Informática en Topología

por

Julio Rubio, Universidad de La Rioja

Dedicado a José Ignacio: la paz se demuestra actuando

Introducción

En primer lugar, precisar que no voy a tratar de la Informática en Topología en toda su generalidad, sino de la Computación (entendida como el desarrollo de programas de computador) en *Topología Algebraica*. En particular no voy a hablar de las aplicaciones de la Informática a la Topología General (como la enumeración de topologías sobre conjuntos finitos, por ejemplo), ni tampoco de la aplicación de la Topología a la Informática, que incluiría desde cuestiones más o menos banales o terminológicas (como cuando se habla, con propiedad, de la *topología de una red* de computadores) hasta el uso de conceptos y técnicas topológicas en Informática Gráfica, Procesamiento de Imágenes Digitales o en Informática Teórica (a través, en este último caso, de la teoría de dominios para la semántica denotacional de lenguajes de programación, debida a Dana Scott [Sc]). Tampoco voy a entrar en las sorprendentes (para mí) aplicaciones de la Topología Algebraica *sofisticada* en el modelado de sistemas informáticos concurrentes [G]. Sin embargo, hacia el final veremos cómo las técnicas necesarias para calcular en Topología Algebraica *avanzada* tienen repercusión en aspectos teóricos de la Informática (programación funcional y especificación algebraica).

El plan del texto es el siguiente. Comenzamos dando algunas ideas intuitivas sobre la Topología para motivar la aparición de la Topología Algebraica y, a continuación, introducimos el concepto clave de *invariante*. Dedicamos la parte central del trabajo a presentar, de modo elemental, tres invariantes concretos: el número de componentes conexas, el grupo fundamental y los grupos de homología. Para cada uno de ellos intentamos dar una idea intuitiva de sus aspectos geométricos y, simultáneamente, una descripción informal de un método de cálculo. En el apartado dedicado al grupo fundamental haremos un par de digresiones: una dedicada a la presentación de grupos por generadores y relaciones, y otra que versará sobre el concepto de *indecidibilidad*. El artículo termina con una breve sección de conclusiones y temas de investigación.

1. El problema de la igualdad: un mundo de plastilina

Para hacer una primera aproximación a la Topología, conviene considerarla como un tipo de *geometría*. Si interpretamos dicha afirmación a la manera formal defendida por Klein (es decir, una geometría es la acción de un grupo sobre un conjunto dado, acción que define cuándo dos elementos son "iguales", indistinguibles desde esa perspectiva geométrica específica), veremos que no es totalmente cierta, pues para aplicarla cabalmente al caso topológico habría que generalizarla para ser aplicada sobre *clases* generales y no sobre conjuntos (en concreto, sobre la clase de los espacios topológicos) y deberíamos pasar de grupos a *grupoides* (en concreto, el grupoide de homeomorfismos). Pero si dejamos de lado el formalismo, sigue siendo cierto que un problema topológico es esencialmente un *problema sobre la igualdad*. Es decir, como sucede en muchos campos de las matemáticas, el problema esencial en Topología consiste en saber cuándo dos espacios pueden ser considerados "el mismo", desde el particular punto de vista de dicha disciplina. La definición formal de dicha igualdad viene dada por la noción de *homeomorfismo* (función biyectiva y continua, cuya inversa es también una aplicación continua). Pero una visión muy física, y más próxima a los intereses de esta exposición, de dicho problema de la igualdad, consiste en imaginar que los espacios objeto de comparación están constituidos por un material fácilmente maleable (plastilina, se suele decir), y considerar que dos espacios son iguales topológicamente si uno puede ser deformado continuamente en el otro, sin rasgar, agujerear o pegar la materia de la que se suponen compuestos. Esta visión geométrica de la Topología puede parecer muy alejada de la presentación abstracta, axiomática, que suele hacerse del concepto de espacio topológico (aunque ambas perspectivas son totalmente compatibles, por supuesto), pero es el punto de vista conveniente para este trabajo y, en la modesta opinión del

autor, también para una primera aproximación a este campo de las matemáticas.

Para ilustrar cómo se comporta ese "mundo de plastilina" podemos imaginar el siguiente rompecabezas. Se toma en primer lugar un conjunto de puntos en el plano, sea dibujados en la pantalla de un ordenador, sea elegidos al azar. Vamos a suponer que en esa "nube" hay al menos tres puntos y que todos ellos están en *posición general*, lo que significa, en palabras llanas, que no hay tres de dichos puntos que estén alineados. Obsérvese que si los puntos son elegidos al azar, la probabilidad de que no estén en posición general es muy pequeña (nula si suponemos que el plano es un continuo y no una malla discreta; esta segunda opción sería más realista si estamos identificando el plano con una pantalla de ordenador). Si tuviesemos la "mala suerte" de dibujar tres puntos alineados podríamos elegir entre "mover un poquito" (en un sentido que es fácil de formalizar) uno de los puntos conflictivos o, simplemente, eliminarlo (pero teniendo entonces cuidado de no quedarnos con menos de tres puntos). En cualquier caso, el siguiente paso en nuestro rompecabezas consiste en calcular la *envolvente convexa* (*convex hull*, en inglés) de dicha nube de puntos. Intuitivamente, se trata de trazar una poligonal cerrada y convexa que tenga sus vértices elegidos entre los puntos de la nube y con la propiedad de que el resto de puntos queden en el interior de la región del plano acotada por dicha poligonal. La existencia y unicidad de la envolvente convexa es muy fácil de demostrar con argumentos de geometría elemental (basta razonar sobre los semiplanos delimitados por las rectas determinadas por cada par de puntos de la nube), pero su construcción efectiva por medio de algoritmos eficaces cae dentro del campo de aplicación de la disciplina conocida con el nombre de *Geometría Computacional* [PS]. A continuación procedemos a triangular la envolvente convexa, pero sin introducir nuevos vértices. (De nuevo se trata de un problema típico de la Geometría Computacional.) Obtenemos de ese modo la triangulación de una zona acotada del plano. Hacemos dos copias de la misma y nos aplicamos a retirar, de modo independiente en cada copia, algunos de los triángulos (llenos) que están en la figura. Si en el proceso queda alguna arista que no bordea ningún triángulo podemos elegir entre quitarla o no. De ese modo obtenemos dos nuevos espacios y el rompecabezas consiste en "adivinar" si ambos son iguales desde el punto de vista topológico.

Puesto que "vivimos" es un mundo de plastilina, está claro que la solución del rompecabezas tiene que consistir en empujar (o estirar) aquellas zonas "libres", que no tienen partes adyacentes. Es decir, hay una serie de movimientos admisibles (que respetan la igualdad que nos interesa) que permiten disminuir la complejidad de cada uno de los espacios, hasta reducirlos a una forma en la que se puede de-

cidir con sencillez si son o no "iguales". En realidad basta con considerar un único tipo de "jugada": si un triángulo lleno tiene una arista que no bordea ningún otro triángulo, puede ser "empujado" sobre sus otras dos aristas. Esta operación, que técnicamente se denomina *colapso*, se aplica de modo natural a las aristas y nos permite ir reduciendo el número de elementos en los espacios, antes de poder decidir "a simple vista". Hay que tener, sin embargo, cuidado con el tipo de igualdad que se considera. Esto es así puesto que, según se ha explicado, por una secuencia de colapsos podemos reducir un triángulo lleno a un solo punto, y está claro que ambos espacios no pueden ser biyectivos y, por tanto, que no pueden ser homeomorfos. Así, si consideramos admisible cualquier tipo de colapso, hemos cambiado la igualdad con la que estamos trabajando. En lugar de considerar el *tipo de homemorfía* hemos pasado al *tipo de homotopía*, una igualdad más débil (que distingue menos espacios). Sin entrar en la definición técnica de esta nueva igualdad, decir que es la que habitualmente se utiliza en el contexto de la Topología Algebraica y, por tanto, la que implícitamente será empleada en el resto del artículo.

Un poco de práctica con rompecabezas de este estilo muestra que la solución depende de dos características: el número de fragmentos en que queda desgajado el espacio y, en cada uno de ellos, el número de "agujeros" que tiene (nótese que el número de agujeros está relacionado con, pero no es siempre igual a, el número de triángulos llenos que hemos eliminado: el lugar que estaba ocupado por varios triángulos adyacentes ha podido dar lugar a un único agujero, si se han eliminado también suficientes aristas y vértices). En cualquier caso, lo que se pretende mostrar con el ejemplo es lo siguiente. El proceso de construir el rompecabezas es programable en un computador, bien con la colaboración del usuario (por ejemplo, dibujando en la pantalla la nube inicial de puntos y eliminando manualmente algunos elementos de los gráficos), bien de forma totalmente automatizado (utilizando para ambas tareas generadores de números pseudo-aleatorios) y llevando a la práctica algoritmos bien conocidos en Geometría Computacional. El objetivo de este trabajo es mostrar, de modo intuitivo, que la *solución* del rompecabezas también puede ser totalmente automatizada, siendo en este caso una aplicación que cae dentro de lo que hemos denominado en la introducción *Computación en Topología*. Las técnicas matemáticas a utilizar provienen específicamente del campo de la Topología *Algebraica*, disciplina que introduciremos en el siguiente apartado.

Sin embargo, antes de terminar esta sección, haremos dos observaciones. Por una parte, pese a que el rompecabezas planteado tiene una solución relativamente simple, no hay que pensar que esta situación es general. Para constatarlo, piénsese

en la generalización del rompecabezas a dimensión tres: nube de puntos en el espacio (en posición general: no más de tres puntos en el mismo plano), clausura convexa, triángulación (con tetraedros en este caso), dos copias y eliminación (en cada una de las copias, de modo independiente) de unos cuantos elementos. Es fácil entender que la situación ahora es mucho más compleja, puesto que la noción de "reducción" de una figura a otra que puede ser sistematizada en el plano, es menos clara en el espacio, donde los recovecos y "laberintos" pueden multiplicarse. Es una cuestión conocida en Topología Geométrica que los problemas que son abordables en el plano, devienen muy difíciles en dimensión 3, para, curiosamente, volver a ser más sencillos en dimensiones altas (el caso paradigmático es la Conjetura de Poincaré, que sólo sigue siendo conjetura en dimensión 3; dimensión en la que, por cierto, pensaba Poincaré al tratar el tema...). En cualquier caso, resulta que el rompecabezas ahora se convierte en un problema combinatorio de mucha dificultad y veremos que, si no este problema concreto, otros problemas del mismo estilo no sólo son difíciles sino incluso *irresolubles* algorítmicamente.

Por otra parte, el lector no debe pensar, desorientado por los ejemplos anteriores, que los problemas topológicos sólo aparecen o en situaciones sumamente abstractas o en pasatiempos para ociosos. Muchas aplicaciones reales, relacionadas con la Informática Gráfica o con el Procesamiento de Imágenes Digitales, tienen componentes de naturaleza topológica (por poner sólo un ejemplo, en medicina, dentro del campo llamado *Tomografía Computerizada*). Otro ejemplo típico es el de los OCR, siglas, en inglés, de los *reconocedores ópticos de caracteres*, ligados a los dispositivos para "escanear" documentos impresos. Pese a que no se trata de un problema puramente topológico, es claro que en nuestro mundo de plastilina podremos distinguir con facilidad, en ocasiones, unas letras de otras (seguramente tras un procesamiento previo de las imágenes digitalizadas: alisamiento de bordes, etc.; de estos temas se ocupa la *Topología Digital*). Por ejemplo, la vocal "**i**" se distingue del resto de vocales pues no puede ser trazada sin levantar la mano del papel, y la "**u**" se distingue de la "**a**" puesto que puede ser colapsada a un punto.

Como se desprende del último ejemplo, cuando un problema de matemáticas es planteado como un problema sobre la igualdad, existen varias tareas relacionadas, pero que no deben confundirse:

(a) reconocer la igualdad;

(b) clasificar (dado un espacio, asignarle un cierto espacio de una lista completa de representantes; en el caso del OCR, reemplazar cada carácter manuscrito por una de las letras de la tipografía de "máquina de escribir");

(c) reconocer la diferencia...

Pese a lo que pudiese parecer, los problemas (a) y (c) distan mucho de ser equivalentes, como explicamos en el siguiente punto.

2. Invariantes en Topología Algebraica

Habitualmente, reconocer la igualdad es un problema mucho más difícil que reconocer la diferencia. La razón es que, para saber que dos objetos son diferentes, basta que lo sean respecto a algún *criterio* específico, mientras que la igualdad respecto a ese criterio no asegura en general la igualdad global de los objetos. Por ejemplo, el número de fragmentos permite distinguir la "**i**" de la "**u**" en el caso del OCR, pero no sirve para diferenciar la "**u**" de la "**a**" ni tampoco, y esto es aún más relevante, para concluir la igualdad topológica de los dos últimos objetos. Esta observación, que tiene incluso implicaciones filosóficas y, desde luego, políticas, se materializa en matemáticas a través de la noción de *invariante*. En el concepto de invariante aparecen involucrados dos dominios matemáticos (el de la Topología y el del Álgebra, por ejemplo). Un invariante es un *proceso* que permite asociar a un objeto X del primer dominio un objeto $i(X)$ del segundo, de modo que las igualdades naturales en los dominios sean respetadas en dicho proceso (si suponemos que el dominio de salida es geométrico, en el sentido de Klein, esta noción de invariante coincide con la habitual: una aplicación que queda invariante por la acción del grupo).

El uso principal de los invariantes, y la razón histórica de su introducción, se centra en lo que hemos llamado problema (c) al final de la anterior sección: sirven para distinguir (en otras palabras, son las formalizaciones de los "criterios de distinción" que hemos evocado más arriba, que se corresponden con las distintas *perspectivas* con las que podemos mirar un mismo objeto). Si tenemos dos objetos A y B que sospechamos que son distintos y encontramos un invariante i tal que $i(A)$ es diferente de $i(B)$ (ojo, que la igualdad con la que comparamos habrá cambiado), podremos concluir que A y B son efectivamente distintos. (Pese a ser éste su uso primigenio, los invariantes también pueden ser utilizados para reconocer la igualdad, e incluso para clasificar, a través de las *familias completas de invariantes*.)

Como se desprende de la descripción informal que acabamos de dar, la noción de invariante es muy amplia y su campo de aplicación se extiende a todas las matemáticas. Así, por ejemplo, si tomamos como dominio de partida los grupos abelianos finitamente generados y como dominio de llegada los números naturales, el proceso que asocia a cada grupo abeliano finitamente generado G su *rango* es un invariante. (Recuérdese que, según el teorema de clasificación de tales grupos,

G será isomorfo a un producto cartesiano de un grupo abeliano finito por $\mathbb{Z}^n$, la n-ésima potencia del grupo de los números enteros $\mathbb{Z}$, para un cierto natural n; en estas condiciones, el número n es el rango de G.) Hablaremos de un *invariante de la Topología Algebraica* cuando el dominio de partida sea alguna subclase de espacios topológicos (o de sus variantes más o menos combinatorias: CW-complejos, complejos simpliciales, etc.), normalmente bajo la igualdad del tipo de homotopía, y como dominio de llegada tengamos una categoría algebraica (grupos, anillos, etc.). Es necesario observar que no todo proceso que asocia a cada elemento del dominio de partida un elemento de la llegada tiene el carácter de invariante. Un ejemplo sencillo, que ha aparecido en la sección anterior, es el del número de vértices en nuestros rompecabezas: queda claro que la operación de colapso (que es "admisible", pues respeta la "igualdad") puede disminuir el número de vértices.

Por otra parte, no todo invariante es *relevante*. Aunque propiedades de este estilo (la relevancia, el interés, la importancia, la utilidad) son muy difíciles de precisar en el lenguaje de las matemáticas, por lo que son excluidas de la definición formal de los conceptos, resulta evidente que no todo invariante será estudiado. Un ejemplo de un tal invariante "inútil" sería el que asocia a cada espacio topológico X la clase $i(X)$ que contiene todos los espacios homeomorfos a él. Pese a que obviamente se trata de un invariante, es claro también que conocer X es más sencillo que conocer $i(X)$ y, por otra parte, que el problema de la igualdad en el dominio de partida es equivalente al problema de la igualdad en la llegada.

Así podemos definir la Topología Algebraica como la parte de la matemática que se ocupa de la definición, estudio y cálculo de invariantes algebraicos asociados a espacios topológicos. La parte de *cálculo* es, por supuesto, la que más nos interesa en este artículo. En la misma definición de la disciplina queda subyacente el que se considera que (el problema de la igualdad en) el Álgebra es más fácil que (el problema de la igualdad en) la Geometría. Sin embargo, para que dicha afirmación sea totalmente cierta, hay que imponer ciertas condiciones tanto a los datos de partida (el *input* en jerga informática), como a los datos de llegada (el *output*) y a los procesos mismos.

Comenzando por el *output*, será necesario que las estructuras algebraicas de llegada satisfagan algunas restricciones para poder distinguirlas de un modo razonable. Así, los grupos deberían ser *finitamente* generados, los espacios vectoriales de dimensión *finita*, etc. Este tipo de condiciones de finitud, que aparecen en todos los trabajos "estándar", no computacionales, en Topología Algebraica (muchos resultados típicos son teoremas en los que se demuestra que tal o cual invariante llega

a estructuras algebraicas de tipo finito), abren la puerta a establecer resultados de calculabilidad *teórica* de los invariantes. Es decir, una vez que se sabe que cierto invariante es de tipo finito es cuando uno puede preguntarse por la existencia de un algoritmo que lo calcule.

Pero si queremos pasar de la informática teórica a la informática práctica (la que se ocupa del diseño de programas de computador), es necesario que también los *input* sean *razonables*. Pero en esto, como en el punto anterior, los que nos hemos dedicado a la informática (práctica) en Topología hemos tenido suerte: lo que es manejable para el computador coincide hasta cierto punto con lo que es cómodamente manejable por los humanos, por lo que históricamente, conforme se iba avanzando en el estudio de invariantes se iba produciendo un proceso de "algebraización" de los espacios topológicos, de modo que los invariantes pudiesen ser leídos (por los topólogos) en ellos de manera más sencilla. De todo ese esfuerzo investigador, surgieron una serie de *modelos combinatorios* de espacios topológicos. Junto a los ya citados CW-complejos y complejos simpliciales, el modelo más empleado es el de los *conjuntos simpliciales* debido a Kan (véase [May]). Para comprobar de otro modo la naturalidad de este paso de los espacios abstractos a sus versiones discretas, baste observar que todos los ejemplos elementales que han aparecido en la sección anterior admiten estas descripciones combinatorias (los espacios de los rompecabezas, las imágenes en la pantalla del computador, las letras digitalizadas del OCR). Así pues, años de investigación en matemática pura, han puesto a nuestro servicio todas las herramientas para plantear el diseño de programas de computador para la Topología Algebraica: tenemos como entrada una descripción combinatoria de un espacio topológico y el programa debe calcular una descripción finita de una cierta estructura algebraica. Sin embargo, como veremos más adelante, no todos los invariantes serán susceptibles de ser calculados (lo que hemos llamado antes restricciones en el *proceso*). Comenzaremos no obstante por un invariante que sí es fácil de calcular: el número de componentes conexas.

3. Primer ejemplo: número de componentes conexas

Cuando hemos comentado la solución del rompecabezas de la sección 1, hemos dicho que dependía, en parte, del número de "fragmentos" que quedasen tras el proceso de eliminación de triángulos, aristas y vértices. El concepto formal que corresponde a esa noción de "fragmento" es el de *componente conexa*. Por ejemplo, diríamos, en el ejemplo del OCR, que la letra "**u**" tiene una componente conexa, mientras que la letra "**i**" tiene dos. El proceso que a cada espacio topológico asocia el número de componentes conexas es un invariante que podemos considerar

perteneciente a la Topología Algebraica, a condición de pensar en la Aritmética (pues el dominio de llegada es el de los números naturales) como una parte (¿la más sencilla?) del Álgebra.

Vamos a describir un algoritmo de cálculo del número de componentes conexas, apoyándonos en el espacio que aparece dibujado en la Figura 1. Se trata de un espacio topológico triangulado y que, por tanto, admite una descripción como complejo simplicial. En concreto, la entrada para el algoritmo podría ser una secuencia de listas: **[(0,1,2),(1,3,4),(6,7,8),(2,3),(3,5),(4,5)]**. Nótese que aunque, por conveniencia, los vértices han sido etiquetados con números, no hubiese habido problema en considerar etiquetas que fuesen otros *símbolos*, como letras **a, b,...**, etc. Esta observación sirve para resaltar que los algoritmos en Topología caerán dentro del campo de las Ciencias de la Computación que se denomina *Cálculo Simbólico*, puesto que los ingredientes de base serán más símbolos que otros datos como pueden ser los números reales en precisión finita (datos usuales en las aplicaciones de *Cálculo Numérico*, entre otras).

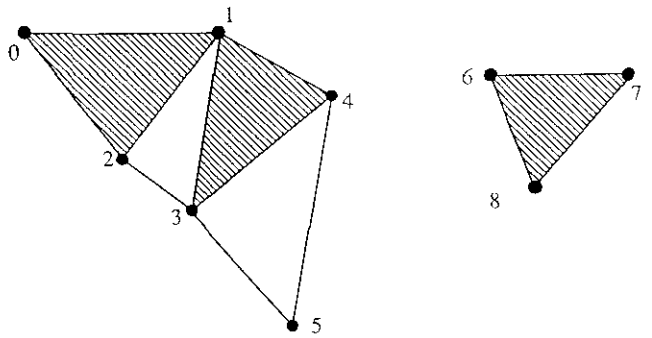

Figura 1

Si uno mira la figura sin prejuicios, hay que reconocer que el problema parece sumamente sencillo: el gráfico está compuesto por dos fragmentos, por dos figuras. Sin embargo, esta tarea de reconocimiento que es tan sencilla para los humanos es difícil de automatizar para que sea llevada a cabo por un computador. Dotar a estas máquinas de capacidades de interpretación de datos visuales es tarea compleja, que cae dentro de la *Inteligencia Artificial*, y específicamente de la *Visión Artificial*. Es por ello que la aproximación simbólica (pese a requerir un trabajo previo de *representación* de los datos de entrada) es mucho más directa a la hora de atacar el problema planteado.

El algoritmo para determinar el número de componentes conexas reposa en la

construcción de un árbol obtenido a partir del grafo constituido por los vértices y aristas del complejo simplicial (el número de componentes conexas sólo depende de los elementos de dimensión 0, los vértices, y de dimensión 1, las aristas; lo que técnicamente se conoce por el nombre de 1-*esqueleto* del espacio). El árbol en cuestión es lo que se denomina *árbol maximal* de un grafo (*spanning tree*, en inglés): cualquier árbol que contenga todos los vértices del grafo (aunque en nuestro caso concreto se trata más bien de determinar *bosques* maximales).

Es destacable la existencia de estructuras de datos que aparecen, de un modo que podríamos calificar de ubícuo, en todos los campos de la Informática: ya sea en las aplicaciones a las finanzas, a las bases de datos, a los sistemas operativos, a Internet, a la Inteligencia Artificial o incluso a un campo tan abstruso (por abstracto) como es la Topología Algebraica. En todos ellos veremos aparecer listas, árboles, grafos, ... Y no sólo se repiten estas estructuras que podríamos denominar *universales*, sino también los algoritmos de manipulación de las mismas. En el caso que nos ocupa hay cientos de líneas de artículos (¡y de código máquina!) dedicadas al cálculo, en diferentes contextos, de árboles maximales. Cuando el topólogo-informático novato llega con su problema de cálculo del número de componentes conexas no tiene más que buscar en la literatura el algoritmo que más convenga a sus necesidades (véase, por ejemplo, [BB]).

Esencialmente, hay dos modos de recorrer un grafo con el objeto de construir un árbol maximal: *en profundidad* y *en anchura*. Explicamos brevemente el primero, dejando como ejercicio al lector que descubra en qué puede consistir el segundo. En el problema concreto que nos ocupa y con el ejemplo de la Figura 1 el proceso sería el siguiente. Comenzamos marcando el vértice 0 como "visitado". A continuación buscamos un vértice adyacente a 0 (el más pequeño, por ejemplo) tal que no haya sido visitado previamente: en este caso el vértice 1 será el siguiente marcado como "visitado". Repetimos el proceso desde 1: puesto que 0 ya ha sido visitado, es el 2 el siguiente vértice marcado. La búsqueda continúa de este modo hasta que no se encuentran más vértices (no previamente visitados) alcanzables por este método, utilizando si es necesario "vuelta atrás" (*backtracking*, en inglés). Así hemos detectado una componente conexa. Si no queda en el grafo ningún vértice no marcado, el algoritmo termina (y devuelve el número 1). Si se encuentran vértices no visitados, se elige uno de ellos (el más pequeño, digamos) y se repite el proceso anterior, llevando la cuenta de las componentes conexas que van apareciendo. En el espacio de la Figura 1, recomenzaríamos marcando el vértice 6, a partir del que son alcanzados los vértices restantes (el 7 y el 8) y se concluye, como el sentido común

quiere, que el número de componentes conexas es 2.

Este algoritmo es fácil de programar, incluso para principiantes, pero hay que reconocer que la información *algebraica* obtenida es bastante escasa. En el siguiente apartado nos ocupamos de otro invariante que llega, esta vez sí, a un dominio algebraico.

4. Segundo ejemplo: grupo fundamental

El grupo fundamental es uno de los invariantes más conocidos, de los que primero se introducen desde el punto de vista de la docencia y que tiene aplicaciones en muchos campos de las matemáticas (por ejemplo, en relación con la integración en variable compleja: teorema del índice de Gauss, etc.). Informalmente hablando, el grupo fundamental mide como está "agujereado" el espacio en dimensión 2. Para hacer un poco más precisa dicha descripción necesitamos algunos preliminares algebraicos (para una exposición más completa sobre el grupo fundamental podemos citar, entre otros muchos, el libro [Mau]).

4.1. Generadores y relaciones

El punto de partida para las construcciones que vamos a explicar es un conjunto $A = \{a, b, c, \ldots\}$, que denominaremos *alfabeto* (y, en concordancia, a sus elementos los llamaremos *letras*). A continuación consideramos *palabras* sobre A: secuencias de letras (por ejemplo, $abcaabc$). Llamemos $M[A]$ al conjunto de palabras sobre A junto a un nuevo símbolo ω ("nuevo" significa que $\omega \notin A$), que representa la *palabra vacía*, la palabra que no contiene ninguna letra. Ahora dotamos a $M[A]$ de una estructura de *monoide* (conjunto con una operación binaria interna que es asociativa y con elemento neutro). Para ello, consideramos la operación de concatenación de palabras, operación evidentemente asociativa y que tiene a la palabra vacía ω como elemento neutro. Hemos construido así el *monoide libre* sobre A. Esta definición matemática tiene aplicaciones en múltiples campos de la Informática: compiladores, tipos abstractos de datos, lenguajes formales, etc. En este trabajo nos interesa, ante todo, por ser la base para la construcción de un *grupo libre* sobre un conjunto, como explicamos a continuación.

Sea $X = \{x, y, z, \ldots\}$ un conjunto y consideramos un conjunto biyectivo con él, al que denotaremos X^{-1}. Por cada elemento $x \in X$ habrá un elemento $x^{-1} \in X^{-1}$. Nótese que la expresión x^{-1} no es más que una notación para señalar que ese elemento es la imagen de x por una biyección canónica establecida entre X y X^{-1}. Pero, como siempre sucede, la simbología tiene una intencionalidad: x^{-1}

tendrá el papel de inverso *formal* de x. Si pensamos que el conjunto de partida X es ya simbólico (un tipo enumerado en un lenguaje concreto de programación, por ejemplo), vemos que la construcción de X^{-1} puede ser automatizada, por una manipulación *simbólica*, formal (como curiosidad señalar que el Cálculo Simbólico, que correspondería a *Symbolic Computation* en inglés, en francés de denomina *Calcul Formel*). En cualquier caso, a continuación construimos $M[X \cup X^{-1}]$, el monoide libre sobre la reunión disjunta de X y X^{-1}. Y ahora introducimos una serie de *relaciones* de los tipos $xx^{-1} \equiv \omega$ y $x^{-1}x \equiv \omega$, por cada $x \in X$. Estas igualdades inducen relaciones entre palabras (por ejemplo: $xyy^{-1}x \sim xx$) y tenemos así definida una relación de equivalencia $\sim$ sobre $M[X \cup X^{-1}]$. Se comprueba fácilmente que la operación de concatenación pasa al cociente y que también está bien definida en el cociente la operación que asocia a cada palabra su "inversa": $(xyz)^{-1} := z^{-1}y^{-1}x^{-1}$. De este modo, hemos dotado al conjunto cociente $M[X \cup X^{-1}] \ / \sim$ de una estructura de grupo, que se denomina *grupo libre (no conmutativo) sobre* X y que denotaremos por $G[X]$. Dicho grupo es siempre infinito, excepto en el caso en que $X = \emptyset$, en el que se obtiene el grupo trivial (con un único elemento). Es siempre no conmutativo, excepto si $X = \emptyset$ o si $X = \{x\}$, caso en el que es isomorfo al grupo de los números enteros con la suma. Una observación importante es que los elementos de grupos libres pueden ser representados de modo *seguro* por medio de palabras (es decir, no es necesario trabajar con las clases de equivalencia: basta elegir representantes), debido a la existencia de *formas canónicas* (este concepto es central en Cálculo Simbólico, pues permite un tratamiento eficiente y consistente de los espacios cociente). La forma canónica coincide en este caso con la palabra de menor número de letras en la clase y, además, las relaciones permiten *reescribir*, *reducir*, cada palabra a su forma canónica: $zx^{-1}yy^{-1}xz \rightarrow zx^{-1}xz \rightarrow zz$.

Una vez introducida la noción de grupo libre sobre un conjunto X, podemos pasar al concepto de *grupo definido por generadores y relaciones*. Para definir un tal grupo necesitamos dos datos: un conjunto X (cuyos elementos denominamos *generadores*) y un conjunto R de palabras (reducidas a forma canónica) extraídas de $M[X \cup X^{-1}]$ (las *relaciones*). El grupo entonces será un cociente del grupo libre $G[X]$ por la relación de equivalencia inducida por los elementos de R. Veamos como funcionaría dicha construcción en un ejemplo concreto, el del grupo $\langle x, y \mid xy \equiv y^{-1} \rangle$. En esta notación, usual en *Teoría Combinatoria de Grupos* (la disciplina que se encarga del estudio de este tipo de grupos), debe entenderse que el conjunto X de generadores es $\{x, y\}$ y que, en este caso concreto, hay una única relación. Se suelen escribir las relaciones no como palabras, sino como *pares* de palabras: $xy \equiv y^{-1}$, pues dicha

relación también puede entenderse como $xyy \equiv \omega$ o, simplificadamente, xyy (o yxy) que sería el verdadero elemento de R. De esta pequeña disquisición se puede deducir fácilmente, apoyados por el uso que hemos hecho de las *relaciones* xx^{-1} para pasar del monoide al grupo libre, cómo se operaría en el grupo definido por dichos generadores y relaciones: $xyyx^{-1}y^{-1} \sim y^{-1}yx^{-1}y^{-1} \sim x^{-1}y^{-1} \sim y$. Aunque volveremos sobre este punto más adelante, nótese que no hemos utilizado, pese a que el ejemplo invitaba a ello, el símbolo de *reescritura* $\rightarrow$ sino el de relación $\sim$: establecer qué es una *reducción* en un grupo definido por generadores y relaciones es una cuestión delicada. Pero dejemos por el momento el dominio de llegada de nuestro invariante, el Álgebra, y volvamos al dominio de partida: la Topología.

4.2. Presentación del grupo fundamental

Como el lector habrá sospechado el grupo fundamental va a ser definido por generadores y relaciones. Pero, ¿cómo obtener de objetos geométricos (vértices, aristas,...) objetos "lingüísticos" (letras, palabras, ...)? Las claves para responder a esta pregunta son, desde el punto de vista algorítmico, la manipulación simbólica y, desde el punto de vista conceptual, la noción de *camino* en un grafo. Comenzaremos por la descripción algorítmica. Trataremos en primer lugar el caso en el que no hay triángulos llenos (es decir, el complejo simplicial coincide con su 1-esqueleto o, dicho en palabras llanas, es un grafo) y nos apoyaremos en el espacio de la Figura 2.

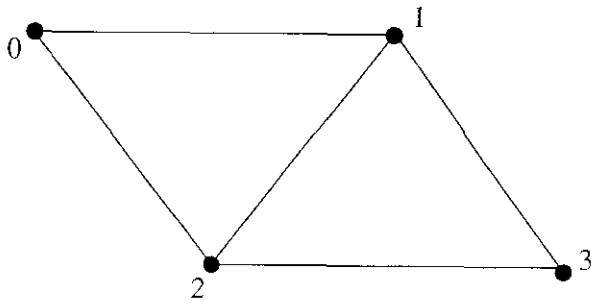

Figura 2

Es necesario para poner en marcha el algoritmo fijar un vértice, que hará el papel de *punto base* para el grupo fundamental, pues, de hecho, habrá un grupo asociado a cada componente conexa del espacio. En el complejo de la Figura 2 sólo habrá un grupo fundamental pues es *conexo* (esto es, con una única componente conexa). En estas condiciones, el proceso es independiente del vértice por el que comience y podemos elegir, por ejemplo, el 0 como punto base. Ahora calculamos un árbol

maximal basado en 0, utilizando por fijar ideas (el resultado también es independiente del árbol maximal elegido) el algoritmo explicado en la sección anterior. El árbol estará constituido por las aristas 01, 12 y 23 (el lector es invitado a dibujar sobre la Figura 2, con lapicero a poder ser). Entonces, el grupo fundamental es el grupo libre generado por las otras aristas: $G[\{02, 13\}]$. Ésta es la parte algorítmica, *ciega*, de nuestra construcción (no más difícil de programar, por otra parte, que el cálculo del número de componentes conexas). Para obtener una interpretación geométrica hemos de apoyarnos en la noción de camino por aristas. Un tal camino, en la Figura 2, puede ser representado por: $01 \rightarrow 13 \rightarrow 32$. En realidad, los caminos que nos interesarán serán los *ciclos* basados en el punto base 0 (caminos que comienzan y terminan en 0, como $01 \rightarrow 12 \rightarrow 20$). Informalmente, podemos decir que el grupo fundamental es el grupo de los ciclos basados en 0 y operados por concatenación: $(01 \rightarrow 12 \rightarrow 20) * (01 \rightarrow 13 \rightarrow 32 \rightarrow 20) := 01 \rightarrow 12 \rightarrow 20 \rightarrow 01 \rightarrow 13 \rightarrow 32 \rightarrow 20$. Si de esa cadena eliminamos las aristas que están en nuestro árbol maximal (y sus "inversas") y escribimos, como es natural, $20 \equiv 02^{-1}$, obtenemos 02^{-1}–13–02^{-1}, que puede ser identificado evidentemente con un elemento del grupo libre $G[\{02, 13\}]$. Recíprocamente, dado un elemento de $G[\{02, 13\}]$ podemos recuperar un único ciclo basado en 0 (único al menos si es canónico: sin "idas y vueltas inútiles", como $02 \rightarrow 20$). Expliquemos el proceso con el generador 13. Puesto que 1 es un vértice del árbol maximal (todos los vértices del espacio *conexo* lo son), existe un único camino dentro de dicho árbol que une la raíz 0 con 1: en nuestro ejemplo el camino consta de una única arista 01. Del mismo modo, el vértice 3 es unido con 0 de forma única por medio de: $32 \rightarrow 21 \rightarrow 10$. Entonces, concatenando adecuadamente, obtenemos el correspondiente ciclo basado en 0: $01 \rightarrow 13 \rightarrow 32 \rightarrow 21 \rightarrow 10$. Vemos pues que la relación entre la construcción algorítmica y la geométrica es natural e intuitiva.

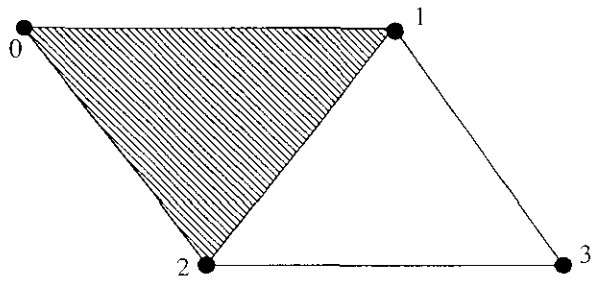

Figura 3

Notemos que este invariante sirve para distinguir, en el caso del OCR, la "**u**" (grupo fundamental trivial) de la "**a**" (grupo fundamental isomorfo al grupo de los números enteros con la suma). Lo que sabemos por el momento no nos permite, sin embargo, aplicar el invariante al caso del rompecabezas, pues en él pueden aparecer involucrados triángulos *llenos*. Para ocuparnos de este caso (que por otra parte es el caso *general*, pues el grupo fundamental de cualquier espacio triangulado sólo depende de su 2-esqueleto) nos apoyaremos en el gráfico de la Figura 3 que, de hecho, no difiere del de la Figura 2 más que en la existencia del triángulo lleno de vértices 0, 1 y 2.

Como es fácil de imaginar, la solución de este caso pasa, como etapa previa, por la del caso anterior. En efecto, consideramos el 1-esqueleto del espacio y calculamos los generadores de su grupo fundamental; en nuestro ejemplo: 02 y 13. A continuación, por cada triángulo lleno que hubiese en el espacio de partida incluimos una relación; en el ejemplo: $(01 \rightarrow 12) \equiv 02$. Es ésta una operación simbólica, pero que puede no tener sentido desde el punto de vista del Álgebra: algunas de las aristas que aparecen en las relaciones así introducidas pueden estar en el árbol maximal y, por tanto, no pueden formar parte de palabras de $G[\{02, 13\}]$. Pues bien, estas aristas que nos molestan, simplemente las eliminamos, obteniendo de este modo un verdadero grupo definido por generadores y relaciones, que no es otro que el grupo fundamental del espacio. En nuestro ejemplo, tendríamos: $\langle 02, 13 \mid (01 \rightarrow 12) \equiv 02 \rangle \simeq \langle 02,13 \mid \omega \equiv 02 \rangle \simeq G[\{13\}]$. Es decir, se trata del grupo libre con un generador.

En cuanto a la interpretación geométrica de esas manipulaciones formales es sencilla: puesto que vivimos en un mundo de plastilina, es equivalente unir el vértice 0 con el 2 directamente a través de la arista 02 que pasando por 1 por medio de $01 \rightarrow 12$ (la "materia" del triángulo lleno permite deformar continuamente 02 sobre $01 \rightarrow 12$). Así, podemos decir, informalmente, que el grupo fundamental mide el número de ciclos (independientes) que no acotan discos en el espacio topológico. (En el ejemplo de la Figura 3 el único ciclo fundamental es el borde del triángulo no lleno 123, que es representado en el grupo por medio del generador 13.)

4.3. Los límites de la Informática

Llegados a este punto, podríamos sentirnos satisfechos: hemos conseguido asociar a cada espacio topológico (o, al menos, a aquellos que pueden ser triangulados con un número finito de elementos) su grupo fundamental, y ello por medio de un algoritmo que es fácil de implementar en el computador. Podríamos intentar aplicar

dicho mecanismo a espacios topológicos (por ejemplo, a los que aparecían en los rompecabezas) para distinguirlos unos de otros. Ahora bien, ¿cuánto sabemos de la estructura algebraica así definida? No conviene confundir lo que "se sabe definir" con lo que es "completamente conocido" (piénsese en Análisis, cuando se definen funciones por medio de un teorema de existencia y unicidad de solución para alguna clase de ecuaciones diferenciales: dada una de esas ecuaciones elegidas al azar, ¿cuánto sabemos del objeto matemático que es la solución de dicha ecuación?). Lo que conocemos del grupo fundamental es que está definido por generadores y relaciones: $\langle g_1, \ldots, g_n \mid r_1, \ldots, r_m \rangle$, con $r_i \in G[\{g_1, \ldots, g_n\}], \forall i = 1, \ldots, m$. Pero, ¿es dicho grupo trivial? ¿Es finito? ¿Es conmutativo? Dado un camino por aristas, ¿representa al elemento neutro del grupo? Son ejemplos de problemas *indecidibles*, es decir, problemas para los que se puede demostrar que nunca habrá un programa de computador que los resuelva. Así, nada más comenzar nuestra andadura por la Topología Algebraica hemos topado con un invariante que pone al descubierto los límites de la Informática. Bien es cierto que se suele esquivar el problema diciendo que se ha calculado una *presentación* (por generadores y relaciones) del grupo fundamental (no confundir con la teoría de *representaciones* de grupos) y no el grupo en sí. Pero esto no deja de ser una suerte de eufemismo, pues la realidad es que un invariante (numérico, como en el caso del número de componentes conexas) tan "inofensivo" como es el que asocia a cada espacio conexo la cardinalidad de su grupo fundamental (fijando, por convenio, que si el grupo es infinito devolvemos el número 0, por ejemplo) es no calculable, define un problema irresoluble desde el punto de vista algorítmico.

Pese a que este tipo de problemas cierran la puerta al tratamiento informático, está claro que las matemáticas pueden proseguir su estudio: los matemáticos han sido capaces de *demostrar* que son irresolubles por medio del computador. Pero si se medita un poco, uno puede preguntarse en qué consiste demostrar que algo es no calculable. Si se trata de mostrar que algo es calculable, el procedimiento es claro: basta describir un algoritmo que lo calcule. Tampoco cuesta mucho hacerse a la idea de que hay gran número de problemas complejos para los que *hasta el momento* no se ha encontrado ningún algoritmo que los resuelva. Sin embargo, establecer que nunca nadie, por ingeniosa que sea su mente o potentes que sean los métodos y recursos que tenga a su disposición, podrá encontrar una solución parece una afirmación de mucho calado. Los lógicos han inventado una técnica para abordar este espinoso asunto: la reducción (calculable) a un problema no calculable. Supongamos que sospechamos que un cierto problema $\mathcal{P}$ es indecibible y supongamos también que conocemos un problema irresoluble $\mathcal{Q}$. Si describimos un algoritmo que para cada

caso q de $\mathcal{Q}$ es capaz de construir otro p de $\mathcal{P}$, con la propiedad de que la solución de p significa la solución de q, habríamos demostrado la no calculabilidad de $\mathcal{P}$ (pues en caso contrario, componiendo los algoritmos, tendríamos una solución algorítmica para $\mathcal{Q}$). Esta técnica ha sido utilizada en múltiples ocasiones, de modo que en la actualidad disponemos de un amplio catálogo de problemas no resolubles (lo que, por supuesto, aumenta las posibilidades de encontrar otros nuevos...).

Pero de algún modo debe empezar la cadena para poder aplicar esa técnica de reducción. Y, en esencia, hay un único problema irresoluble demostrado partiendo de cero, sin recurrir al método de reducción: *el problema de la parada de Turing*. El problema consiste en decidir si la ejecución de una máquina de Turing (un predecesor rudimentario y *virtual* de los actuales computadores) termina. Turing, en los años 30, inventó las máquinas que ahora llevan su nombre y demostró que si el problema de la parada fuese resoluble se alcanzaría una contradicción con principios lógicos que sustentan la idea misma de algoritmo (y también los fundamentos de las matemáticas). Una consecuencia práctica de dicho resultado es que nunca será posible diseñar un programa de ordenador que tomando como entrada el texto de otro programa cualquiera decida si su ejecución termina o no (lo que, dicho sea de pasada, supone una cortapisa insalvable para el problema *general* de la demostración automática de la corrección de algoritmos: mal podremos deducir automáticamente si un programa calcula resultados correctos si ni siquiera podemos determinar si termina o no su ejecución).

Dicho problema de la parada está, en particular, en la base de los resultados de no calculabilidad en Topología. Abreviadamente, la historia es la que sigue. Novikov demostró, apoyándose en el problema de la parada (o más bien en resultados de Post sobre semigrupos, que son monoides sin elemento neutro), la irresolubilidad del *problema de la palabra* en grupos definidos por generadores y relaciones: no puede existir algoritmo que dada una palabra decida si representa al elemento neutro o, dicho de otro modo, si es equivalente dentro del grupo a la palabra vacía. Esto muestra que, en general, es difícil imaginar un concepto de "reducción a forma canónica" en un grupo presentado por generadores y relaciones. Ahora, dada una presentación por generadores y relaciones de un grupo, se construye de manera algorítmica un complejo simplicial tal que su grupo fundamental sea isomorfo al grupo de partida. De ahí se deduce que el problema de la *simple conexión* es indecidible. (Un espacio topológico conexo se dice *simplemente conexo* si su grupo fundamental es trivial.) Esta propiedad de no calculabilidad va a condicionar todo desarrollo algorítmico en Topología Algebraica: los resultados de calculabilidad

irán frecuentemente acompañados de la hipótesis de simple conexión en los datos de entrada, hipótesis no calculable...

Terminamos este pequeño paseo por la indecidibilidad señalando que pese a que hemos presentado la Topología Algebraica como una disciplina subsidiaria, que sirve de apoyo para la Topología (pues sólo sirve para distinguir, no para reconocer la igualdad), en estos resultados de tipo "negativo" la flecha va en el otro sentido: de la no decidibilidad de la simple conexión se puede deducir (por reducción) que el problema general del homemorfismo es también irresoluble algorítmicamente. Es decir, no existe algoritmo que tome como argumento (la descripción finita de) dos espacios topológicos y decida si son o no homeomorfos (este resultado, en el caso particular de *variedades*, fue demostrado por Markov).

4.4. Los grupos de homotopía de orden superior

La notación con la que los topólogos se suelen referir al grupo fundamental de un espacio X es $\pi_1(X)$ (o $\pi_1(X;x)$ si es necesario destacar un punto base $x \in X$), y a dicho grupo se le denomina también *primer grupo de homotopía*. Esto significa, por supuesto, que existen otros invariantes que se denominan *grupos de homotopía* (de orden superior) y que se denotan $\pi_n(X)$, con n un número natural mayor que 1. Sin entrar en la definición de los mismos, indiquemos que son conmutativos si $n > 1$, y que bajo buenas condiciones (los espacios X deben ser finitos en algún sentido y, sobre todo, *deben* ser simplemente conexos) son calculables. Sin embargo, los algoritmos que se conocen son complicados, tanto por las matemáticas que aparecen en ellos como por las dificultades de implementación concreta (requieren muchos recursos en tiempo de ejecución y en espacio de memoria), y su descripción escapa de las posibilidades de este texto. Señalemos, no obstante, que ha habido mucho trabajo computacional sobre el tema, sobre todo en el caso particular de los *grupos de homotopía de esferas* (véase, a título de ejemplo, la memoria [T]). En el caso de los grupos de homotopía de esferas se han desarrollado técnicas específicas muy especializadas, pero todos los algoritmos *generales* (es decir, aplicables a clases muy amplias de espacios topológicos) para el cálculo de grupos de homotopía se basan en unos invariantes más simples de calcular, en los que nos concentramos a continuación: los grupos de homología.

5. Tercer ejemplo: grupos de homología

Los grupos de homología de un espacio X son, al igual que los grupos de homotopía, una familia de invariantes $H_n(X)$ indexada por los números naturales $n \geq 0$. Todo manual de Topología Algebraica se ocupa de definir los grupos de

homología; sin embargo, la presentación constructiva que vamos a esbozar aquí no aparece en la mayoría de ellos (el libro [Mu] es una excepción), lo que me sorprende un poco, pues es sencilla y es una buena fuente de ejemplos *concretos*.

Los primeros grupos de homología de cualquier espacio son fáciles de describir y, de hecho, tienen mucho que ver con nuestros dos ejemplos anteriores. Así, $H_0(X) = \mathbb{Z}^k$, el grupo *abeliano* libre con k generadores, donde k es el número de componentes conexas de X. Si X es conexo (es decir, si el k anterior es igual a 1), el grupo $H_1(X)$ es el *abelianizado* de $\pi_1(X)$, el grupo fundamental de X. Esto implica que conocemos algoritmos sencillos para calcular los primeros grupos de homología. Esta afirmación, obvia en el caso de $H_0(X)$, podría chocar al lector tras nuestra insistencia en la no calculabilidad de $\pi_1(X)$. Pero sucede que, en presencia de la conmutatividad, los problemas difíciles devienen sencillos (definitivamente, vivir en un mundo conmutativo sería muy aburrido): si a la presentación por generadores y relaciones de $\pi_1(X) \simeq \langle g_1, \ldots, g_n \mid r_1, \ldots, r_m \rangle$ le añadimos las relaciones $g_i g_j \equiv g_j g_i, \forall i, j = 1, \ldots, n$, inmediatamente disponemos de formas canónicas para el grupo así abelianizado $H_1(X)$ y podemos caracterizar completamente de qué grupo se trata (en particular, si es trivial, si es finito, etc.).

Estos dos invariantes $H_0(X)$ y $H_1(X)$ dan una solución *algorítmica* completa (bajo la igualdad de la homotopía, no la del homeomorfismo) de los rompecabezas planos de los que hemos hablado al comienzo del artículo. (De hecho, el número de componentes conexas y el grupo fundamental nos hubiesen también bastado, pues es fácil demostrar que en el caso de esas figuras planas el π_1 es un grupo libre, por lo que el problema de la palabra es resoluble y el número de generadores determina totalmente el grupo y el tipo de homotopía de cada espacio conexo.)

Por tanto, podemos decir que $H_1(X)$ mide "conmutativamente" el número de agujeros de X en dimensión 2 y, generalizando, que $H_2(X)$ mide lo equivalente en dimensión 3, etc. Si queremos una descripción más precisa, tenemos que apoyarnos en un resultado del Álgebra que ha aparecido previamente: el teorema de clasificación de grupos abelianos finitamente generados. En efecto, siendo los $H_n(X)$ siempre conmutativos, si X es de tipo finito (en nuestro contexto esta condición significa que X está triangulado por un número finito de elementos), entonces $H_n(X)$ es finitamente generado y por tanto, aplicando el teorema de clasificación, se sabe que es isomorfo a $\mathbb{Z}^k \times \mathbb{Z}/q_1\mathbb{Z} \times \ldots \times \mathbb{Z}/q_m\mathbb{Z}$, con $k, q_1, \ldots, q_m$ números naturales (y denotando $\mathbb{Z}/q\mathbb{Z}$ el grupo abeliano finito de q elementos de los enteros modulo q). Por tanto, determinar un tal grupo es equivalente a calcular $(k; q_1, \ldots, q_m)$. Obsérvese que pese a que se trata de un invariante algebraico admite (por ser un

output "razonable") ser representado de modo numérico. Nótese también que la lista $(k; q_1, \ldots, q_m)$ constituye para los grupos abelianos finitamente generados lo que hemos denominado en la sección 2 una *familia completa de invariantes* (con dominio de partida algebraico, en este caso) que sirve, evidentemente, para clasificar. Indiquemos, por último, que al rango k se le denomina también *número de Betti* del grupo (en honor al topólogo Betti) y que los $q_1, \ldots, q_m$ se conocen como *coeficientes de torsión* del grupo (donde la palabra *torsión* tiene un sentido geométrico en los espacios que "se tuercen sobre sí mismos"), lo que no sería comprensible sin conocer la génesis *geométrica* del estudio de los grupos abelianos finitamente generados: históricamente, fueron los matemáticos (y Betti entre ellos) que se dedicaron a la Topología (conocida entonces con el nombre de *Analysis Situs*) los que introdujeron los conceptos clave sobre los grupos abelianos y el famoso teorema de clasificación. (El estudiante de matemáticas debería comparar esta breve reseña histórica con la presentación *compartimentada* que se hace actualmente de estas materias: el Álgebra es Álgebra y, después, es utilizada en Topología Algebraica.)

Sea como fuere, hemos reducido el problema de conocer los grupos de homología $H_n(X)$ al de calcular, para cada n, la lista $(k; q_1, \ldots, q_m)$, es decir los números de Betti y los coeficientes de torsión. El algoritmo para calcularlos es elemental y su descripción nos va a permitir, simultáneamente, definir formalmente los grupos de homología. Nos apoyaremos de nuevo en el ejemplo de la Figura 3. El proceso comienza definiendo los *grupos de cadenas* del espacio X: $C_*(X)$. En el ejemplo, puesto que sólo hay elementos de dimensión 0, 1 y 2, sólo tendremos tres grupos de cadenas no triviales:

$C_0(X) := \mathbb{Z}^4$, generado por los vértices $\{0, 1, 2, 3\}$;

$C_1(X) := \mathbb{Z}^5$, generado por las aristas $\{01, 12, 02, 13, 23\}$;

$C_2(X) := \mathbb{Z}$, generado por el triángulo lleno $\{012\}$.

Vamos ahora a hacer un poco de "álgebra lineal" (aunque no sea sobre un cuerpo, sino sobre el anillo $\mathbb{Z}$), para definir un homomorfismo $d_2 : C_2(X) \to C_1(X)$. Dicho homomorfismo se define sobre los generadores a partir de la relación "ser cara de": $d_2(012) := 01+12+20 \equiv 01+12-02$ (los signos provienen de la elección previa de una orientación sobre los triángulos y aristas del espacio). Definido sobre los generadores, existe una única extensión a un homomorfismo de grupos. Sobre aristas tendríamos análogamente $d_1 : C_1(X) \to C_0(X)$ tal que, por ejemplo, $d_1(01) := 1-0$. Obsérvese el carácter simbólico, formal, de esta operación (es decir, en ningún caso hay que identificar 1–0 con 1), pues los números no son sino etiquetas, sin significado

aritmético. Sería tal vez más claro definir, genéricamente, $d_1(ab) := b-a$. Así, en el ejemplo tenemos definido un par de homomorfismos de grupos libres:

$$C_2(X) \xrightarrow{d_2} C_1(X) \xrightarrow{d_1} C_0(X).$$

En general, tendríamos una secuencia de homomorfismos, llamados *operadores de borde* o *diferenciales*:

$$\ldots \to C_{n+1}(X) \xrightarrow{d_{n+1}} C_n(X) \xrightarrow{d_n} C_{n-1}(X) \to \ldots \to C_2(X) \xrightarrow{d_2} C_1(X) \xrightarrow{d_1} C_0(X)$$

que satisfacen $d_n d_{n+1} = 0, \forall n > 0$. Ésta es la condición que plasma la observación geométrica de que "el borde de un objeto no tiene borde" y que, desde el punto de vista simbólico, es puramente combinatoria. Por ejemplo, $d_1 d_2(abc) = d_1(ab) + d_1(bc) - d_1(ac) = b-a+c-b-(c-a) = 0$, donde 0 representa el elemento neutro del grupo abeliano libre $C_0(X)$ (no confundir con la etiqueta 0 que hemos usado para los vértices en las distintas figuras). La estructura algebraica $(C_*(X), d_*)$ se denomina *complejo de cadenas*. Según la definición, en cualquier complejo de cadenas se verifica que la imagen del operador diferencial d_{n+1}, denotémosla por $Im\ d_{n+1}$, está contenida en el *núcleo* de d_n (formado por los elementos de $C_n(X)$ que van por d_n al elemento neutro de $C_{n-1}(X)$). Si como es habitual denotamos por $Ker\ d_n$ al núcleo de d_n (por *kernel*, en inglés) y teniendo en cuenta que estamos trabajando con grupos abelianos, podemos considerar el grupo cociente $Ker\ d_n / Im\ d_{n+1}$. Pues bien, ese cociente es el *n-ésimo grupo de homología* de X, o escribiendo con fórmulas: $H_n(X) := Ker\ d_n / Im\ d_{n+1}, \forall n > 0$.

Nuestro problema algorítmico se reduce pues a calcular, para cada n, el número de Betti y los coeficientes de torsión de ese grupo cociente. El primer paso consiste en construir las *matrices de incidencia* del espacio X que son la expresión matricial de los operadores diferenciales en términos de las bases (sistemas de generadores, más bien) canónicas. En concreto, en el ejemplo de la Figura 3 tendríamos dos matrices:

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} 01 & 12 & 02 & 13 & 23 \end{array} \\ \begin{pmatrix} -1 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array} \qquad \text{y} \qquad \begin{array}{c} \\ 01 \\ 12 \\ 02 \\ 13 \\ 23 \end{array}\!\!\begin{array}{c} \begin{array}{c} 012 \end{array} \\ \begin{pmatrix} 1 \\ 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} \end{array}.$$

Ahora nos aplicamos a diagonalizar estas matrices de enteros utilizando para ello, por ejemplo, una variante del bien conocido (incluso para los estudiantes de primeros cursos de matemáticas o ingeniería) *algoritmo del pivote* (también llamado de *elimi-*

nación de Gauss):

$$\begin{pmatrix} -1 & 0 & -1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \overset{(1)}{\to} \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \overset{(2)}{\to} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

$$\overset{(3)}{\to} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \to \cdots \to \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(donde el paso (1) corresponde a cambiar de signo la primera fila, el (2) a hacer ceros en la primera fila, el (3) a hacer ceros en la primera columna y los puntos suspensivos indican que se repite el proceso con la submatriz obtenida al eliminar la primera fila y la primera columna).

Una vez la matriz diagonalizada, basta con contar el número de entradas no nulas para calcular el núcleo: $Ker\, d_1 = \mathbb{Z}^{5-3} = \mathbb{Z}^2$. Algunas de esas componentes pueden verse "anuladas" por los unos que aparezcan en la matriz diagonal correspondiente a d_2 y otras pueden dar lugar a componentes de torsión (si en la diagonal de d_2 aparecen elementos estrictamente mayores que 1, situación que no se da en nuestro ejemplo). Puesto que en el ejemplo obviamente $Im\ d_2 = \mathbb{Z}$ se deduce que $H_1(X) = \mathbb{Z}$. Resultado que, por otra parte, conocíamos de antemano, puesto que $\pi_1(X) = \mathbb{Z}$, que es ya abeliano.

En cierto modo, este trabajo debería acabar aquí, puesto que hemos descrito procedimientos algorítmicos (y fácilmente trasladables al computador) para resolver el rompecabezas planteado en la sección 1. No me resisto sin embargo a dar unas breves pinceladas sobre los problemas que interesan, más allá de la divulgación, en el área de la Computación en Topología Algebraica.

6. Investigaciones y conclusiones

Podemos afirmar que aquí no termina, sino que empieza, el trabajo sobre la cuestión que nos ocupa. La razón es que, pese a que los algoritmos descritos son fáciles de programar, lo que no es tan fácil es que sean *útiles*: la talla de las matrices y el número de operaciones a realizar en los casos que realmente interesan a los topólogos es tan grande que una implementación irreflexiva será inutilizable para ellos. Dos mejoras evidentes, pero que ya requieren cierta sofisticación desde el punto de vista informático, son representar las matrices de incidencia como *matrices*

dispersas (*sparse*, en inglés; puesto que tienen muchos ceros, no es conveniente almacenarlos todos explícitamente), adaptando el algoritmo del pivote para ellas, y también utilizar técnicas de *aritmética modular* (pues los enteros que aparecen en los pasos intermedios de la diagonalización pueden llegar a ser tan grandes que los computadores pueden ver rebasadas sus capacidades de manipulación). En una línea más conceptual, otra buena estrategia es buscar espacios topológicos más simples en su descripción (con menor número de elementos en la triangulación, por ejemplo) pero del mismo tipo de homotopía que los de partida, para que los algoritmos sean aplicados a estos modelos topológicos "pequeños".

Esta última observación me sirve para insistir en algo que ya dije anteriormente: la Topología Algebraica no es sólo cálculo (ni siquiera es éste un aspecto que sea central en ese área, al menos en su vertiente más explícita, informática). La búsqueda de modelos *minimales*, los teoremas de estructura (¿cómo están "pegados" los distintos invariantes?) o la destilación de qué sea un *tipo de homotopía* (a través de la propuesta de familias completas de invariantes algebraicos), por citar sólo tres líneas, son temas de investigación en Topología Algebraica.

Ahora bien, una característica curiosa, que parece muy específica de la Topología Algebraica, es que para obtener información (finita) sobre espacios topológicos de tipo finito (como las esferas, por ejemplo), los métodos conocidos pasan por la construcción y manipulación de espacios de *tipo infinito* (no se trata sólo de que los conjuntos subyacentes sean de cardinalidad infinita, pues el de una simple arista ya lo es, sino de que no admitan descripciones finitas, que no puedan ser triangulados con un número finito de elementos, por ejemplo). Uno de esos espacios de naturaleza infinita es el *espacio de lazos* de un espacio X. Es éste un conjunto ΩX de funciones (concretamente, aplicaciones de la circunferencia S^1 en X), que es muy útil para estudiar propiedades homotópicas de X. La utilización en la computación práctica de dichos espacios infinitos parecía un verdadero desafío. Hacia mitad de los años 80, Sergeraert puso a punto unas técnicas (la *codificación funcional* y la *homología efectiva* [Se]) que permitieron más adelante desarrollar una serie de programas de computador [RSS], [SS] para el cálculo de grupos de homología de espacios de lazos iterados (o para ser más preciso, de las versiones combinatorias, como conjuntos simpliciales, de estos espacios [May]) y de grupos de homotopía.

Dichos sistemas, llamados respectivamente EAT (Effective Algebraic Topology) y Kenzo, necesitaron en su desarrollo la utilización de técnicas informáticas avanzadas, que caen dentro de lo que se denomina *Programación Funcional*. Con este punto de partida, mi interés fue derivando hacia temas cada vez más relacionados con

la Informática Teórica, al comprobar que la experiencia adquirida en la construcción de sistemas de Cálculo Simbólico para la Topología Algebraica, podía ser utilizada para realizar aportaciones en las Ciencias de la Computación y, en concreto, en la Especificación Algebraica de sistemas de Programación Funcional.

Sirvan estas anécdotas de mi modesta trayectoria como investigador para ilustrar la conclusión que me gustaría que fuese extraída de todo lo precedente, más allá del tema específico tratado. La moraleja sería que cuando un tema es interesante, es esencialmente interdisciplinar (lo que he intentado hacer explícito introduciendo a lo largo del texto los *nombres* de las distintas disciplinas involucradas). Que no hay fronteras en el conocimiento y que el estudiante de matemáticas tiene que estar dispuesto a abandonar, cuando necesite enfrentarse a fondo con cualquier problema, la visión compartimentada que los planes de estudios le ofrecen. ¡Viva el mestizaje!

Bibliografía

[BB] G. Brassard, P. Bratley, *Fundamentos de Algoritmia*, Prentice Hall, 1997.

[G] P. Gaucher, *Homotopy invariants of higher dimensional categories and concurrence in Computer Science*, Math.Structures in Computer Sci.10, 481–524, 2000.

[Mau] C.R.F. Maunder, *Algebraic Topology*, Dover, 1996.

[May] J.P. May, *Simplicial Objects in Algebraic Topology*, Van Nostrand, 1967.

[Mu] J.R. Munkres, *Elements of Algebraic Topology*, Addison-Wesley, 1984.

[PS] F.P. Preparata, M.I. Shamos, *Computational Geometry*, Springer, 1985.

[RSS] J. Rubio, F. Sergeraert, Y. Siret, *EAT:Symbolic Software for Effective Homology Computation*, **ftp://fourier.ujf-grenoble.fr/pub/EAT**, Inst. Fourier Grenoble, 1997.

[Sc] D.S. Scott, *Logic and programming languages*, Comm. ACM 20, 634–641, 1977.

[Se] F. Sergeraert, *The computability problem in Algebraic Topology*, Advances in Maths 104, 1–29, 1994.

[SS] F. Sergeraert, Y. Siret, *Kenzo: Symbolic Software for Effective Homology Computation*, **ftp://fourier.ujf-grenoble.fr/pub/KENZO**, Inst. Fourier Grenoble, 1999.

[T] M.C. Tangora, *Computing the homology of the lambda algebra*, Memoirs of the AMS 337, 1985.