuesta a la pregunta de sobre si tal procedimiento podría, *en principio*, existir.

Hilbert quería encontrar un esquema comprensivo que incluyera todos los tipos correctos de razonamiento para cualquier área matemática. El creía que podría ser posible demostrar que tal esquema estaría libre de contradicciones, y de esta forma se proporcionaría a las matemáticas una sólida y segura fundamentación.

El punto de vista de que es posible prescindir de los significados de las proposiciones matemáticas, considerándolas como nada más que secuencias de símbolos de algún sistema matemático formal es adoptado por el *formalismo*. Con esta visión, las matemáticas podrían considerarse como un juego simbólico sin ningún significado; sin embargo, el matemático austriaco Kurt Gödel propinó al formalismo, mediante un resultado asombroso, demostrado en 1931, un golpe devastador que echó por tierra las esperanzas de Hilbert. De hecho, es el "significado", y no la computación algorítmica ciega, lo que da tanto a las matemáticas como a la informática su sustancia.

LAS MÁQUINAS DE TURING

Parte de la dificultad en responder la pregunta de Hilbert consistía en decidir qué se quiere decir con procedimiento mecánico. Turing trató de imaginar cómo se podría formalizar el concepto de máquina manipuladora de símbolos.

Una *máquina de Turing* es un esquema de cálculo que consta de una cinta infinita en ambas direcciones que posee un número infinito de celdas que pueden alojar cada una un símbolo tomado de un cierto alfabeto; por ejemplo, el que consta solamente de los símbolos `0`, `1` y el blanco (representado por el símbolo "`-`"). Posee además, por un lado, un conjunto finito de estados posibles que puede adoptar, entre los que se cuentan estados iniciales y finales o de terminación; y por otro, una "cabeza lectora" que se posa sobre una celda de la cinta y puede desplazarse hacia la derecha o a la izquierda de la cinta, así como inspeccionar o modificar el contenido de la celda donde se halle posada.

La decisión de modificar el contenido de la celda inspeccionada, o desplazarse una celda hacia la derecha (`R`) o hacia la izquierda (`L`), así como la de adoptar un nuevo estado está dada por un conjunto finito de instrucciones (el *programa)* que, con base en el estado vigente y el símbolo en la celda inspeccionada, ordena qué nuevo estado adoptar y qué acción tomar sobre la cinta dentro de las contempladas.

Por ejemplo:

Informática Educativa, **6** (2), 1993

↑
q

Estados: i (inicial), f (final), q,...

Programa:

```
i,0    →    q,0
i,1    →    q,1
q,1    →    q0,0
q,0    →    f,0
q0,0   →    q,L
```

Por convención, los datos se colocan a la derecha de la cabeza lectora en el estado inicial y los resultados se recolectan también a la derecha de la cabeza lectora, en un estado final, cuando la máquina se detenga. Ya que es posible usar los números naturales para codificar cualquier otro conjunto de símbolos, no perderemos generalidad, para estudiar abstractamente las máquinas de Turing, si pensamos que cada una de ellas permite calcular una función numérica definida parcialmente sobre los números naturales.

De hecho, debido a las propiedades mismas de codificación de los números naturales, es posible encifrar con un solo número la descripción completa de una máquina de Turing. Este número se podrá usar como *índice* o identificación de la función calculada por una máquina de Turing. Si un número cualquiera no encifra la descripción de ninguna máquina de Turing, supondremos por convención que encifra la función que no está definida para ningún valor. De esta manera, podemos actuar como si todo número natural fuera el índice o encifrara una *función Turing-calculable*.

EL PROBLEMA DE LA PARADA

Para solucionar el *Entscheidungproblem*, Turing se dio cuenta de que podía frasear su versión de la pregunta en términos del problema de decidir si la n-sima máquina de Turing (aquella con índice n) se detendría alguna vez al actuar sobre el número m. Este problema es conocido como el *problema de la parada*. Una solución al problema de la parada permitiría establecer la verdad sobre dos conjeturas famosas de la teoría de números, como el llamado "último teorema de Fermat" y la conjetura de Goldbach.

El famoso enunciado conocido como "último teorema de Fermat", hecho en la margen de la *Aritmética* de Diofanto, por el gran matemático del siglo XVII Pierre de Fermat, es la afirmación de que la ecuación:

$$(x + 1)^{w+3} + (y+1)^{w+3} = (z+1)^{w+3}$$

no se satisface para *ningún* conjunto de números naturales w, x, y, z. Fermat aseguraba poseer "una prueba verdaderamente formidable" de su afirmación, que no cabía en la

margen. Hasta el día de hoy nadie ha podido reconstruir tal prueba, ni por otro lado, encontrar un contraejemplo a la afirmación de Fermat [1].

Claramente, dada una cuádrupla de números (w, x, y, z), es meramente cuestión de calcular, para decidir si la ecuación se satisface o no. Por tanto, podríamos imaginar un algoritmo de computador que ensayara con todas las cuádruplas de números, una tras otra, y se detuviera solamente cuando la ecuación se satisficiera. Si pudiéramos establecer que el algoritmo que se acaba de describir no para, tendríamos una prueba de la afirmación de Fermat.

De manera similar, es posible refrasear otros problemas no resueltos de la matemática en términos del problema de la parada de Turing. En particular, la "conjetura de Goldbach", que afirma que todo número par mayor que 2 es la suma de dos números primos. Podríamos diseñar una máquina de Turing que pasara sobre los números pares 6, 8, 10, 12, 14, ... ensayando todas las diferentes formas de descomponerlos en pares de números impares

$$6 = 3 + 3, \quad 8 = 3 + 5, \quad 10 = 3 + 7 = 5 + 5,$$
$$12 = 7 + 5, \quad 14 = 3 + 11 = \ldots,$$

verificando que *cada uno* de tales números pares se descompone en *algún* par de números primos. Nuestra máquina se detendría solamente cuando llegara a un número par que no pudiera ser descompuesto como la suma de dos números primos. En tal caso, tendríamos un contraejemplo a la conjetura de Goldbach. En consecuencia, si pudiéramos decidir si una máquina de Turing se detiene alguna vez, tendríamos también una forma de decidir la conjetura de Goldbach.

LA INSOLUBILIDAD DEL PROBLEMA DE LA PARADA (Y DEL PROBLEMA DE HILBERT)

Suponga que existe alguna máquina de Turing H que "decide" si la n-sima máquina de Turing (T_n) para, eventualmente, cuando actúa sobre el número m. Es decir H es una función Turing-calculable tal que

$$H(n,m) = \begin{cases} 0 & \text{si } T_n(m) \text{ nunca se detiene;} \\ 1 & \text{si } T_n(m) \text{ eventualmente se detiene} \end{cases}$$

Por la tesis de Church-Turing la siguiente también sería una función Turing-calculable:

$$B(n) = \begin{cases} T_n(m)+1 & \text{si } H(n,m) = 1; \\ 0 & \text{si } H(n,m) = 0 \end{cases}$$

1 La conjetura fue resuelta afirmativamente en junio de 1993.

por tanto $B = T_b$ para algún índice b. Observe además, que B (al igual que H) se detiene al actuar sobre cualquier valor numérico. En consecuencia $H(b,b) = 1$ (ya que $T_b(b) = B(b)$ se detiene). Sin embargo,

$$T_b(b) = B(b) = T_b(b) + 1,$$

lo que claramente es contradictorio. Luego, una tal máquina de Turing H no existe! No hay un algoritmo universal para decidir si una máquina de Turing eventualmente se detendrá o no.

La pregunta de si una máquina de Turing particular se detendrá, o no, es una noción matemática perfectamente bien definida. Por lo tanto, al mostrar que no existe algoritmo para decidir la parada de las máquinas de Turing, Turing mostró (como también lo hizo Church con su cálculo lambda) que no podía haber algoritmo general para responder preguntas matemáticas. El *Entscheidungsproblem* no tiene solución!

Esto no es lo mismo que decir que en cualquier caso una máquina de Turing *individual* para; es posible que con algo de ingenio, o simplemente de sentido común, se pueda decidir tal pregunta en un caso dado. Pero no hay algoritmo que funcione para todas las preguntas matemáticas, ni para todas las máquinas de Turing y todos los números sobre los cuales pudieran actuar.

EL MOMENTO DE LA VERDAD

"¿Qué es la verdad? ¿Cómo formamos nuestros juicios sobre lo que es verdad y lo que no es verdad en el mundo? Seguimos simplemente algún algoritmo -sin duda más favorecido que otros algoritmos posibles menos efectivos, por el poderoso método de la selección natural? O podría haber otra ruta posiblemente no algorítmica -quizás intuición, instinto o visión interior- para adivinar la verdad? Esta parece una pregunta difícil."

Roger Penrose en "The Emperor's New Mind"

Como ya fue mencionado, es posible expresar simbólicamente (formalmente) proposiciones matemáticas. Podemos expresar afirmaciones tales como el "último teorema de Fermat":

$$\neg\exists w,x,y,z\ [(x + 1)^{w+3} + (y + 1)^{w+3} = (z + 1)^{w+3}]$$

La proposición anterior se lee (terminando en el primer paréntesis cuadrado):

"No es el caso que existan números naturales w,x,y,z tales que...".

También es posible reescribir el último teorema de Fermat usando $\forall$:

$$\forall w,x,y,z\ [\neg\ (x + 1)^{w+3} + (y + 1)^{w+3} = (z + 1)^{w+3}]$$

que se lee (terminando después del símbolo "no" a continuación del primer paréntesis):

"Para todos los numeros naturales w,x,y,z no es el caso que...",

que es lógicamente la misma cosa que la primera expresión.

Para denotar proposiciones matemáticas completas usaremos letras mayúsculas `P, Q, R, ….` Una proposición tal podría, de hecho, ser la afirmación de Fermat ya mencionada:

$$F = \neg\exists w,x,y,z\ [(x + 1)^{w+3} + (y + 1)^{w+3} = (z + 1)^{w+3}]$$

Una proposición podría depender de una o más variables; por ejemplo, podríamos estar interesados en la afirmación de Fermat para alguna potencia particular `w+3`:

$$G(w) = \neg\exists x,y,z\ [(x + 1)^{w+3} + (y + 1)^{w+3} = (z + 1)^{w+3}]$$

de tal manera que `G(0)` afirma que "ningún cubo puede ser la suma de cubos positivos", `G(1)` afirma lo mismo para las cuartas potencias, y así sucesivamente. La afirmación de Fermat es ahora que `G(w)` vale para todo `w`:

$$F = \forall w\ [G(w)]$$

`G(w)` es un ejemplo de lo que se llama una *función proposicional* , o sea, una proposición que depende de una o más variables.

EL TEOREMA DE GÖDEL

La idea del programa de Hilbert era encontrar, para cualquier área bien definida de las matemáticas, una lista de axiomas y reglas de inferencia suficientemente comprensiva para que *todas* las posibles formas de razonamiento matemático correctas, apropiadas a esa área, fueran incorporadas.

Gödel mostró que si un sistema formal es suficientemente poderoso como para expresar proposiciones de la aritmética, entonces tal sistema de axiomas y reglas es *incompleto*, en el sentido de que le es imposible al sistema decidir de manera general la verdad o falsedad de cualquier proposición matemática que pueda ser formulada dentro del mismo.

La esperanza de Hilbert era que para cualquier secuencia finita de símbolos que representara una proposición matemática, digamos `P`, uno pudiera probar `P`, o bien `¬P`, dependiendo de que `P` fuera verdadero o falso. Si esto fuera cierto, contando con los métodos mecánicos que proporciona el sistema formal, se podría prescindir completamente y sin ningún problema, del significado de las proposiciones, pues bastaría asignar el valor *verdadero* a la secuencia de símbolos que representa a `P`, si `P` es un teorema -o sea, si `P` es demostrable dentro del sistema- o el valor *falso*, si `¬P` es un teorema.

En parte, el argumento de Gödel es muy detallado y complejo. La idea central, por otro lado, es simple, profunda y hermosa. La parte complicada -que requería también de mucho ingenio- consistía en mostrar en detalle cómo codificar las reglas de producción de un sistema formal mediante operaciones aritméticas. Es decir, lograr expresar, mediante operaciones aritméticas, afirmaciones sobre el sistema formal de la aritmética misma!

Para efectuar esta codificación se asigna un índice (número) a toda secuencia finita de símbolos del sistema formal. Nos interesan particularmente las funciones proposicionales que dependen de una sola variable (tal como $G(w)$ mencionada anteriormente). Sea $P_n(w)$ la n-sima función proposicional aplicada al número w.

Las secuencias de proposiciones que constituyen una prueba de algún teorema del sistema, pueden también codificarse con números naturales. Supongamos que Π_n denota la n-sima prueba. Considere ahora la función proposicional que depende del número natural w:

$$\neg\exists x\ [\Pi_x \text{ pruebe } P_w(w)].$$

Aunque la proposición entre paréntesis cuadrados está dada parcialmente en palabras, es una proposición perfecta y precisamente definida. Afirma que la x-sima proposición es una prueba de la proposición que resulta de aplicar $P_w(.)$ al valor w mismo. A tal función proposicional debe corresponderle un número que la codifica, digamos k y por tanto:

$$\neg\exists x\ [\Pi_x \text{ pruebe } P_w(w)] = P_k(w).$$

Examinemos ahora esta función para el caso particular en que w toma el valor k. Obtenemos:

$$\neg\exists x\ [\Pi_x \text{ pruebe } P_k(k)] = P_k(k).$$

La proposición $P_k(k)$ es una afirmación matemática perfectamente bien definida. ¿Tiene una demostración dentro del sistema? ¿Tiene su negación $\neg P_k(k)$ una prueba? La respuesta a estas preguntas es no(!), pues si observamos con cuidado, $P_k(k)$ afirma su propia indemostrabilidad.

De hecho, como no existe una demostración para $P_k(k)$, entonces $P_k(k)$ es una proposición verdadera. ¡Hemos encontrado una proposición verdadera que no tiene prueba dentro del sistema!

¿Qué podemos decir de su negación $\neg P_k(k)$? Acabamos de establecer que $\neg P_k(k)$ es falsa -puesto que $P_k(k)$ es verdadera- y no se supone que debamos poder probar proposiciones falsas dentro del sistema! Por lo tanto, ni $P_k(k)$ ni $\neg P_k(k)$ son demostrables dentro del sistema. Esto establece el teorema de Gödel.

Informática Educativa, **6** (2), 1993

CONCLUSIONES

Las apreciaciones anteriores ilustran ampliamente tanto el poder como las limitaciones de los métodos formales. Obsérvese que, como lo muestra el resultado de Gödel, el sistema formal de la aritmética no puede establecer la verdad o falsedad de una proposición matemática expresada en el sistema mismo. Sin embargo, nosotros, como seres inteligentes, pudimos establecer sin dificultad que, en efecto, era verdadera, pues pudimos reflexionar sobre su *significado*!

Esta es una de las diferencias fundamentales entre el modo de funcionamiento de la mente humana y el de un sistema formal. No tiene sentido hablar de que un sistema formal sea consciente del significado de las manipulaciones simbólicas que realiza. El dotar a un sistema de estos con una semántica formal o con un metasistema que describa el sistema original, simplemente transfiere la pregunta al sistema formal añadido.

Sin embargo, la utilidad y eficacia de los sistemas formales es tan formidable y sorprendente que mucha gente ha sucumbido a la tentación de adscribirles (a través de su realización material: los computadores) cualidades netamente humanas -como la inteligencia- que de todas formas carecen de una definición precisa (es decir ¡formal!).

REFERENCIAS

HOFSTADTER Douglas, GÖDEL, Escher (1980) *Bach: An Eternal Golden Braid*. Vintage Books.

PENROSE Roger (1989). *The Emperor's New Mind.* Oxford: Oxford University Press.