SUMA 47
Noviembre 2004, pp.29-36

Tres problemas clásicos y complejidad

En este trabajo se muestra como algunos de los problemas de más interés de las ciencias de la computación pueden encontrarse en la historia de las Matemáticas. En concreto, se comentan los tres problemas clásicos como un ejemplo de búsqueda de la solución de un problema dentro de un modelo de computación, esto es, con una restricción sobre las operaciones que se pueden efectuar. Se comenta someramente la historia de estos problemas. El concepto de complejidad en el peor de los casos se presenta como una manera adecuada de medir la efectividad de un algoritmo que resuelve un problema.

This paper shows how some of the most interesting problems within the computer sciences can be found in the history of Mathematics. In fact, the three classic problems are mentioned as an example of the quest for problem solution within a computer pattern; that is to say, with some restriction over the operations to be carried out. The concept of complexity in the worst of cases is put forward as a suitable way of measuring how effective a problem-solving algorithm is.

Cuando una persona adquiere un nuevo instrumento, digamos por ejemplo un vídeo diferente o un robot doméstico nuevo, su primer interés es conocer todo lo que dicho instrumento es capaz de hacer. En el manual de instrucciones de cada aparato se describen todas las operaciones que éste puede realizar. Así, con el nuevo vídeo puede ver películas, grabar programas de televisión... o con el robot puede preparar tal o cual guiso, exprimir, triturar... Por otro lado el usuario también está interesado en saber con qué tipo de objetos funciona su electrodoméstico. Se pregunta si su vídeo admite cintas del tipo VHS o puede usar un CDROM o un DVD, o si su robot doméstico admite todo tipo de alimentos o debe tener precauciones con algunos de ellos. Este sencillo ejemplo ilustra las dos preguntas generales sobre las que se va a construir el concepto abstracto de modelo computacional.

Pregunta 1: ¿Qué tipo de datos admite el sistema?
Pregunta 2: ¿Qué operaciones es capaz de realizar dicho sistema con estos datos?

Entendiendo el término *datos* en su sentido etimológico como *lo que se da*, esto es, lo dado.

Por tanto, si queremos construir un modelo abstracto que represente las potencialidades de una máquina, un conjunto de máquinas, un ordenador o ciertos instrumentos, se debe basar sobre la respuesta a esas dos preguntas anteriores: datos y operaciones.

Un *modelo computacional* se describe entonces dando:

Los datos de entrada que va admitir.
Las operaciones básicas que va a poder efectuar con dichos datos.

Ejemplo: Un modelo computacional muy habitual en el contexto de las ciencias de la computación es el conocido como *Real RAM* (*Random access machine*) donde los datos de entrada están formados por números reales y las operaciones básicas son:

- Asignación a una variable de una unidad de memoria.
- Operaciones aritméticas: suma, resta, multiplicación y división.
- Comparaciones: dados dos números reales *a* y *b* determina si *a* es menor, igual o mayor que *b.*

Observación: No vamos a entrar en estas notas en dos interesantes cuestiones que aparecen relacionadas con este ejemplo pero que tienen carácter general. La primera acerca de cómo representar los números reales (en general cómo representar

Roberto Muñoz
ESCET, Universidad Rey Juan Carlos. Madrid.

adecuadamente los datos de entrada de un modelo computacional, sean números, letras, palabras, o datos de alguna otra naturaleza). La segunda sobre los tipos de datos que se pueden construir (y qué características y ventajas tienen) con los números reales (o en general con los elementos que se hayan elegido como datos de entrada en el modelo computacional) digamos por ejemplo, listas, árboles, arrays...

Una vez que tenemos este ente abstracto que es el modelo computacional, y dado un problema *P*, nos gustaría saber si podemos encontrar una combinación de las operaciones básicas del modelo que resuelva el problema *P*, esto es, un algoritmo que resuelve *P*. Para ser precisos, demos una definición del término algoritmo.

Definición: Dado un modelo computacional y un problema *P*, definimos un algoritmo que resuelve *P* (en dicho modelo) como una lista ordenada y finita de operaciones básicas del modelo que, aplicadas a unos datos de entrada precisamente definidos, y tras realizar un número finito de operaciones, construyen una solución del problema *P*.

Por ejemplo en el modelo computacional *Real RAM* podemos resolver ecuaciones del tipo siguiente: $Ax + B = C$, donde A, B, C son números reales y, en particular, A es no nulo. En efecto, la solución es el cociente $(C-B)/A$ y se ha construido mediante una diferencia y un cociente, que son operaciones básicas para el modelo considerado *Real RAM*.

Entonces podemos hablar de *problemas resolubles* (respectivamente *irresolubles*) en un modelo computacional, como aquellos para los que se puede construir (respectivamente no se puede construir) un algoritmo que los resuelva.

Desde el siglo V AC, distintos matemáticos han trabajado tratando de resolver algunos problemas como la trisección de un ángulo, la duplicación de un cubo y la cuadratura de un círculo mediante la geometría con regla y compás.

Tomemos ahora un problema *P* resoluble en un modelo computacional. Supongamos que tenemos una máquina que efectúa las operaciones básicas de nuestro modelo computacional y que necesita, por ejemplo, una décima de segundo para efectuar cada una de dichas operaciones básicas. Si el algoritmo que resuelve *P* necesita realizar demasiadas operaciones, por ejemplo, 3.1536 por la décima potencia de 10, entonces tendremos que esperar 100 largos años para que el algoritmo nos ofrezca una solución del problema en cuestión, lo que parece un tiempo demasiado largo. En este sentido podremos hablar de *problemas resolubles efectivamente* en un modelo computacional como aquellos problemas para los que puede construir un algoritmo que realice una cantidad de operaciones que precisen de un tiempo *razonable* para su solución. El término razonable es, en este contexto, un término relativo que no será igual para una empresa con balance anual, que para un proyecto quinquenal, que en otros contextos.

Por tanto, resumiendo, dado un modelo computacional nos estamos preguntando:

Sobre la resolubilidad. ¿Qué problemas son resolubles dentro del modelo, es decir, para qué problemas se puede construir un algoritmo que aporte una solución del problema.

Sobre la resolubilidad efectiva ¿Qué problemas, dentro de los que son resolubles en el modelo, se pueden resolver efectivamente, esto es, el algoritmo que los resuelve precisa de un tiempo razonable (en un sentido que precisaremos más adelante) para construir la solución del problema?

Pongamos un ejemplo del alcance de estas preguntas. La seguridad informática está en muchos casos basada sobre métodos criptográficos que se fundamentan en que el problema de la factorización de números naturales es, por ahora, un problema no efectivamente resoluble. Esto quiere decir que los algoritmos que se conocen necesitan un tiempo demasiado largo para decodificar mensajes, por lo que el sistema es seguro. Pero el hecho de que no se conozcan algoritmos efectivos no significa (salvo que haya una demostración que lo indique y no es el caso) que no puedan existir.

Los ejemplos que vienen a continuación ilustran situaciones en las que ciertos problemas son no resolubles en un modelo computacional. La infructuosa búsqueda de soluciones durante siglos parecía indicar que esto era así. Pero conviene señalar que el mero paso del tiempo no es un argumento en sí mismo. Por ejemplo, el problema de la construcción con regla y compás de un polígono regular de 17 lados pareció durante cientos de años (de la Grecia clásica a los tiempos de Gauss) irresoluble, hasta que Gauss realizó dicha construcción, de la que se sintió tan orgulloso que la mandó tallar sobre su tumba.

Construcciones con regla y compás

Situados tras la introducción en los conceptos básicos, podemos pensar que las cuestiones planteadas han aparecido con la tecnología, que antes de los ordenadores este tipo de preguntas sobre la resolubilidad de un problema carecían de inte-

rés. La historia de las matemáticas muestra que esta curiosidad está presente en el trabajo de diferentes personas, aun antes de que fuera un problema la evaluación de las potencialidades de un instrumento. Como ejemplo podemos mencionar que desde el siglo V antes de Cristo hasta el siglo XIX distintos matemáticos han trabajado duramente para tratar de resolver algunos problemas, a saber: *la trisección de un ángulo, la duplicación de un cubo y la cuadratura de un círculo* dentro del modelo computacional conocido como *la geometría con regla y compás.*

Definamos el modelo computacional: *Geometría con regla y compás.*

Los *datos de entrada* están formados por tres tipos de objetos:

- Los puntos.
- Las circunferencias.
- Las rectas.

Las *operaciones básicas* que vamos a poder realizar son las siguientes:

- Apoyar una pata del compás en un punto.
- Producir una circunferencia.
- Apoyar la regla en un punto.
- Producir una recta.
- Intersecar rectas, circunferencias y rectas con circunferencias.

Y nos estamos rigiendo por los axiomas habituales de la geometría euclídea, que son, citando textualmente (Boyer, 1987):

Postulados

- Trazar una recta desde un punto a otro cualquiera.
- Prolongar una línea recta finita de manera continua a otra línea recta.
- Describir un círculo con cualquier centro y cualquier radio.
- Que todos los ángulos rectos son iguales.
- Que si una línea recta corta a otras dos líneas rectas formando con ellas ángulos interiores del mismo lado menores que dos ángulos rectos, las dos líneas rectas, prolongadas indefinidamente, se cortan del lado por el cual los ángulos son menores que dos ángulos rectos.

Nociones comunes

- Cosas que son iguales a la misma cosa son iguales entre sí.
- Si iguales se suman a iguales, los resultados son iguales.
- Si iguales se restan de iguales, los restos son iguales.
- Cosas que coinciden una con otra son iguales entre sí.
- El todo es mayor que la parte.

Pongamos dos ejemplos de problemas resolubles en este modelo computacional:

Ejemplo: Construcción de ángulos rectos. Para construir un ángulo recto se traza un segmento *AB*. Se pincha el compás en *A* y se apoya la otra pata en *B*. Se traza una circunferencia *C*1 con el compás así situado. Se repite el proceso pinchando el compás en *B* y apoyando la otra pata en *A*, construyendo la circunferencia *C*2. Se traza el segmento *PQ* que une los dos puntos de intersección de *C*1 y *C*2. Este segmento es perpendicular al segmento *AB* y además divide el segmento *AB* en dos partes iguales, es decir, si *M* es el punto de intersección de los segmentos *PQ* y *AB* se tiene que la longitud de *AM* es igual a la longitud de *MB*.

En la Figura 1 el segmento *AB* es el segmento horizontal. De este modo haciendo la construcción que se detalla en el párrafo anterior, si llamamos *P* al punto de la intersección de *C*1 y *C*2 que está en el semiplano superior definido por la recta *AB*, se forman dos triángulos equiláteros: *APB* y *BQA* (pues su lado es el radio común de *C*1 y *C*2, esto es, la longitud del segmento *AB*). Al ser triángulos equiláteros se tiene que *AP* y *BQ* (respectivamente *PB* y *AQ*) son rectas paralelas. Esto permite concluir que los triángulos *APM* y *MBQ* son iguales y por tanto que la longitud del segmento *AM* es igual a la de *MB*. Esto justifica el hecho de que *AMP* es un ángulo recto, pues finalmente se tiene la igualdad de los cuatro triángulos *AMP=BMP=AMQ=BMQ.*

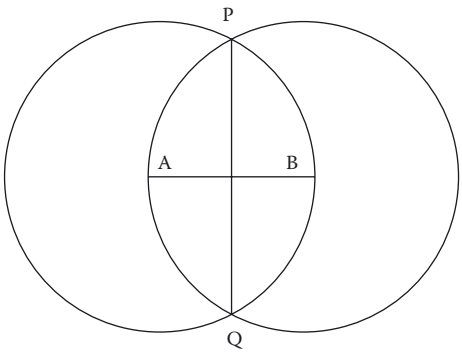

Figura 1. Construcción de ángulos rectos

Ejemplo: Construcción de hexágonos regulares. Para construir un hexágono regular se tiene el siguiente algoritmo:

- Dibujar un círculo *C* de radio *r*.

- Elegir un punto *p1* en *C* y pintar un círculo *C1* con centro en *p1* y radio *r*. Se tiene que la intersección de *C* y *C1* es el conjunto *{p2, p3}*.

- Dibujar un círculo *C2* con centro *p2* y radio *r*. Entonces la intersección de *C2* y *C* es *{p1, p4}*.

• Dibujar un círculo *C3* con centro *p4* y radio *r*. Entonces la intersección de *C3* y *C* es *{p2, p5}*.

• Dibujar un círculo *C4* con centro *p5* y radio *r*. Entonces la intersección de *C4* y *C* es *{p6, p4}*.

El hexágono es, dando sus vértices: *p1*, *p2*, *p4*, *p5, p6, p3, p1.*

El lector puede reproducir esta construcción y verificar por sí mismo que, en efecto, se ha formado un hexágono regular.

Este modelo computacional de la geometría con regla y compás es conocido desde la Grecia clásica donde comenzaron a preguntarse acerca de qué construcciones eran posibles dentro de él. En la colección de libros *Los Elementos* de Euclides (un compendio del conocimiento de la geometría en la Grecia clásica) se pueden encontrar múltiples construcciones con regla y compás para resolver problemas de índole geométrica (e incluso aritmética). Entre los problemas para los que no se obtuvo una construcción con regla y compás se plantearon los siguientes, que por su interés para la comunidad matemática griega y para la de los siglos venideros hasta la actualidad, se han venido a denominar *Problemas clásicos*.

1. Problema de la duplicación de un cubo: Dado un cubo *C* de arista *a,* dar una manera de construir (con regla y compás) un cubo *C'* de volumen el doble. Esto es, el volumen de *C* es el cubo de *a* y el volumen de *C'* debe ser el doble del cubo de *a*.

2. Problema de la cuadratura de un círculo: Dado un círculo *T* de radio *r,* construir con regla y compás un cuadrado *T'* de lado *l* de modo que el área de *T* y el área de *T'* sean iguales.

3. Problema de la trisección de un ángulo: Dado un ángulo, digamos *a*, construir con regla y compás el ángulo *a/3*.

Después de dar algunas soluciones a estos problemas en modelos computacionales que no son la geometría con regla y compás (esto es, esencialmente, permitiendo alguna operación más), muchos siglos después (en el siglo XIX) se demostró que no son resolubles en ese modelo computacional, en algunos casos mediante el uso de teorías que, en principio, no presentaban relación con el problema inicial. Recorramos someramente la historia de estos problemas.

El problema de la duplicación del cubo

El enunciado del problema es el siguiente, expresado en los términos que hemos ido definiendo en párrafos anteriores:

Problema: ¿Es el problema de la duplicación de un cubo un problema resoluble en el modelo computacional *Geometría con regla y compás*?

Hay dos explicaciones sobre la génesis del problema, que, en cualquier caso, nos permiten datarlo en el siglo V AC. Por un lado una de las afirmaciones sostiene que el oráculo de los dioses informó a los Delianos de que para librarse de una plaga deberían construir un altar de volumen el doble del altar cúbico existente. Por otro, un episodio mitológico en el que el poeta describe que Minos encuentra la tumba de Glauco (cúbica de arista 100 pies) demasiado pequeña para su categoría y para duplicar su volumen propone simplemente duplicar su arista. Claramente, lo que propone el poeta no es una solución del problema porque si la arista *a* se duplica entonces el nuevo volumen es ocho veces el volumen inicial.

Como señalamos antes estas dos fuentes (véase la página *web* [SA]) permiten datar el problema (porque la plaga más importante que asoló Atenas fue alrededor del 430 AC) y nos hace comprender que este problema de duplicar un sólido igual a sí mismo estaba en el interés matemático de la época.

El problema en dos dimensiones, esto es, duplicar el cuadrado, es una construcción muy simple, consecuencia del teorema de Pitágoras. Si el cuadrado *C* tiene arista *a*, entonces la diagonal mide exactamente el producto de *a* por la raíz cuadrada de 2. Por tanto un cuadrado *C'* con esa diagonal como arista tiene área exactamente el doble de la de *C*.

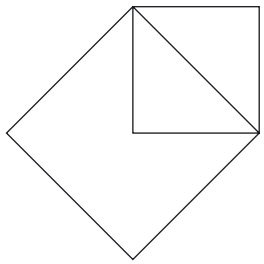

Figura 2. Duplicación de un cuadrado

Volviendo al problema del cubo, la primera idea interesante, posiblemente debida a Hipócrates, acerca del problema es la equivalencia con el siguiente problema:

Problema equivalente: Dados dos números *a* y *b* encontrar dos medias proporcionales entre ellos, esto es, dos números *x* e *y* de modo que *a/x = x/y = y/b*. Así, tomando *b = 2a*, se tiene que *x* es justamente la arista del cubo de volumen el doble.

Se han ido construyendo dichas medias *x* e *y* de distintas maneras a lo largo de la historia, presentamos por ejemplo cómo lo hizo Menecmo, con la ayuda de secciones cónicas. Nuestras medias buscadas forman un punto *p* del plano, digamos *p = (x, y)*. De la igualdad *a/x = x/y* se obtiene que *p* yace en la parábola *P* definida por la igualdad entre el cuadrado de *x* y el producto *ay*. Por otro lado de la igualdad *a/x = y/b* se obtiene que *p* yace en la hipérbola *H* definida por la ecuación

$xy = ab$. De este modo p es el único punto de intersección de P y H que yace en el primer cuadrante.

Esta solución del problema no pertenece a nuestro modelo computacional pues necesita de la construcción de dos secciones cónicas, en concreto, una parábola y una hipérbola, objetos geométricos que no se pueden construir con regla y compás. A lo largo de la historia otros matemáticos fueron dando soluciones al problema pero siempre usando más instrumentos que meramente la regla y el compás.

Hasta el siglo XIX, concretamente en el año 1837, no se demostró que el problema de la duplicación de un cubo *no era resoluble en el modelo computacional de la geometría con regla y compás*. Fue el matemático francés P. Wantzel (ver [Wantzel, 1837]) quien publicó en el *Journal of Liouville* una demostración de que no se podía duplicar el cubo con regla y compás. En los detalles de la demostración no entraremos en estas notas.

Hasta el año 1837, no se demostró que el problema de la duplicación de un cubo no era resoluble en el modelo computacional de la geometría con regla y compás.

El problema de la cuadratura de un círculo

Problema: ¿Es el problema de la cuadratura de un círculo un problema resoluble en el modelo computacional *Geometría con regla y compás*?

Este es un problema cuyo origen también puede situarse en la Grecia clásica y que ha interesado a muchos matemáticos a lo largo de la historia. De hecho, la pregunta central gira en torno al número π, ese número real no racional que describe la proporción entre el radio y la longitud de la circunferencia, de carácter atractivo y misterioso.

Si anteriormente, en el problema de la duplicación del cubo, y después, en el problema de la trisección de un ángulo, hemos aportado unas construcciones geométricas que resuelven el problema (aunque no dentro de nuestro modelo computacional), en este problema vamos a ver cómo la historia ha ido traduciendo el problema de la cuadratura del círculo en un problema algebraico. Y también cómo la solución a este problema algebraico equivalente resolvió el problema original, dando además una mirada más general sobre muchos otros problemas relacionados.

Evidentemente se trata de saber si π es una longitud que se puede construir con regla y compás pues ésta debe ser la arista de nuestro cuadrado (fijando por ejemplo el radio unidad). Si miramos cuidadosamente las ecuaciones de los objetos que podemos construir con regla y compás se trata de: ecuaciones lineales, del tipo $Ax + By = C$, y ecuaciones cuadráticas, del tipo de la ecuación de la circunferencia. Como no podemos medir, (nuestra regla no tiene marcas) debemos suponer que los puntos que podemos escoger de partida tienen coordenadas enteras, de este modo las ecuaciones tienen coeficientes racionales. Un estudio sistemático de las posibles soluciones de estas ecuaciones nos llevan a la siguiente conclusión fundamental.

Teorema: Si un número real c puede construirse con regla y compás entonces debe ser solución de una ecuación polinomial de grado una potencia de 2 con coeficientes racionales.

Este teorema entonces transforma la pregunta sobre la constructibilidad de π en una pregunta sobre si π es *algebraico*, es decir, raíz de un polinomio con coeficientes racionales. Obsérvese que por ejemplo la raíz cuadrada de 2 no es racional y sí es algebraico pues es raíz de un polinomio mónico de grado 2 y además es constructible (como no podía ser de otra manera según el teorema) pues es la diagonal del cuadrado unidad.

Cuando un número real no es algebraico se dice que ese número es *trascendente*.

Fue Lindemann en 1880 quien probó que π es trascendente, esto es, que no es solución de ninguna ecuación polinomial con coeficientes racionales, demostrando con ello que la cuadratura del círculo es un problema no resoluble en el modelo computacional *Geometría con regla y compás.*

El problema de la trisección de un ángulo

Problema: ¿Es el problema de la trisección de un ángulo un problema resoluble en el modelo computacional *Geometría con regla y compás*?

En 1837, Wantzel, en el mismo artículo señalado anteriormente, demostró también que el problema de la trisección del ángulo no se puede resolver con regla y compás. Veamos, sin embargo, algunas construcciones geométricas (no con regla y compás) que permiten resolver esta cuestión.

Comencemos con un problema más simple: construir la *bisección de un ángulo*, esto es, dividir un ángulo escogido en dos partes iguales. Esta construcción se puede efectuar con regla y compás. Para esto, tomando el ángulo BAC, basta completar el paralelogramo $ABDC$ y trazar la diagonal.

Veamos una manera de trisecar un ángulo, que ya conocía Hipócrates, y que no es una construcción con regla y compás. Es lo que vamos a llamar una solución *mecánica*; después de describirla, veremos qué significa esto.

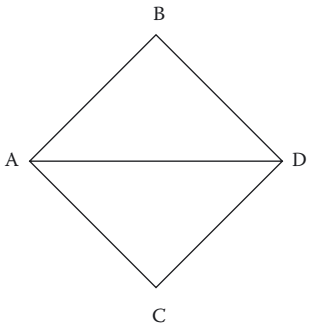

Figura 3. Bisección de un ángulo

Partimos del ángulo *CAB* y trazamos la perpendicular a *AB* pasando por *C*. Esta perpendicular corta al segmento *AB* en un punto que denominamos *D*. Construimos ahora el punto *F* que completa el rectángulo *ADCF*. Elegimos un punto *E* en la recta *FC* de modo que la intersección *H* de las rectas *AE* y *DC* verifique que la longitud de *HE* sea el doble de la longitud *AC*. Se tiene que el ángulo *EAB* es la tercera parte del ángulo *CAB*. En efecto, situamos el punto *G* en la mitad del segmento *HE*. Como la longitud de *HE* es el doble de la longitud de *AC* (así se ha construido *E*) entonces *HG = GE = AC*. Ahora bien, el ángulo *ECH* es un ángulo recto y por tanto *CG = HG = GE* (para ver esto basta situar los catetos del triángulo *CEH* sobre los ejes coordenados y razonar analíticamente). El ángulo *EAB* es igual al ángulo *CEA* (puesto que las rectas *AB* y *EF* son paralelas) e igual al ángulo *ECG* (ya que *ECG* es un triángulo isósceles). También el ángulo *CAG* es igual al *CGA* debido a que *AC = CG*. Finalmente el ángulo *CGA* es igual a la suma de los ángulos *GEC* y *ECG*, por lo que el ángulo *CGA* es dos veces el ángulo *EAB,* como queríamos demostrar.

Fue Lindemann en 1880 quien probó que π es trascendente, esto es, que no es solución de ninguna ecuación polinomial con coeficientes racionales.

La solución que hemos presentado, como decíamos en párrafos anteriores, no es una solución con regla y compás. El problema radica en la construcción de *E* con la condición de que *HE = 2AC*. Esta construcción se puede realizar de forma *mecánica*. Simplemente se marca una distancia *2AC* en la regla y se apoya un extremo de la marca sobre la recta *CD* y el otro sobre *FC* hasta hacerla pasar por *A*.

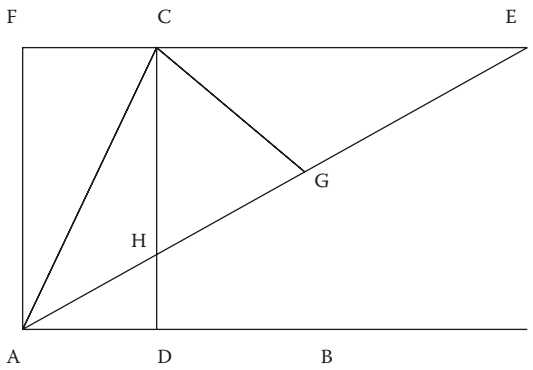

Figura 4. Trisección mecánica de un ángulo

Complejidad

Si las preguntas acerca de la resolubilidad de problemas dentro de un modelo computacional son cuestiones presentes en la historia de la matemática, la aparición de los ordenadores ha modificado parcialmente estas preguntas. Este cambio que sugerimos proviene del hecho de tener una máquina que permite hacer con total exactitud y bastante rapidez una cantidad ingente de operaciones, para las que un hombre necesitaría demasiado tiempo y además la probabilidad de que las hiciera mal sería muy elevada. Existen en la actualidad algoritmos que resuelven problemas que no serían considerados como tales por un matemático, digamos, del siglo XIX, puesto que no existía el instrumento (ni la mente) que los llevara a la práctica. Luego una pregunta muy interesante en la actualidad es acerca de los problemas *efectivamente resolubles*, esto es, aquellos problemas resolubles en nuestro modelo computacional para los que el algoritmo que los resuelve precisa de un tiempo *razonable* para obtener su solución.

En esta sección definiremos la *complejidad de un algoritmo* como una manera de medir el tiempo que necesita un algoritmo para resolver un problema. La complejidad será entonces una manera de medir la bondad temporal de un algoritmo. Este concepto nos permitirá también definir la *complejidad de un problema* como el orden de complejidad del mejor algoritmo que lo resuelve.

Definición: Sea un algoritmo *A* en un modelo computacional. Se define la función *T(n) número de operaciones en el peor de los casos* como el máximo del número de operaciones que realiza el algoritmo *A* para todas las entradas que tienen tamaño *n*.

De esta manera hay que definir precisamente lo que entendemos por tamaño de la entrada. Si, por ejemplo, estamos en el modelo computacional *Real RAM* y el algoritmo consiste en encontrar un número *a* en una lista de números podemos definir el tamaño de la entrada como la longitud *n* de la lista en cuestión.

Tomemos como ejemplo de algoritmo de búsqueda el algoritmo denominado de *búsqueda secuencial* que consiste simplemente en recorrer la lista dada y comparar si el término *a* es igual a algún elemento de la lista. Si lo encuentra el algoritmo se detiene y la salida es *Sí.* Si recorre la lista sin encontrarlo la salida es *No.* Podemos escribirlo en pseudocódigo de la siguiente manera:

Entrada: a1, ..., an; a
i:=1
while *i<n+1*
if a=ai then i:=n+2 else i:=i+1
if i=n+2 then r:=Sí else r:=No
Salida: *r*

El tamaño de la entrada es *n* y el peor de los casos es aquél en que *a* no está en la lista, porque ésta ha de recorrerse entera. Podemos computar entonces *T(n).*

- Se comienza con una asignación para el contador *i.*
- Entramos en el bucle que se repite *n* veces. Dentro del bucle se hacen:
- una comparación de entrada (*i* se compara con *n*+1) que involucra también una suma (*n*+1),
- una comparación de *a* con *ai* y una asignación al contador *i* de *i*+1 (por tanto también una suma),
- una comparación que nos saca del bucle (*i*=*n*+1 con *n*+1).
- Finalmente se realiza una comparación de la *i* de salida y una asignación a *r*.

Por tanto $T(n) = 1 + n(2 + 2) + 2 + 3 = 4n + 6$.

Podemos hacer dos precisiones sobre la función *T(n)*. En primer lugar resulta en ocasiones ser una función difícil de calcular exactamente. En el ejemplo que hemos presentado el cálculo es simple pero puede haber situaciones en las que no sea sencillo decir con exactitud el número de operaciones (por ejemplo en el algoritmo conocido como *búsqueda binaria* [Rosen, 1999]). Por otro lado no es necesaria una total precisión: si la operación básica necesita una unidad de tiempo muy pequeña (pongamos una milésima de segundo) para ser ejecutada entonces un error por ejemplo de una unidad no es significativo.

Teniendo en cuenta estas precisiones necesitamos un instrumento que permita describir la naturaleza de *T(n)*, en concreto, en su comportamiento asintótico, esto es, para valores muy grandes de la *n*. Este concepto lo tomamos del Análisis Matemático.

Definición: Sean *T* y *S* dos funciones de los números naturales en los reales positivos; se dice que *T domina a S* o que *S(n) pertenece a O(T(n))* si existen dos números reales *n0* y $k > 0$ de modo que para cada $n > n0$ se verifique la desigualdad (no necesariamente estricta) $S(n) < kT(n)$. Si *S* domina a *T* y *T* domina a *S*, diremos que *S* y *T* son *del mismo orden de complejidad*.

Definición: Se llama *complejidad de un algoritmo* al orden de complejidad de su función *T(n),* número de operaciones en el peor de los casos.

La complejidad de un algoritmo es entonces una medida del número de operaciones que éste realiza en el peor de los casos para tamaños muy grandes de la entrada *n* y establece una jerarquía que permite determinar si un algoritmo es mejor que otro (siempre en este sentido). Si el algoritmo *A1* tiene como función número de operaciones en el peor de los casos a *T1* y respectivamente el algoritmo *A2* tiene como función a *T2* y *T2* domina a *T1* entonces (según esta forma de comparar) el algoritmo *A1* es mejor (o al menos igual) que el algoritmo *A2*.

Por ejemplo, el algoritmo de búsqueda secuencial es un algoritmo de complejidad $O(T(n)) = O(4n+6)$. Se puede demostrar que todos los polinomios de grado uno son del mismo orden de complejidad y por tanto $O(T(n)) = O(n)$. Esto es lo que se suele conocer como un algoritmo de *complejidad lineal*.

Así podemos hablar de complejidad lineal (si *T(n)* es un polinomio de grado 1) o de *complejidad cuadrática* (si *T(n)* es un polinomio de grado 2), o de *complejidad cúbica*... o en general de *complejidad polinomial* si *T(n)* es un polinomio. Todas estas definiciones están sustentadas en el siguiente lema, que dejamos como ejercicio al lector.

> *Wantzel, en 1837, demostró que el problema de la trisección del ángulo no se puede resolver con regla y compás.*

Lema: Si *T(n)* es un polinomio de grado *d* entonces es del orden de complejidad del polinomio potencia *d*-ésima de *n.*

Y una muestra de la jerarquía que establece este concepto es la siguiente cadena de dominancias: la exponencial domina a cualquier polinomio, un polinomio de grado mayor domina a otro de grado menor, el logaritmo es dominado por cualquier polinomio de grado positivo. Basta representar las gráficas de las funciones señaladas para comprobar la veracidad de las dominancias.

Por tanto un algoritmo de complejidad lineal es mejor que uno de complejidad cuadrática, la complejidad logarítmica es

mejor que la complejidad polinomial y la complejidad polinomial es mejor que la exponencial.

Un límite teórico para los problemas efectivamente resolubles es la complejidad polinomial. El crecimiento exponencial es demasiado rápido y rápidamente toma valores excesivamente grandes para ser considerado efectivo. Por ejemplo, si *T(n)* es la potencia enésima de 2 y la operación básica se realiza en una décima de segundo se tiene que si *n = 42* entonces en el peor de los casos se realizan 2 elevado a la 42 operaciones básicas, que son muchos años esperando la solución. Entradas de tamaño no demasiado grande necesitan de un tiempo exagerado para ser manipuladas.

El matemático S. Smale viene observando que el estudio de la búsqueda efectiva de soluciones de ecuaciones (mediante algoritmos de tiempo polinomial, si es posible) va a ser el problema central de las matemáticas del siglo XXI, a diferencia del siglo XX en el cual el problema central ha sido el estudio de las propiedades de dichas soluciones, sin buscarlas explícitamente. En la nota del autor de este trabajo (Muñoz) y en las propias reflexiones de Smale (Smale, 1998) se puede ampliar la información sobre este tema.

El matemático S. Smale viene observando que el estudio de la búsqueda efectiva de soluciones de ecuaciones va a ser el problema central de las matemáticas del siglo XXI.

Conclusión

La teoría de los modelos computacionales, las definiciones precisas de algoritmo, problema resoluble, complejidad, problema efectivamente resoluble y complejidad de un problema y el estudio de todos estos conceptos son cuestiones fundamentales para la ciencia de la computación y, citando al propio Smale, un regalo de dicha ciencia a las matemáticas.

La historia de las matemáticas no ha sido ajena a este tipo de cuestiones: ya en la Grecia clásica se preocuparon de si ciertas construcciones se podían efectuar con regla y compás, que no es más que el problema de la resolubilidad de dichos problemas en un modelo computacional concreto, la *Geometría con regla y compás*.

La historia de los problemas de la duplicación del cubo, la trisección del ángulo y la cuadratura del círculo nos enseñan que este tipo de cuestiones han desarrollado importantes teorías matemáticas y son fascinantes, tanto desde el punto de vista matemático, como ahora desde el tecnológico. La historia de los problemas de construcciones de polígonos regulares nos hace ser prudentes, en el sentido de que el hecho de que ahora no sepamos como resolver un problema en un cierto modelo computacional (y carentes de una demostración que pruebe que no es posible) no significa que con el tiempo no seamos capaces. Como fue Gauss, muchos siglos después de los primeros intentos griegos, capaz de construir el polígono de 17 lados con regla y compás (Boyer, 1987).

Las matemáticas actuales y la ciencia de la computación han de trabajar conjuntamente en el desarrollo de teorías cada vez más potentes y en la construcción de algoritmos nuevos o de modelos de computación distintos que resuelvan las limitaciones actuales.

La arquitectura de ordenadores, la tecnología de la programación y las estructuras de datos deben tener un papel importante en la concreción de los entes abstractos que son los modelos computacionales. La computación cuántica, los lenguajes de programación más potentes y la representación más oportuna de la información son novedades que pudieran ser fundamentales para que los modelos computacionales y su capacidad para resolver problemas se conviertan en una tecnología poderosa y a la vez disponible para el ciudadano. ■

REFERENCIAS BIBLIOGRÁFICAS

BOYER, C.B. (1987): *Historia de las Matemáticas*, Alianza Universidad.

MUÑOZ, R.: *Matemáticas para el nuevo siglo*. http://www.escet.urjc.es/~rmunoz

ROSEN, K. H. (1995, 1999): *Discrete Mathematics and its applications*, McGraw-Hil.

Smale, S. (1998): *Mathematical problems for the next century*, Math. Intelligenzer 20, 7-15.

WANTZEL, P. (1837): *J. of Liouville Volume II*, pp. 366-372.

http://www-groups.dcs.sst-andrews.ac.uk/~history/HistTopics