PROBLEMAS ALGORITMICOS EN LAS MATEMATICAS

por

E. BURGER
Universidad de Colonia (Köln)

1. INTRODUCCION

Existe una gran clase de problemas matemáticos de la forma siguiente :

Hallar un procedimiento algorítmico para determinar si dada una propiedad T y un número natural n, n tiene la propiedad T o no. Un ejemplo completamente trivial de este género es el siguiente : ¿ Cómo se puede determinar si un número natural n es divisible por 3 ?. Para contestar esta pregunta se puede proceder de la manera usual: Escribiendo el número n en su representación decimal, proceder según las reglas del bien conocido algoritmo de división por 3 y hacer constar si esta división termina con el resto 0 o no. Evidentemente, es este un procedimiento algorítmico en el sentido intuitivo de este término. Otro procedimiento algorítmico para decidir la divisibilidad de un número por 3 se basa en la aplicación iterativa de la bien conocida regla de la "suma transversal" : n es divisible por 3 si y sólo si su suma transeversal es divisible por 3. Se forman las sumas transversales sucesivas hasta obtener un número x menor que 10, y entonces mirar si x es igual a 0, 3, 6, 9 o no.

Estos dos ejemplos muestran claramente cuál es el carácter decisivo del concepto de algoritmo. Primeramente, el procedimiento ha de ser de naturaleza general, es decir : debe ser aplicable a cada número natural n arbitrariamente prefijado.

En segundo lugar, el procedimiento ha de ser de naturaleza algorítmica, es decir, debe proceder mecánicamente y debe producir la decisión efectivamente en un número finito de pasos. Evidentemente, son éstos los procedimientos que pueden ser ejecutados por una máquina, por lo menos en principio.

Teniendo en cuenta la gran importancia que han logrado las grandes máquinas matemáticas de hoy día, es clara la necesidad de una teoría general de algoritmos en el sentido anterior. Naturalmente,

la descripción dada arriba de una algoritmo no es bastante precisa para basar una teoría matemática en ella. Por eso, el primer problema de una teoría de algoritmos será el de precisar el concepto mismo de algoritmo.

En el curso del tiempo los matemáticos han logrado varias maneras de precisar este concepto, y con sorpresa de su parte todas estas definiciones estrictas se han encontrado equivalentes. Es este un caso casi único en la historia de las matemáticas : el hecho de que un concepto aparentemente intuitivo se pueda precisar por un único concepto matemático. El fenómeno mucho más frecuente es el que un concepto intuitivo se desintegra en un gran número de conceptos matemáticos no equivalentes al intentar hacerlo exacto y preciso. Por ejemplo, para los conceptos intuitivos de una curva o de una superficie es bien conocido que existen muchas traducciones matemáticas. Pero para el concepto de algoritmo la situación es muy agradable, porque como ya hemos indicado todas las traducciones matemáticas de este concepto han resultado equivalentes.

Por lo demás, estos conceptos precisos de algoritmos son más antiguos que las grandes máquinas matemáticas; por eso, antes de discutirlos, vamos a considerar algunos ejemplos matemáticos que muestran el gran interés que han tenido los matemáticos desde hace muchos años en una definición exacta del concepto de algoritmo. Estos ejemplos no se confinan a la aritmética, sino que se encuentran en todos los dominios de la matemática. Indicaremos ejemplos de la aritmética, del algebra, de la topología y de la lógica. Para cada una de estas disciplinas daremos 2 ejemplos : uno simple y clásico y un famoso ejemplo difícil. Pero antes, observemos que es natural generalizar un poco los problemas considerados : no buscamos sólo los procedimientos algorítmicos para determinar si un número natural n tiene una propiedad T, sino también algorítmos para determinar si una relación R de v argumentos tiene lugar para cualquier sistema $n_1, \ldots, n_v$ de números naturales arbitrariamente prefijados. He aquí los ejemplos :

ARITMETICA -

1. Determinar si los números naturales n y m son primos entre sí. El algoritmo clásico que resuelve este problema es el bien conocido algoritmo euclídeo para determinar el máximo común divisor de m y n. Recordemos este algoritmo : Sean $m > n$, y ha-

gamos $m = a_0$, y $n = a_1$. Vamos a producir la serie siguiente de divisiones con residuos :

$$a_0 = q_1 a_1 + a_2 \qquad (0 < a_2 < a_1)$$
$$a_1 = q_2 a_2 + a_3 \qquad (0 < a_3 < a_2)$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$
$$a_{\nu-2} = q_{\nu-1} a_{\nu-1} + a_\nu \qquad (0 < a_\nu < a_{\nu-1})$$

Continuando hasta que el residuo sea nulo, lo cual debe suceder después de un número finito de pasos, ya que los residuos siempre decrecen. Sea, entonces :

$$a_{\nu-1} = q_\nu a_\nu$$

Entonces a_ν es el máximo divisor común de a_0 y a_1. Para determinar si m y n son primos entre sí, tenemos todavía que determinar si a_ν es 1 ó no. Evidentemente, el procedimiento entero es un procedimiento algorítmico, y muy fácilmente puede programarse en una máquina.

2. El famoso llamado décimo problema de Hilbert es el siguiente : determinar si una ecuación diofántica prefijada posee una solución en números enteros. Una ecuación diofántica, como es bien conocido, es una ecuación del tipo :

$$(1) \qquad \sum_{i_1,\ldots,i_k=0}^{n} a_{i_1\ldots i_k} x_1^{i_1} \cdots x_\kappa^{i_\kappa} = 0$$

donde los $a_{i_1\ldots i_k}$ son números enteros fijos.

En primer lugar, observamos que este problema de Hilbert no tiene la forma indicada arriba, i. e. ; determinar si un número entero n posee una propiedad T; pero es fácil reducirlo a esta forma. Tal reducción se efectúa mediante una numeración de GÖDEL : en primer lugar prefijar la ecuación (1) no es otra cosa que prefijar el sistema ordenado de los $(m - 1)^k$ coeficientes $a_{i_1\ldots i_k}$; ahora bien, el conjunto de todos estos sistemas posibles de coeficientes puede numerarse de manera efectiva en tal forma que conociendo el sistema, el coeficiente se calcula en un número finito de pasos, así como también su número indicial en esta numeración , y viceversa. Una tal

numeración se logra haciendo corresponder al sistema de coeficientes ($a_{i_1 \ldots i_k}$) el número

$$(2) \qquad n = p_1^{\varepsilon(a_{(1)})} p_2^{|a_{(1)}|} \cdot p_3^{\varepsilon(a_{(2)})} \cdot p_4^{|a_{(2)}|} \cdots$$

donde $p_1 < p_2 < \ldots < p_k < \ldots$ son los números primos consecutivos, $a_{(1)}, a_{(2)}, \ldots$ son los coeficientes $a_{i_1 \ldots i_k}$ ordenados según el orden lexicográfico de todas las k-plas $(i_1, \ldots, i_k)$ y

$$(3) \qquad \varepsilon(a) = \begin{cases} 1 & \text{si} \quad a > 0 \\ 2 & \text{si} \quad a > 0 \\ 0 & \text{si} \quad a = 0 \end{cases}$$

Evidentemente, esta numeración posee el carácter efectivo deseado, porque conociendo el número n se puede unívoca y efectivamente determinar el correspondiente sistema de coeficientes mediante la descomposición de n en factores primos. Naturalmente, existen otras muchas posibilidades para efectuar tal numeración efectiva ; ahora bien, utilizando uno u otro método para enumerar las ecuaciones diofánticas, el problema de Hilbert se transforma en el problema siguiente : hallar un procedimiento algorítmico para determinar si cualquier número natural n posee la propiedad siguiente o no la posee : n es el número de una ecuación diofántica que tiene una solución en números enteros. Indiquemos que hasta hoy no se conoce ni una solución positiva ni una solución negativa al décimo problema de Hilbert.

ALGEBRA -

1. Sea Z un grupo cíclico con un elemento generador a, de orden N. Determinar si para dos números naturales n, m cualesquiera, se tiene $a^n = a^m$. La solución positiva de este problema resulta inmediatamente del siguiente teorema elemental de la teoría de grupos : para cualquier elemento a de un grupo, $a^n = a^m$ si y sólo si n y m son congruentes módulo el orden de a. Así pues, el problema se reduce al determinar si n - m es divisible por N, y para este problema ya existe el bien conocido algoritmo de la aritmética elemental.

2. Una generalización del problema precedente es el famoso problema de palabras (Wortproblem) de la teoría de grupos : hallar un procedimiento algorítmico para determinar si dos palabras cualesquiera en los generadores $a_1, \ldots, a_n$ son iguales, es decir : hallar un procedimiento que nos permita decidir cuando dos productos de potencias de los $a_1, \ldots, a_n$, con relaciónes generadoras.

$$(4) \quad R_1(a_1, \ldots, a_n) = R_2(a_1, \ldots, a_n) = \cdots = R_N(a_1, \ldots, a_n) = e$$

son iguales (e es el elemento unidad del grupo). Efectuando una numeración de GÖDEL sobre el conjunto de todos los sistemas de generadores y relaciones generadoras, este problema se reduce al de determinar si entre dos números naturales prefijados existe una cierta relación. En el problema general de las palabras se exige, como hemos dicho, un algoritmo común para todos los valores de n y para todos los sistemas de relaciones generadoras. En 1953 NOVIKOV demostró que este problema es insoluble. Sin embargo se pueden resolver algunos problemas particulares de palabras para valores particulares de n y m sistemas particulares de relaciones generadoras; por ejemplo, para $n \geq 1$ y la relación $a^N = e$, tenemos el problema precedente del grupo cíclico.

TOPOLOGIA .

1 Un problema clásico en topología es aquel que nos pide decidir el tipo topológico de una superficie cerrada, el cual se enuncia así : dados los esquemas combinatorios de sendas triangulaciones de dos superficies cerradas, determinar si las superficies son homeomorfas, es decir del mismo tipo topológico.

El esquema combinatorio de una triangulación no es otra cosa que la indicación de sus triángulos, aristas y vértices diciendo que aristas inciden con tales triángulos, y qué vértices inciden con tales aristas. Por ejemplo, el esquema de la triangulación usual de la superficie de un tetraedro (que, naturalmente, es del tipo topológico de la esfera S^2) tiene cuatro triángulos, seis aristas y cuatro vértices. incidentes de manera bien conocida. (figura 1).

Otro ejemplo es la triangulación del plano proyectivo que indicamos aquí en en la figura 2:

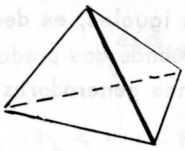

FIGURA 1.

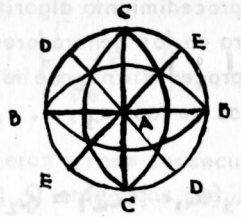

FIGURA 2.

consideremos el círculo del centro A con la subdivisión de el hecha allí. Si identificamos los puntos de la circunferencia diametralmente opuestos, obtenemos como es bien sabido, un plano proyectivo. El problema en los casos de superficies de dimensión dos se resuelve por un teorema clásico de la topología algebraica : el tipo topológico de una superficie cerrada está unívocamente determinado por su característica de EULER y por su carácter de orientabilidad. Estos a su vez están determinados unívocamente por el primer y el segundo grupo de homología y es bien conocido como se calculan éstos de manera algorítmica para un poliedro. Se tiene, por lo tanto, un proceso algorítmico para determinar el tipo topológico de una superficie cerrada de dimensión dos. Ahora bien, como el conjunto de todas las triangulaciones es numerable, y además admite una numeración gódeliana, este problema pertenece al género de los indicados anteriormente, i. e. : determinar si dos números naturales corresponden, en la dicha numeración de GÖDEL, a superficies cerradas del mismo tipo topológico.

2. Evidentemente, el problema anterior puede generalizarse a variedades de dimensión ≥ 2 . Es éste <u>el problema general de homeomorfía</u> : hallar un algoritmo para decidir si dos triangulaciones cualesquiera determinan poliedros del mismo tipo topológico. Es conocido desde los albores de la topología algebráica que el cálculo de los diversos invariantes topológicos que pueden efectuarse mediante una triangulación, a saber : dos grupos de homología, el grupo fundamental, el anillo de cohomología, etc., no nos ofrece una solución completa del problema considerado, porque ellos no constituyen un sistema completo de invariantes. Y podrá pensarse que al encontrar nuevos invariantes efectivos se encuentra una solución al problema. Sin embargo, en el año de 1958 MARKOV demostró que eso no es posible, porque el problema general de homeomorfía es insoluble ; es

decir, no existe un algoritmo del género deseado.

Lógica - 1. Para la construcción de la lógica formal en primer lugar es preciso construir un lenguaje formal, es decir, un lenguaje exacto y con reglas sintácticas y gramaticales precisas, porque los lenguajes cotidianos no son bastante exactos. Existen diversos lenguajes formales. el más sencillo, pero a la vez el menos expresivo de estos lenguajes, es el del cálculo proposicional.

Saliendo de variables proposicionales $P_1, Q_1, \ldots$ se forman por aplicación iterativa de los signos conectivos $\sim$ (no); $\&$ (y); $\vee$ (ó; lat. vel); $\Rightarrow$ (si ... entonces); $\Leftrightarrow$ (si y solo si), expresiones del tipo

$$P_1 \vee Q_1,$$

$$P_1 \Rightarrow (Q_1 \vee Q_2), \text{ etc....}$$

La significación de estas expresiones para los valores de verdad (V) ó falsedad (F) es bien conocida. También es sabido que entre ellas existen algunas universalmente válidas, es decir, que toman siempre el valor de V para toda combinación de valores de las variables $P_1, Q_1, \ldots$ Ejemplos de tales expresiones son las siguientes :

$$(P_1 \vee \sim P_1) \Rightarrow P_1$$

$$P_1 \Rightarrow (P_1 \vee Q_1)$$

$$P_1 \Rightarrow (Q_1 \Rightarrow P_1)$$

Este tipo de expresiones nos plantea el problema de decidir cuándo una expresión del cálculo proposicional es universalmente válida (problema de la decisión). Evidentemente existe un algoritmo para resolver este problema y es fácil describirlo en detalle, por medio de las llamadas tablas de verdad (Igualmente, es claro que es posible, en principio, construir una máquina que haga sistemáticamente el papel de las tablas de verdad). Y en esta forma el problema se reduce por una numeración gódeliana de las expresiones a un problema del tipo anterior; de manera precisa, a determinar si un número natural arbitrario corresponde al número de GÖDEL de una expresión universalmente válida.

2. Una lengua más expresiva que la del cálculo proposi-

cional es la lengua de cálculo funcional de orden uno o, simplemente, cálculo de predicados. Las expresiones de esta lengua se construyen partiendo de variables de predicados de un argumento $P^1\ Q^1 \ldots$ de dos argumentos $P^2, Q^2, \ldots$ etc. y de variables de individuos o sujetos x, y, z, ... En primer lugar se construyen las expresiones atómicas P^1x, Q^2xy . . . que se interpretan de la manera siguiente :

" el individuo x tiene la propiedad P^1 "

" entre x ó y existe la relación Q^2 "

etc. .

Luego se forman las expresiones generales por medio de las expresiones atómicas, los signos conectivos del cálculo proposicional y los cuantificadores $\exists$ (existe un), $\forall$ (para todo). Por ejemplo, el cálculo de predicados contiene las siguientes expresiones :

$$(\forall x)(P^1x \Rightarrow \exists y \mid Q^2yx),$$

$$(\forall x)(\forall y)(Q^2yx \Rightarrow Q^2xy),$$

. . . . . .

Algunas de estas expresiones son también universalmente válidas, es decir : válidas para toda interpretación de las variables. Por ejemplo. es evidente que las expresiones siguientes :

$$(\forall x)(P^1x \Rightarrow \exists x \mid P^1x)$$

$$(\forall x)(\forall y)(P^2xy) \Leftrightarrow (\forall y)(\forall x)(P^2xy)$$

son de este género. Aunque la validez universal de estos dos ejemplos se ve inmediatamente, no debe creerse que la situación sea semejante en los otros casos. En efecto, el problema de la decisión para el cálculo de predicados (es decir) hallar un algorítmo para determinar si una expresión arbitrariamente prefijada del cálculo de predicados es universalmente válida o no), es mucho más difícil que el problema de la decisión en el cálculo proposicional, ya que en el año de 1936 CHURCH demostró que el problema general es insoluble. Este resultado fué el primero de carácter negativo acerca de la existencia de ciertos algoritmos; mientras que algunos resultados positivos (como, por ejemplo la construcción del algorítmo euclídeo), ya

tienen la edad de dos mil años, los resultados negativos, como vemos, no tienen más de treinta años.

Este hecho no es accidental. Si existiese un algoritmo para cualquier problema podríase entonces hallársele y, en particular, podría considerarse como un algoritmo sin que existiese una definición precisa de este concepto, ya que entonces la naturaleza algorítmica del procedimiento hallado sería evidente, por lo menos, en el sentido intuitivo. Pero en cambio, para demostrar resultados negativos, es decir, la inexistencia de un algoritmo para un problema determinado, es evidentemente necesaria una definición precisa de algoritmo porque en caso contrario no sería posible delimitar, con bastante precisión, el dominio de los procedimientos admisibles. Es esta la razón por la cual los resultados negativos aparecieron después de que se había dado una definición precisa de algoritmo, y no antes. Varias definiciones fueron dadas en los años corridos entre 1930 y 1950 por HERBRAND, GöDEL, KLEENE, CHURCH, BEMOYS, TURING, POST, MARKOV. Ya hemos anotado que estas definiciones han resultado equivalentes. Por eso bastará, en primer lugar, considerar sólo una de estas definiciones y elegimos la definición de TURING por su sabor intuitivo. En primer lugar, hemos mencionado muchas veces que por una numeración gödeliana apropiada los problemas considerados en la teoría de algoritmos, pueden reducirse al siguiente : determinar de manera efectiva si una relación R de v argumentos tiene lugar para un sistema $n_1 \ldots, n_v$ de números naturales arbitrariamente dados. Es fácil ver que este problema, en general, es equivalente al de calcular de manera efectiva el valor de una función aritmética f en $(n_1 \ldots, n_v)$. (Aquí una función aritmética de v argumentos es una función f que a cada sistema de v números naturales $n_1 \ldots, n_v$ le hace corresponder un número natural $f(n_1, \ldots, n_v)$.)

En efecto, para convencerse de esta equivalencia, tomemos una función aritmética f de v argumentos, y consideremos la relación R de v argumentos definida por

$$(6) \qquad R(n, n_1, \ldots n_v) \iff n = f(n_1, \ldots, n_v)$$

Es claro que entonces el problema de calcular efectivamente los valores de la función f, se reduce al de determinar si efectivamente

la relación R tiene lugar o nó. En efecto, si se tiene un algorítmo para este último problema, entonces decidiendo si la relación R tiene lugar para $(1, n_1, \ldots, n_v)$, $(2, n_1, \ldots, n_v)$, ... después de un número finito de pasos se encontraría el valor de $f(n_1, \ldots, n_v)$. Recíprocamente, sea R una relación de v argumentos, y consideremos la función aritmética definida por

$$(7) \qquad f(n_1, \ldots, n_v) = \begin{cases} 1 \text{ si } R(n_1, \ldots, n_v) \text{ tiene lugar,} \\ 0 \text{ en caso contrario.} \end{cases}$$

Es claro que entonces decidir si efectivamente R tiene lugar para $n_1, \ldots, n_v$ no es otra cosa que calcular el valor $f(n_1, \ldots, n_v)$.

(Continuará).