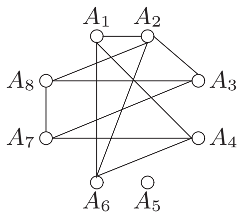
Capítulo 10

El lenguaje de los grafos

Los grafos, los objetos y el lenguaje cuyo estudio iniciamos aquí, son extremadamente útiles para representar primero, y luego analizar, problemas muy diversos. De una manera informal, un grafo es una colección de **vértices** y de **aristas** que unen estos vértices. Los vértices los dibujaremos como puntos (o pequeños círculos) sobre el plano; las aristas serán líneas que unen estos puntos. Adelantándonos a las definiciones formales, y para que el lector vaya comprobando la amplia capacidad de representación del lenguaje de los grafos, empezamos con algunos ejemplos.

Problema de horarios.

El plan de estudios de una cierta licenciatura universitaria consta de ocho asignaturas. Digamos que no deseamos programar a la misma hora dos asignaturas si hay algún alumno matriculado en ambas. Se desea minimizar el número de horas necesario para programar todas las asignaturas salvando esa dificultad.



Representamos primero cada asignatura con un vértice. Si entre dos asignaturas hay incompatibilidad, entonces dibujamos una arista entre los vértices correspondientes. La representación de toda esta información podría dar lugar a un grafo como el de la izquierda. Ahora se trata de asignar a cada vértice un símbolo (la hora a la que se imparte la asignatura) de manera que vértices que vayan unidos por una arista *no* lleven el mismo símbolo. Pero además nos interesará hacerlo empleando el mínimo número posible de símbolos. Para abordar esta cuestión, que quizás el lector haya relacionado ya con la de construir listas con restricciones (recuérdese las discusiones del capítulo 2), desarrollaremos, en la sección 11.3, el lenguaje del coloreado de grafos.

¿Cómo ordena Google los resultados de sus búsquedas?

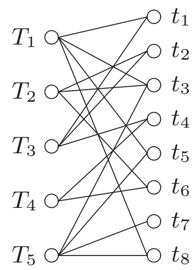
Los buscadores de la red tienen almacenada, en gigantescas bases de datos, información sobre los contenidos de millones de páginas *web*. Cuando se introduce un cierto término, el buscador encuentra las páginas en las que éste aparece. La cuestión que aquí nos interesa es la siguiente: ¿en qué orden se deben mostrar las páginas localizadas?

En los últimos años, Google se ha convertido en el buscador más eficaz, con mucha diferencia, hasta el punto de haberse convertido en un estándar. ¿Cuál es el secreto (o uno de los secretos) de Google? Al lector quizás le sorprenderá al descubrir que la explicación se basa, simplemente, en un poco de teoría de grafos y otro poquito de Álgebra lineal. No desvelamos nada más por el momento y remitimos al lector interesado a la subsección 10.3.

Asignación de tareas

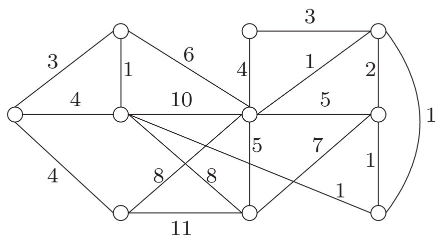
Formamos parte del Departamento de estajanovismo a destajo de un empresa. Contamos con un conjunto de trabajadores y otro de tareas y la información de para qué tareas está capacitado cada trabajador. Buscamos una asignación de tareas (a cada trabajador, una tarea distinta para la que esté preparado) de forma que consigamos ocupar al mayor número de trabajadores posible.

Digamos que en la empresa hay cinco trabajadores, $T_1, \dots, T_5$, y ocho tareas, $t_1, \dots, t_8$. El grafo va a tener 13 vértices en total, tantos como trabajadores y tareas, pero dibujamos, pues así resulta conveniente, los de los trabajadores a un lado, y los de las tareas a otro. Además, trazamos una arista entre un vértice que represente a un trabajador y otro que represente a una tarea si efectivamente el trabajador está preparado para realizar esa tarea (véase el dibujo de la derecha). Querríamos, por ejemplo, asignar a cada trabajador una tarea de entre las que sabe realizar. Determinar las condiciones necesarias para que esta asignación exista (y encontrar un algoritmo que la construya) es un problema clásico de la Combinatoria, que aquí abordaremos desde el punto de vista de la Teoría de grafos (véase la subsección 11.2).



Construcción de redes

Se busca conectar una serie de ciudades mediante un oleoducto. Un estudio de ingeniería previo nos permite conocer qué tramos se pueden construir, y quizás el coste de cada uno de ellos. Se trata de conectar todas las ciudades usando el menor número de tramos posible (o quizás con el menor coste posible).



Las ciudades serán los vértices y las conexiones, las aristas. En ocasiones, además, conviene asignar un número a cada arista (su "peso"), que indica el coste de construcción del tramo correspondiente (véase el dibujo de la izquierda). Lo que buscamos es unir todos los vértices del grafo utilizando el menor número posible de aristas. Si además consideramos los pesos, querremos que esa conexión sea lo más barata posible (véanse las subsecciones 10.2.2 y 11.4.1).

Obsérvese que en estos ejemplos aparecen términos como "minimizar el número de...", "conseguir la máxima asignación...", "determinar la red más barata que...". Y es que una de las aplicaciones fundamentales del lenguaje de los grafos consiste en 1) *representar* adecuadamente y de forma manejable la *información* de un cierto sistema; 2) para luego resolver *problemas de optimización*: realizar ciertas tareas de la "mejor" manera posible, atendiendo a un cierto criterio. Problemas de éstos hay muchos, además de los esbozados, y en las secciones 11.1 y 11.4 trataremos varios de ellos.

Por ejemplo, en el último grafo dibujado, podríamos decir que las aristas representan tramos de carretera entre ciudades y que los pesos son las longitudes respectivas (o el coste que supone recorrerlas). Si un viajante debe visitar clientes en cada ciudad, le interesará diseñar un recorrido que, empezando en un cierto vértice, visite todos los demás y vuelva al de partida. Y, quizás, de la manera más “barata” posible. O quizás queramos determinar simplemente cuál es el camino más corto entre una ciudad y otra, siguiendo las rutas establecidas.

Para tratar estas cuestiones, y otras más con las que nos iremos encontrando, conviene introducir cierto lenguaje, al que dedicaremos todo este capítulo. Un lenguaje en el que caben términos tan pictóricos como coloreado y número cromático, o tan botánicos como árboles y bosques. Este mismo capítulo contiene algunas aplicaciones de este lenguaje, mientras que unas cuantas más serán tratadas en el capítulo 11.

10.1. Las definiciones

Empecemos, como corresponde, con la definición matemática formal.

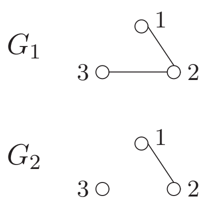
Definición 10.1 *Un **grafo** G es un conjunto no vacío V de **vértices** y un conjunto A de **aristas** extraído de la colección de subconjuntos de dos elementos de V. Una arista de G es, pues, un subconjunto $\{a, b\}$, con $a, b \in V$, $a \neq b$.*

Para las aristas hablamos de *pares* de vértices, así que dará lo mismo escribir $\{a, b\}$ que $\{b, a\}$ para referirse a la arista que, como diremos habitualmente, *une* o *conecta* los vértices a y b.

Para ser precisos, deberíamos decir que esta definición se refiere a los conocidos como **grafos simples** (en la subsección 10.1.1 presentaremos algunas generalizaciones de esta noción inicial de grafo). Mientras no se diga lo contrario, cuando hablemos de un grafo siempre estaremos refiriéndonos a este caso.

Diremos que dos grafos son **iguales** si tienen el mismo conjunto de vértices y la misma colección de aristas. Nombraremos un grafo G mediante $G = (V, A)$. A veces, cuando manejemos varios grafos a la vez, utilizaremos símbolos como $V(G)$ y $A(G)$ para recordar a qué grafo corresponden los conjuntos de vértices y aristas a los que nos estamos refiriendo.

Ejemplo 10.1.1 *Consideremos un conjunto de vértices $V = \{1, 2, 3\}$. Construyamos algunos grafos distintos con ese conjunto de vértices.*



Para determinar el grafo, dado que el conjunto de vértices está fijo, basta con dar el conjunto de aristas. Los subconjuntos de dos elementos de V son $\{1, 2\}$, $\{1, 3\}$ y $\{2, 3\}$. De ellos, deberemos escoger unos cuantos (o quizás ninguno) para formar el conjunto de aristas. Las elecciones $A_1 = \{\{1, 2\}, \{2, 3\}\}$ y $A_2 = \{\{1, 2\}\}$ dan lugar a los grafos $G_1 = (V, A_1)$ y $G_2 = (V, A_2)$ que dibujamos a la izquierda. ♣

En este ejemplo, podríamos haber construido hasta ocho grafos distintos con el conjunto de vértices $\{1, 2, 3\}$, pues hay tres “candidatos” a ser aristas y, por tanto, $2^3 = 8$ posibles elecciones de conjuntos de aristas (incluyendo el conjunto vacío). Si partiéramos de un conjunto de vértices que tuviera n elementos, por ejemplo $V = \{1, 2, \ldots, n\}$, entonces tendríamos

$$\#\big\{\text{subconjuntos de tamaño 2 extraídos de } \{1, \ldots, n\}\big\} = \binom{n}{2}$$

"candidatos" para ser aristas. Para formar el grafo, deberíamos escoger unas cuantas aristas de entre estos candidatos; así que el número de grafos distintos con vértices $\{1, 2, \dots, n\}$ es

$$\#\Big\{\text{subconjuntos extraídos de } \Big\{1, \dots, \binom{n}{2}\Big\}\ \Big\} = 2^{\binom{n}{2}}.$$

Un número gigantesco, en cuanto n sea grande. Por ejemplo, con siete vértices hay ya

$$2^{\binom{7}{2}} = 2^{21} = 2097152 \quad \text{grafos distintos.}$$

¡Más de dos millones de grafos a partir de siete puntos sobre el plano!, quién lo habría dicho (véase también el ejercicio 10.1.4). El número de grafos con vértices $\{1, \dots, n\}$ y k aristas (donde $k \leq \binom{n}{2}$) resulta ser

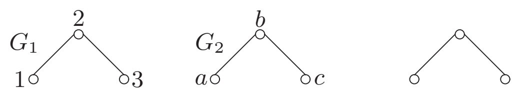
$$\binom{\binom{n}{2}}{k}.$$

Sirvan estos sencillos cálculos para avisar al lector de que muchas de las cuestiones por las que nos interesaremos en lo que sigue resultarán "difíciles" desde el punto de vista computacional.

Muchas veces representaremos los vértices con los símbolos $\{1, \dots, n\}$, o también con los símbolos $u, v, w \dots$, o quizás con $v_1, v_2, v_3 \dots$. Para las aristas utilizaremos los símbolos a (por arista) o quizás e (del inglés *edge*).

Pero esta cuestión de los "nombres" o "etiquetas" que utilizamos para marcar los vértices de un grafo es sutil. Por ejemplo, desde un punto de vista formal, los grafos $G_1 = (V_1, A_1)$, donde $V_1 = \{1, 2, 3\}$ y $A_1 = \{\{1, 2\}, \{2, 3\}\}$, y $G_2 = (V_2, A_2)$, con $V_2 = \{a, b, c\}$ y $A_1 = \{\{a, b\}, \{b, c\}\}$ apenas "tienen que ver" entre sí: no tienen el mismo conjunto de vértices, y tampoco, por tanto, de aristas.

Pero, salvo por un simple *cambio de etiquetas* en los vértices, nos atreveríamos a decir que ambos contienen la misma información (en cierto sentido que conviene precisar). Vea el lector los dibujos de la derecha y diga, a la vista de ellos, si no está de acuerdo. Más aún, reconozca que siente la tentación de quedarse, simplemente, con el esquema desnudo, sin etiquetas, que aparece más a la derecha. Confiamos en que la discusión de la subsección 10.1.2 arroje luz sobre esta cuestión.

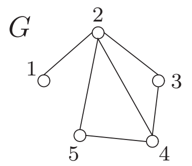


Subgrafos

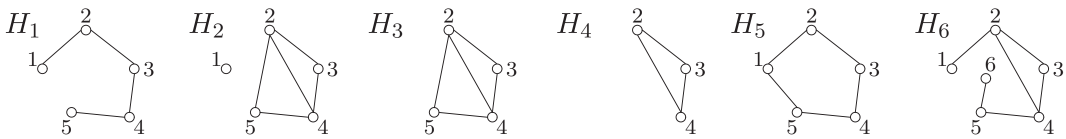
En muchas ocasiones, conviene considerar grafos que están incluidos "dentro" de otros. Dado un grafo $G = (V, A)$, formamos un **subgrafo** $H = (V', A')$ de G seleccionando primero algunos de los vértices de G (esto es, $V' \subseteq V$) para después, de las aristas que unieran vértices del conjunto V' en el grafo original G, quedarnos con algunas de ellas (o todas)[1].

Un caso especialmente relevante es aquél en el que el subgrafo incluye *todos* los vértices del grafo original. Un subgrafo H (de G) con $V(H) = V(G)$ se llamará un **subgrafo abarcador**. Algunos de los ejemplos que expusimos al principio del capítulo requieren hallar subgrafos abarcadores (en realidad, "árboles" abarcadores; esto lo veremos en la subsección 10.2.2).

[1] Si sólo exigiéramos que H contuviera algunos de los vértices y algunas de las aristas de G, podríamos llegar a situaciones sin sentido como incluir una arista pero no alguno de sus vértices (y no sería un verdadero grafo).



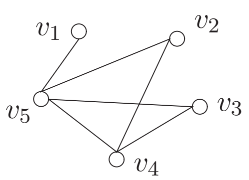
Partamos, por ejemplo, del grafo G que dibujamos a la izquierda. El conjunto de vértices es $V(G) = \{1, 2, 3, 4, 5\}$, y el de aristas $A(G) = \{\{1, 2\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{4, 5\}\}$. De los seis grafos que representamos debajo de estas líneas, sólo son subgrafos de G los cuatro primeros. Porque H_5 contiene una arista (la $\{1, 5\}$) que no estaba en el original, mientras que H_6 tiene un vértice (y alguna arista) de más.



Los grafos H_1 y H_2 son, además, subgrafos abarcadores de G, porque incluyen a todos los vértices de G. Los subgrafos H_3 y H_4 son también especiales: son los subgrafos que se obtienen considerando sólo los conjuntos de vértices $\{2, 3, 4, 5\}$ y $\{2, 3, 4\}$, respectivamente, y quedándonos con todas las aristas que unieran esos vértices en G. A estos últimos subgrafos nos referiremos como los **subgrafos inducidos** por esos conjuntos de vértices.

10.1.1. Representaciones matriciales

Una forma muy útil de representar la información de un grafo $G = (V, A)$ es mediante su **matriz de vecindades** (o **matriz de adyacencia**). Si el conjunto de vértices es $V = \{v_1, \ldots, v_n\}$ y le asignamos una ordenación arbitraria, por ejemplo $(v_1, \ldots, v_n)$, el grafo se puede describir mediante una matriz $n \times n$, cuyas filas y columnas van etiquetadas con los vértices, y cuyas entradas son: en la posición (v_i, v_j) pondremos un 1 si $\{v_i, v_j\} \in A$, y un 0 en caso contrario (obsérvese que la matriz depende de la ordenación de los vértices elegida). La matriz tendrá ceros en la diagonal y será simétrica: si en la posición (v_i, v_j) aparece un 1 es porque $\{v_i, v_j\} \in A$; por tanto, en la posición (v_j, v_i) deberá aparecer también un 1. Véase el ejemplo:



	v_1	v_2	v_3	v_4	v_5
v_1	0	0	0	0	1
v_2	0	0	0	1	1
v_3	0	0	0	1	1
v_4	0	1	1	0	1
v_5	1	1	1	1	0

Por ejemplo, el 1 que aparece en la fila tercera, columna cuarta (y el 1 de la correspondiente posición simétrica, fila cuarta, columna tercera) significa que el grafo contiene la arista entre los vértices v_3 y v_4. Así que un grafo G simple con n vértices es exactamente lo mismo (o mejor, contiene la misma información) que una matriz $n \times n$ simétrica de ceros y unos con ceros en la diagonal, pues a partir de la matriz podemos reconstruir fácilmente el grafo.

Obsérvese también que un grafo $G = (V, A)$ se corresponde con una relación definida en el conjunto V (recuérdese la la discusión sobre las relaciones de la sección 3.5). En el caso de los grafos simples, se tratará de una relación simétrica, desde luego no reflexiva y que, en algún caso, será además transitiva (véase la discusión más general al respecto en la subsección 10.3).

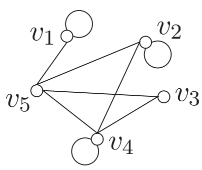
Esta representación matricial sugiere dos reflexiones. Por un lado, como veremos, el de los grafos es un lenguaje especialmente diseñado para un uso algorítmico, algoritmos que requerirán la intervención del ordenador. Y para un ordenador, por supuesto, un grafo no es un dibujo sobre el papel, sino una matriz de ceros y unos[2]. Así que, aunque habitualmente argumentaremos con las representaciones simbólicas del grafo (su conjunto de vértices y aristas, o los dibujos correspondientes), convendrá siempre tener en cuenta si los procedimientos y algoritmos que desarrollemos tienen sencillas implementaciones matriciales.

Por otro lado, una vez representado el grafo como una matriz, podemos hacer uso de nuestros conocimientos de Álgebra Lineal, que en ocasiones ayudan en el análisis de ciertos problemas. Obsérvese que la matriz de vecindades de un grafo no es cualquiera: es simétrica y sus entradas son números no negativos (ceros y unos, de hecho). Muchas de las características del grafo se reflejan en algunos invariantes de la matriz, en especial los autovalores y autovectores; todo un mundo, conocido como la *teoría espectral de grafos*, que aquí sólo visitaremos en contadas ocasiones (véanse, por ejemplo, la discusión de la página 759 y siguientes, y también el ejercicio 10.1.28).

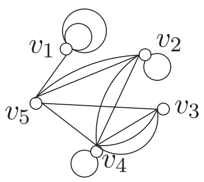
A. Grafos dirigidos, aristas múltiples, lazos y pesos

Algunas de las cuestiones que trataremos en este texto exigen generalizar el concepto de grafo. Las presentaremos de manera algo informal en este apartado, y desarrollaremos algunas de sus características cuando el problema que nos incumba nos exija utilizarlas.

Podríamos, por ejemplo, permitir **lazos**, esto es, aristas cuyos dos vértices sean el mismo. Un grafo simple con lazos contiene la misma información que una matriz cuadrada, simétrica, cuyas entradas son ceros y unos y que, quizás, tiene unos en la diagonal. A la derecha mostramos un ejemplo.



	v_1	v_2	v_3	v_4	v_5
v_1	1	0	0	0	1
v_2	0	1	0	1	1
v_3	0	0	0	1	1
v_4	0	1	1	1	1
v_5	1	1	1	1	0



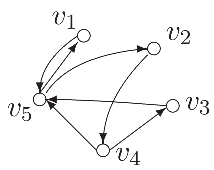
	v_1	v_2	v_3	v_4	v_5
v_1	2	0	0	0	1
v_2	0	1	0	2	2
v_3	0	0	0	3	1
v_4	0	2	3	1	1
v_5	2	1	1	1	0

Pero también podríamos permitir que hubiera aristas múltiples (pares de vértices conectados por dos o más aristas); en este caso hablaremos de **multigrafos**. A un multigrafo le corresponde una matriz simétrica cuyas entradas[3] son ceros o números naturales: en la diagonal irían ceros si no permitiéramos lazos. A la izquierda exhibimos un ejemplo, en el que se pueden apreciar aristas dobles y triples, y hasta lazos dobles.

[2]Existen otras maneras de representar la información de un grafo en el ordenador. Por ejemplo, en la demostración del lema 10.1 aparecerá una representación alternativa, conocida como la "matriz de incidencias".

[3]Esta representación matricial sólo tiene en cuenta *cuántas* aristas hay entre cada par de vértices, y en las aplicaciones que aquí trataremos esto es todo lo que necesitaremos. Pero es concebible que quisiéramos distinguir entre, por ejemplo, dos aristas que unan el mismo par de vértices (imaginemos que fueran dos carreteras distintas que unieran las mismas ciudades) ¿Cómo se le ocurre al lector que se podría representar matricialmente esta situación?

Otra posibilidad, que estudiaremos con detalle en la subsección 10.3, es la de dar *orientación* a las aristas. En los **grafos dirigidos** o **digrafos** (con lazos o sin ellos), las entradas de la matriz siguen siendo ceros y unos; pero, como se aprecia en el ejemplo que mostramos a la derecha, ésta ya no es necesariamente simétrica (por cierto, en la matriz leemos las aristas de filas a columnas; véase, por ejemplo, el 1 en la fila v_4 y columna v_3). Obsérvese que, en el dibujo, cada arco ha sido sustituido por una "flecha".



	v_1	v_2	v_3	v_4	v_5
v_1	0	0	0	0	1
v_2	0	0	0	1	0
v_3	0	0	0	0	1
v_4	0	0	1	0	1
v_5	1	1	0	0	0

La última generalización, que ya contemplábamos en algunos de los ejemplos que abrían este capítulo, consiste en asociar, en cualquiera de los esquemas anteriores (grafos simples, multigrafos o grafos dirigidos), a cada arista a del grafo un número real, su **peso**, $p(a)$, un cierto número que en cada problema tendrá un significado particular. En este caso, hablaremos de un **grafo con pesos** (o grafo ponderado[4]). Si, por ejemplo, tuviéramos un grafo simple con pesos, su matriz sería simétrica y sus entradas serían los pesos de cada arista. En la subsección 11.4.3 nos interesaremos en particular por los (multi)-**grafos dirigidos con pesos**, el caso más general posible, en el que las matrices correspondientes son, simplemente, matrices cuadradas.

Pero insistimos en que en lo sucesivo, mientras no se diga lo contrario, todos los grafos a los que nos referiremos serán simples.

B. Vértices vecinos, grado de los vértices y sucesión de grados

Un concepto fundamental en un grafo es el de grado de un vértice:

Definición 10.2 *Dado un grafo $G = (V, A)$, diremos que dos vértices $v, w \in V$ son **vecinos** si $\{v, w\} \in A$. El **grado de un vértice** es el número de vecinos que tiene en el grafo:*

$$\operatorname{grado}(v) = \#\{w \in V \,:\, \{v, w\} \in A(G)\}\,.$$

Si el grado de un vértice es 0, esto es, si el vértice no tiene vecinos en el grafo, diremos que es un vértice **aislado**. Para determinar el grado de un vértice *basta sumar los unos que aparezcan en su fila* de la matriz de vecindades[5]. La lista de los grados de los vértices de un grafo G es su **sucesión de grados**:

$$(\operatorname{grado}(v_1), \operatorname{grado}(v_2), \ldots, \operatorname{grado}(v_n))\,.$$

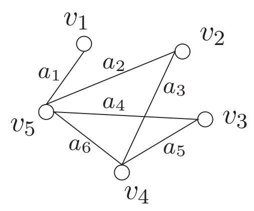
Por convenio, se suele escribir esta lista con los valores ordenados de menor a mayor. Pero no toda lista de n números ≥ 0 se corresponde con la sucesión de grados de un grafo con n vértices. Por ejemplo, se tiene la siguiente restricción que nos dice, en particular, que la suma de los grados de un grafo es siempre un número par.

[4]Dícese del grafo sobre el que se piensa mucho, y muy detenidamente.

[5]Un lazo, si los permitiéramos y los hubiere, contribuye con un 2 al grado del vértice. Como en la matriz de vecindades hemos representado los lazos con unos en la posición de la diagonal correspondiente, en este caso tenemos que la suma de las entradas de la fila (o columna) de un vértice v es el grado de v menos el número de lazos.

Lema 10.1 *En cualquier grafo $G = (V, A)$ se tiene que* $\sum_{v \in V} \text{grado}(v) = 2|A|$.

Demostración (véase también el *lema de los saludos*, ejemplo 2.1.4). Vamos a argumentar sobre otra manera de registrar la información contenida en el grafo, conocida como "matriz de incidencias": sus columnas están etiquetadas con los vértices $\{v_1, \ldots, v_n\}$, pero ahora, como etiquetas de las filas, situamos las aristas $\{a_1, \ldots, a_m\}$. En la posición (a_i, v_j) colocaremos un 1 si el vértice v_j es extremo de la arista a_i; y un 0 en caso contrario. Véase el ejemplo de la figura.



	v_1	v_2	v_3	v_4	v_5
a_1	1	0	0	0	1
a_2	0	1	0	0	1
a_3	0	1	0	1	0
a_4	0	0	1	0	1
a_5	0	0	1	1	0
a_6	0	0	0	1	1

En la fila etiquetada con la arista a_1 aparecerán sólo dos unos (sus dos extremos); lo mismo ocurre con el resto de las filas. Así que, sumando por filas, obtenemos $2|A|$. La columna correspondiente al vértice v_j contendrá tantos unos como vecinos tenga este vértice: su suma valdrá justamente $gr(v_j)$. Sumando los resultados de todas las columnas, y por doble conteo, obtenemos lo que buscábamos. Piense el lector si el resultado sigue siendo válido cuando permitimos lazos y/o aristas múltiples (véase el ejercicio 10.1.7; recomendamos también el ejercicio 10.1.8). ■

Finalizamos la subsección presentando un par de características importantes de un grafo:

Definición 10.3 *Llamaremos* ***mínimo grado*** *y* ***máximo grado*** *de un grafo a los números*

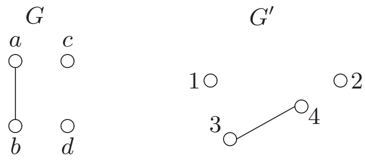
$$\delta(G) = \min_{v \in V(G)} \{\text{grado}(v)\} \qquad y \qquad \Delta(G) = \max_{v \in V(G)} \{\text{grado}(v)\}\,.$$

Si los dos números coinciden, por ejemplo en el valor k, entonces todos los vértices del grafo tendrán grado k, y hablaremos de un grafo ***k*-regular**.

10.1.2. Grafos en cuerpo y alma: isomorfía de grafos

Los grafos, como hemos visto, tienen como objeto representar relaciones entre los elementos de una colección de vértices. La verdadera naturaleza de esta relación variará según el problema considerado; por ejemplo, pueden ser personas que se conozcan o no, ciudades conectadas o no mediante carreteras, asignaturas con incompatibilidades, etc. Pero sea cual sea su carácter, nos gusta interpretar esta relación en términos genéricos de *vecindad*: así, "estar relacionados" lo entendamos como "ser vecinos" en el grafo.

Para que dos grafos sean iguales han de tener primero, los mismos vértices, exactamente, y después, las mismas aristas, exactamente. Observemos los dos grafos de la derecha. Tienen vértices distintos: los del de la izquierda son $\{a, b, c, d\}$ y los del de la derecha, $\{1, 2, 3, 4\}$, y cada uno tiene una única arista. Son grafos distintos, simplemente porque tienen vértices distintos.



Pero el lector condescendiente admitirá que, en realidad, los dos grafos se corresponden con la misma relación de vecindad, que, de hecho, podemos describir en palabras que no hacen referencia ni a los vértices concretos ni a las aristas concretas; a saber, se trata de una relación en un conjunto de 4 elementos en la que hay exactamente un par de ellos relacionados.

Como con los grafos queremos analizar y entender relaciones de vecindad, el lector convendrá con nosotros que resultará provechoso disponer de un concepto que capture el hecho de que esos dos grafos, en realidad, *dicen* lo mismo. Vamos, que son iguales aunque sean formalmente distintos, *sensu stricto*[6]. Ese concepto es el de **grafos isomorfos**. Isomorfo significa, filológicamente hablando, con la misma forma. Vamos pues con la definición matemática.

Definición 10.4 *Decimos que dos grafos $G = (V, A)$ y $G' = (V', A')$ son* ***isomorfos*** *si podemos encontrar una aplicación biyectiva entre los conjuntos de vértices, $\phi : V \to V'$, de manera que*

$$\{v, w\} \in A \quad \textit{si y solamente si} \quad \{\phi(v), \phi(w)\} \in A' .$$

La aplicación ϕ es un diccionario que traduce la información del primer grafo en la del segundo. Si dos vértices son vecinos en el primer grafo, entonces los vértices correspondientes en el segundo también son vecinos, y recíprocamente. Así queda exactamente trasladada la relación de vecindad: es, a todos los efectos, la misma. Obsérvese que el isomorfismo está definido por una biyección entre los vértices, pero que induce automáticamente otra entre las aristas: basta asociar a una arista con vértices v y w la arista con vértices $\phi(v)$ y $\phi(w)$.

La aplicación ϕ de $\{a, b, c, d\}$ en $\{1, 2, 3, 4\}$ definida por

$$\phi(a) = 3, \quad \phi(b) = 4, \quad \phi(c) = 1 \quad \text{y} \quad \phi(d) = 2$$

es un isomorfismo entre los grafos que considerábamos antes, pues justamente hace corresponder la única arista de G, la $\{a, b\}$, con la única arista de G', la $\{3, 4\}$.

Muchas de las nociones y propiedades de grafos que iremos introduciendo en las páginas que siguen serán **invariantes por isomorfía**. Es decir, que si un grafo G tiene una tal propiedad, los grafos isomorfos a él gozarán asimismo de esa propiedad. En cierto sentido, sólo las propiedades invariantes por isomorfía son importantes, porque sólo ellas refieren exclusivamente a lo esencial de la estructura de vecindad. Por ejemplo, el número de vértices y el número de aristas son invariantes por isomorfía. También la lista de grados de los vertices es invariante por isomorfía; de hecho, si ϕ define un isomorfismo entre G y G', entonces $\text{grado}(v) = \text{grado}(\phi(v))$ para cada vértice v de G.

Una letanía recurrente con la que el lector paciente se topará repetidamente a lo largo de este capítulo dedicado a los grafos, justo tras la introducción de casi todo concepto nuevo, reza: “comprobemos que la propiedad tal o cual es invariante por isomorfismo”. No suele resultar fácil decidir si dos grafos dados son isomorfos o no, y de hecho no se dispone de ningún procedimiento *efectivo y general* para ello. Para comprobar que dos grafos son isomorfos bastará con exhibir un isomorfismo. Encontrarlo puede no ser fácil, como hemos dicho más arriba, pero comprobar que el supuesto isomorfismo es tal es un asunto directo.

Por otro lado, para comprobar que dos grafos no son isomorfos habrá que argumentar que no existe ningún isomorfismo entre ellos, lo que no es tarea fácil. Piense en abstracto el aplicado lector: cuán difícil es demostrar —o convencer(se) de— que algo (por ejemplo, la isla de San Borondón) no existe. Pero, las matemáticas, como ciencia de lo abstracto –ésta es la clave– están cabalmente pertrechadas para este fin.

[6]Disculpe el amable lector el tono escolástico, y es que tanta disquisición quintaesencial. . .

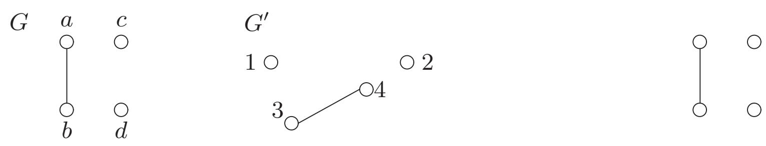
Para buscar un isomorfismo entre dos grafos (o, si es el caso, comprobar que no hay tal) con n vértices —si no tuvieran el mismo número de vértices ya estaría claro que no son isomorfos— podríamos analizar las $n!$ biyecciones del primer conjunto de vértices en el segundo, y comprobar una a una si lleva aristas en aristas. Pero $n!$ es, a poco que n no sea pequeño, un número desorbitadamente grande. Como iremos viendo en diversos ejemplos, las propiedades invariantes por isomorfía son una herramienta útil para decidir que dos grafos *no* son isomorfos. Pues si comprobamos que uno de los dos grafos en cuestión tiene una tal propiedad y el otro no, entonces no pueden ser isomorfos.

A. Isomorfía y matrices de vecindades

Las matrices de vecindades nos permiten visualizar la noción de isomorfía. Como la matriz de vecindades M de un grafo G tiene filas y columnas etiquetadas con los vértices de G, es fácil ver[7] que dos grafos G y G' son isomorfos si podemos encontrar una forma común de reordenar las filas y las columnas de la matriz M de G que nos dé la matriz M' de G'. La receta para esa reordenación común es simple: si el vértice v de G etiqueta, por ejemplo, la tercera fila (y la tercera columna) de M y el vértice $\phi(v)$ etiqueta, por ejemplo, la sexta fila (y la sexta columna) de M', entonces en la reordenación la fila 3 pasará a ser la fila 6 y, asimismo, la columna 3 pasará a ser la columna 6. Nótese el significado del término "reordenación común".

B. Almas de grafos

Lo común de los dos grafos que tratábamos antes queda recogido por la representación que aparece debajo de estas líneas, más a la derecha:



Este último es una suerte de grafo *desnudo*, sin etiquetas en los nodos que representan vértices, a la que llamaremos **alma del grafo**[8].

En las almas no hablamos de vértices sino de nodos, y no hablamos de aristas sino de conexiones entre nodos, simplemente para resaltar que las almas de los grafos son entes[9] de naturaleza distinta a la de los grafos en sí. Frecuentemente, usaremos almas de grafos y no los grafos en sí en discusiones generales en las que los nombres de los vértices no aportan información relevante.

Obsérvese que los grafos en sí guardan, respecto de sus almas, la misma relación que la matriz de vecindades (con sus filas y columnas etiquetadas) respecto de esa misma matriz pero sin etiquetas ni nombres en filas o columnas.

A un alma se le puede dar *cuerpo* de grafo asignándole nombres a los nodos para transformarlos en vértices. Nótese que dos grafos son isomorfos si tienen la misma alma, o recíprocamente, si se obtienen de la misma alma asignando nombres a los nodos del alma.

[7] A esto nos referíamos más arriba con visualizar.

[8] ¡Ah!, pero, ¿los grafos tienen alma? Y cuerpo, sí señor. Un alma es un grafo desnudo: metafísica escolástica.

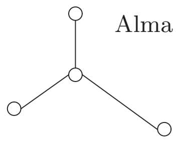
[9] ¿Entelequias? Esta discusión empieza a exigir, como música ambiental, canto gregoriano.

C. Almas en acción[10]

Las almas son muy útiles (como paso intermedio) cuando intentamos exhibir todos los grafos que cumplen determinadas características. Veamos aquí un ejemplo para ilustrar esta idea, aunque más adelante la usaremos con denuedo. Supongamos que queremos exhibir todos los grafos con los cuatro vértices $\{1,2,3,4\}$ que tienen 3 aristas. En total hay

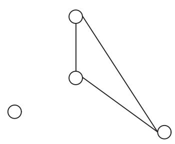
$$\binom{\binom{4}{2}}{3} = 20 \quad \text{de ellos.}$$

La suma de los grados de los vértices es el doble que el número de aristas, es decir, 6. Las posibles sucesiones de grados son cuatro números $g_1 \leq g_2 \leq g_3 \leq g_4$, todos ellos ≥ 0, de manera que $g_1 + g_2 + g_3 + g_4 = 6$. Toca ser metódico y analizar la consiguiente casuística. Vamos con ello.

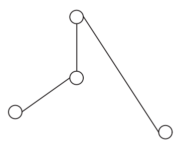


Alma número 1

Como hay 4 vértices, el máximo grado posible es 3. Pero si hay un vértice de grado 3, ya se usan las 3 aristas y los otros vértices habrán de ser de grado 1. La sucesión de grados es $(1,1,1,3)$, que se corresponde con el alma de la derecha.



Alma numero 2



Alma numero 3

Suponemos ya que no hay vértices de grado 3. Ha de haber al menos dos vértices de grado 2, porque si no la suma de los cuatro grados no llegaría a 6. Por tanto, restan dos posibilidades para la sucesión de grados: $(0,2,2,2)$ y $(1,1,2,2)$, que dan lugar a las dos almas que dibujamos a la derecha.

Puesto que las sucesiones de grados en estas almas son distintas, no hay posibles isomorfismos entre los grafos que tengan distintas almas, y, por tanto, a mayor abundamiento, grafos con distintas almas serán distintos. Por tanto, he ahí nuestro enfoque: para cada una de las tres almas buscaremos todos los grafos distintos que podemos tener con ese alma concreta, es decir, asignaremos nombres de entre $\{1,2,3,4\}$ a los nodos (para transformarlos en vértices). El alma es como una plantilla en la que al poner nombres damos cuerpo al grafo.

A primera vista, sin pensar, podíamos decir que, como hay cuatro posiciones en el alma y cuatro símbolos posibles, entonces cada alma da lugar a 4! grafos: cierto, pero no son todos distintos. Procedamos con cuidado.

Consideremos el alma numero 1. Observará el lector atento que aquí basta con decidir cuál es el nombre del nodo de grado 3. De manera que sólo hay 4 grafos distintos con vértices de $\{1,2,3,4\}$ con este alma. Por ejemplo, si decidimos que el nodo de grado 3 recibe el nombre de 2, entonces el grafo tiene aristas: $A = \{\{2,1\},\{2,3\},\{2,4\}\}$.

Vamos con el alma 2. De nuevo, hay sólo 4 grafos distintos: basta con decidir qué nombre le damos al nodo de grado 0, pues los otros tres vértices tienen todos los pares posibles. Si, por ejemplo, el nodo de grado 0 recibe el nombre de 2, entonces el grafo tiene aristas: $A = \{\{1,3\},\{3,4\},\{4,1\}\}$.

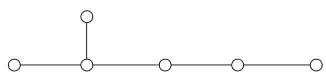
[10] ¿Una ONG?

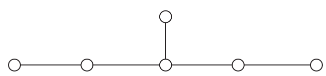
Finalmente, analicemos el alma 3, que es sin duda la que más trabajo da de las tres. Elegimos nombre para uno de los nodos de grado 1, después nombre para el vecino de grado 2, y luego, por último, de los dos nombres restantes elegimos el que asignamos al nodo de grado 1, y el grafo queda determinado. Este proceso se puede llevar a cabo de $4 \times 3 \times 2 = 24$ maneras. Por ejemplo, si en este proceso elegimos, por orden, el nombre "2" para el nodo de grado 1, el nombre "4" para el nodo vecino de grado 2 y el nombre "1" para el otro nodo de grado 1, tendremos las aristas $A = \{\{2,4\},\{4,3\},\{3,1\}\}$.

Pero —tenía que haber un pero— hemos impuesto un orden espurio en los nodos de grado 1 y no hay tal. Por ejemplo, si en este proceso elegimos por orden el nombre "1" para un nodo de grado 1, el "3" para el nodo vecino de grado 2 y el "2" para el otro nodo de grado 1, obtenemos las mismas aristas. Como consecuencia del orden impuesto en los nodos de grado 1, cada grafo aparece 2 veces. De manera que tenemos que dividir por 2, para obtener finalmente que con este alma tenemos $24/2 = 12$ grafos distintos.

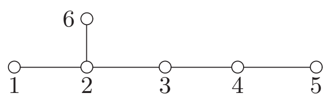
Reuniendo las consideraciones sobre las 3 almas, vemos que tenemos un total de $20 = 4+4+12$ grafos distintos con vértices $\{1,2,3,4\}$ y 3 aristas. Los que tienen alma 1 son todos isomorfos entre sí, y lo mismo ocurre para los que tienen alma 2 o los que tienen alma 3. De manera que distintos hay 20, y no isomorfos sólo tenemos 3, tantos como almas.

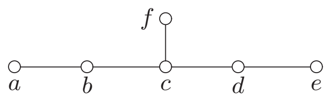
No quisiéramos que el ejemplo anterior pueda inducir al lector a creer que la sucesión de grados determina el alma de un grafo. Si así fuera tendríamos un discriminador de isomorfía. Pero no, ¡qué pena! Las dos almas que siguen tiene la misma sucesión de grados, a saber, $(1,1,1,2,2,3)$ pero no son la misma.





Para argumentar por qué estos dos grafos (almas) no son isomorfos conviene darles cuerpo nombrando los vértices; usaremos números para el de la izquierda y letras para el de la derecha[11]:





Si hubiera un isomorfismo ϕ entre los conjuntos de vértices $\{1,2,3,4,5,6\}$ y $\{a,b,c,d,e,f\}$, los respectivos vértices de grado tres deberían corresponderse, así que necesariamente tendríamos que $\phi(2) = c$. El vértice 3 tiene grado dos y es vecino del vértice 2, así que su imagen $\phi(3)$ ha de tener grado dos y ser vecino de $\phi(2) = c$. Hay dos posibilidades para $\phi(3)$, puede ser b ó d. Supongamos que fuera b (el argumento si fuera d es análogo, como comprobará el lector), esto es, que $\phi(3) = b$. Seguimos. Nos fijamos ahora en el vértice 4, que tiene grado dos y es vecino del vértice 3. Así que $\phi(4)$ ha de ser un vértice de grado dos, vecino de $\phi(3) = b$, pero, ¡b no tiene vecinos de grado dos!. Concluimos que no hay tal isomorfismo ϕ.

Más expeditivamente, para ver que estos grafos no son isomorfos basta con observar que el grafo de la izquierda tiene un par de vértices de grado 2 que son vecinos entre sí (propiedad invariante por isomorfismos), pero el de la derecha no.

[11]No atribuya el suspicaz lector, por favor, connotación política o cultural alguna a esta asignación. Nada más lejos de nuestra intención.

Puede sorprender lo sencillo que resulta contar el número de grafos que hay con n vértices y k aristas ($\binom{\binom{n}{2}}{k}$ de ellos), en comparación con lo laborioso que resulta clasificarlos (en función de las distintas almas) y luego contar los posibles etiquetados. ¿Es sencillo contar cuántos grafos *no isomorfos* con n vértices y k aristas hay? No lo es. Allá por la subsección 18.3.2 encontraremos un método para enumerar grafos no isomorfos. Pero, ¡atención!, para ello necesitaremos aplicar la combinación de funciones generatrices y teoría de grupos que es conocida como Teoría de Pólya. ¡Alta tecnología!

D. Sobre cuerpos y almas

El meticuloso lector, quizás abrumado por tanta disquisición de tintes escolásticos sobre cuerpos y almas de grafos, echará en falta una definición formal de alma de un grafo.

Parece, y así es en realidad, a qué negarlo, que hemos argumentado con la representación pictórica de un grafo como puntos (vértices) etiquetados en el plano y arcos entre ellos. Que el alma no es sino ese mismo dibujo, en el que eliminamos las etiquetas de los vértices (para que queden en simples nodos). Y que dar cuerpo no es sino la operación de nombrar nodos.

La definición formal casi puede sonar a una argumentación *ad divinitatem*:

Definición 10.5 *El alma de un grafo es su clase de equivalencia por isomorfía.*

Es decir, el alma de un grafo es la colección completa de grafos que son isomorfos a él. El lector interesado debería comprobar que la relación de isomorfía entre grafos define una relación de equivalencia en la colección de todos los grafos[12].

La críptica y económica frase de que, por ejemplo, los grafos que se exhiben a continuación son todos los grafos con 3 vértices, *salvo isomorfismos*, quiere decir que son un sistema de representantes de las clases de isomorfía de los grafos con 3 vértices. Es decir, que no son isomorfos y que cualquier grafo con 3 vértices es isomorfo a uno de ellos. O, finalmente (nótese que, en realidad, se trata de almas de grafos), que cualquier grafo de 3 vértices se obtiene dando cuerpo (asignando nombre a los nodos) a uno (y sólo uno) de ellos.

Por cierto, a las almas de grafos se les suele denominar (en definición negativa) **grafos no etiquetados**, y a la operación de dar cuerpo, **etiquetar**.

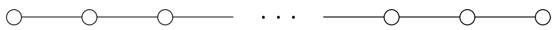
Nuestra última precisión en este contexto concierne a las matrices de vencindades. Las almas y los grafos no etiquetados con n vértices (nodos) pueden también entenderse como clases de equivalencia en el conjunto $\mathcal{M}$ de matrices cuadradas $n \times n$ simétricas, con entradas que son ceros y unos, y con ceros en la diagonal. En esta relación de equivalencia (el lector comprobará sin dificultad que es tal), dos matrices de $\mathcal{M}$ se dicen equivalentes si una se obtiene de la otra aplicando una misma permutación a las filas y a las columnas.

[12] Como ya sabemos que la isomorfía entre grafos requiere tener un mismo número de vértices, el lector que se pudiera sentir (lógicamente) cohibido ante la inmensidad de la colección de *todos* los grafos concebibles quizás prefiera restringirse a los de n vértices, de los que "sólo" hay $2^{\binom{n}{2}}$, y comprobar que en ese conjunto de grafos la isomorfía es relación de equivalencia.

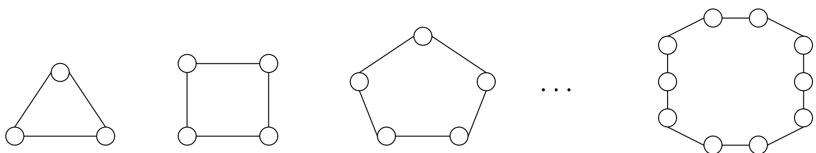
10.1.3. Algunas familias de grafos

Muchas de las cuestiones que trataremos en estas páginas se pueden describir con algunos grafos particulares, que enumeramos y describimos someramente a continuación. En general hablaremos de grafos salvo isomorfía, por lo que no haremos explícitas las etiquetas de los vértices. Nos estaremos refiriendo, pues, a almas de grafos[13].

Diremos que un grafo es un L_n, un **grafo lineal** con n vértices ($n \geq 2$) si tiene n vértices, de los que dos son de grado 1 y el resto, si los hay, de grado 2. Por lo tanto, es isomorfo a:

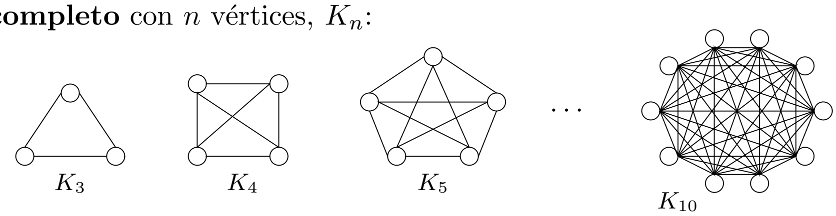


Otra familia relevante de grafos la conforman los llamados **grafos circulares** C_n, que tienen n vértices, $n \geq 3$, todos de grado 2 (véase una caracterización en el ejercicio 10.1.10):



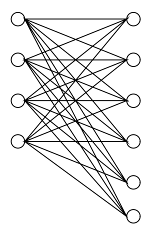
C_3 C_4 C_5 C_{10}

Si un grafo con n vértices tiene todas las $\binom{n}{2}$ posibles aristas, diremos que estamos ante el **grafo completo** con n vértices, K_n:



K_3 K_4 K_5 K_{10}

En el otro extremo encontramos los **grafos vacíos** N_n, con n vértices y ninguna arista (son los *grafos complementarios* de los completos, véase el ejercicio 10.1.6).

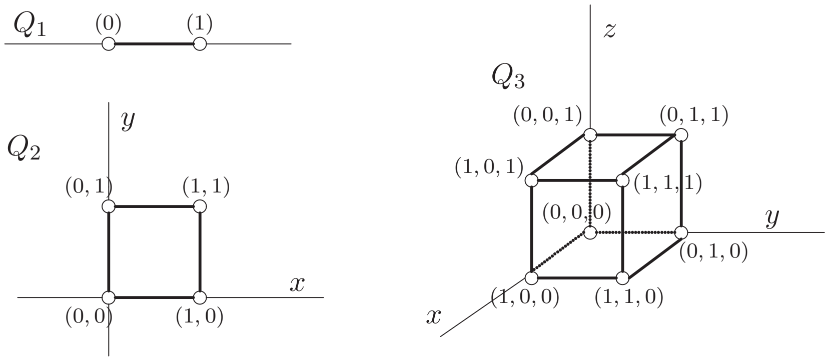


Una clase de grafos que tienen relevancia en diversos problemas (por ejemplo, en los problemas de asignación de tareas) son los llamados **grafos bipartitos**. Se trata de aquéllos en los que podemos partir el conjunto de vértices en dos clases, de manera que no haya aristas entre vértices de la misma clase. Un caso particular son los **grafos bipartitos completos**, que nombraremos como $K_{r,s}$. En el dibujo de la derecha aparece un $K_{4,6}$. Un grafo $K_{r,s}$ consta de $r+s$ vértices, divididos en dos clases, una con r vértices, y la otra con s; e incluye las $r \times s$ aristas que van de los vértices de un tipo a los del otro. Un grafo bipartito con r vértices de un tipo y s de otro se puede obtener del $K_{r,s}$ escogiendo un subconjunto de las aristas (véase una caracterización de los grafos bipartitos en el ejercicio 10.1.11).

La definición del grafo del **cubo**, Q_n, es abstracta: sus vértices van etiquetados con las listas de longitud n que podemos formar con ceros y unos. En total, pues, tiene 2^n vértices. Dos vértices de Q_n serán vecinos si las listas de ceros y unos que los identifican difieren en una *única* posición.

[13] Almas gemelas, podríamos decir, que en lugar de vagar eternamente se reúnen, por afinidades, en familias.

Mostramos, a continuación, los dibujos en los retículos de una, dos y tres dimensiones, que nos ayudan a entender su estructura (y justifican su nombre). Obsérvese que Q_1 es isomorfo a L_2, y Q_2, a C_4. Quizás el lector quiera intentar dibujar el grafo Q_4.

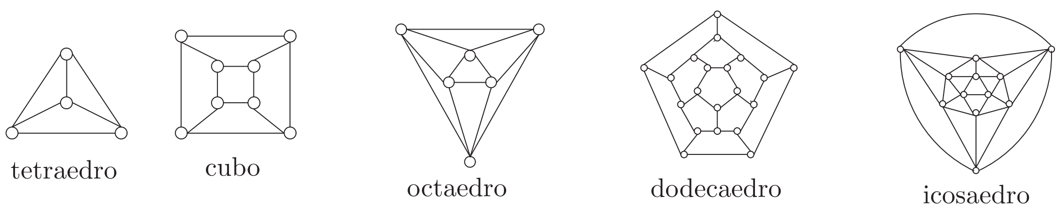


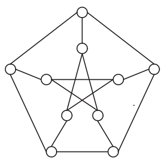
Como hay n distintas maneras de variar una posición en una n-lista, todos los vértices de Q_n tienen grado n (por lo que Q_n es un grafo n-regular). De lo que deducimos que el número de aristas de un Q_n es

$$2\,|A(Q_n)| = \sum_{v\in V(Q_n)} gr(v) = \sum_{j=1}^{2^n} n = n\,2^n \quad\Longrightarrow\quad |A(Q_n)| = n\,2^{n-1}\,.$$

Pese a que los dibujos esbozados no hagan sospecharlo, los grafos Q_n son bipartitos. Veamos: la mitad de sus vértices están etiquetados con n-listas con número par de ceros y la otra mitad, con un número impar. Pero dos listas que tienen un número par de ceros no pueden ser vecinas en este grafo (lo mismo ocurre para las impares).

El grafo del cubo ("tridimensional") Q_3 es uno de los cinco grafos conocidos como **grafos platónicos**, que están asociados a los conocidos y casi mágicos sólidos platónicos, y que aparecerán en la discusión sobre planaridad de la sección 11.5. Debajo de estas líneas puede contemplar el lector su elegante aspecto.

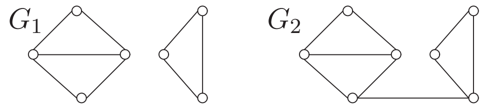




Existen otros muchos grafos interesantes que no pertenecen a ninguna de las familias anteriores: algunos son parte de familias más exóticas, otros son casi huérfanos... En el dibujo de la izquierda, por ejemplo, mostramos uno famoso, el llamado **grafo de Petersen**[14], que consta de 10 vértices, todos ellos de grado 3.

[14] Bautizado así en honor de su "diseñador", el matemático danés Julius Peter Christian Petersen (1839–1910). Para dar una idea al lector de su ubicuidad, señalemos que ilustra las portadas de dos de las más afamadas revistas de investigación en cuestiones relacionadas con la Teoría de grafos, como son *Graph Theory* y *Discrete Mathematics*.

10.1.4. De paseo por un grafo. Conexión



Si el lector observa con detenimiento los dos grafos que dibujamos a la derecha, comprobará que la diferencia fundamental entre ellos es que en el grafo G_2 las aristas del grafo nos permiten "llegar" de un vértice a cualquier otro: todos sus vértices están "conectados". Algo que no ocurre en G_1. Formalicemos esta noción, tan natural por otra parte, de "conexión" en grafos.

A. Paseos

Para empezar, necesitamos precisar en qué consiste la acción de "moverse" por un grafo: se trata de ir de vértice a vértice, pero *siempre* siguiendo las aristas del grafo.

Definición 10.6 *Un* ***paseo*** *en un grafo* $G = (V, A)$ *es una sucesión finita de vértices*

$$x_0, x_1, x_2, \ldots, x_l \qquad \text{(pueden repetirse vértices)}$$

de forma que $\{x_i, x_{i+1}\}$ *es una arista de* G*, para cada* $i = 0, 1, \ldots, l-1$.

Sitúese el lector en un vértice del grafo: ya tiene x_0. Decida entonces, a la vista de las aristas que inciden en x_0, a qué vértice puede trasladarse, escoja uno de ellos y ya tendrá x_1 (podría ser el mismo vértice, si es que hubiera un lazo y lo eligiéramos). Ahora observe detenidamente las aristas que parten de x_1 y vuelva a elegir (¿es posible que le haya dado por volver al vértice anterior?, quizás). Cuando haya decidido terminar este entretenido recorrido, le rogamos que anote los vértices que haya ido visitando (en riguroso orden). Ya tendrá un paseo.

El primer vértice de un paseo será el **vértice inicial** y el último, claro, el **vértice final**. Diremos que el paseo **conecta** estos dos vértices. Si ambos extremos coinciden, esto es, si el paseo comienza y termina en el mismo vértice, hablaremos de un paseo **cerrado**. Obsérvese que podríamos haber ido registrando también las aristas utilizadas, y describir nuestro paseo de la forma

$$x_0, (a_1), x_1, (a_2), x_2, (a_3), \ldots, x_{l-1}, (a_l), x_l$$

donde cada arista a_j une los vértices x_{j-1} y x_j; aunque esta lista tiene la misma información que la anterior[15]. La **longitud** del paseo es el *número de aristas* que hayamos utilizado (l, en cualquiera de las listas anteriores). Nótese que no nos referimos al número de aristas distintas utilizadas, pues éstas se pueden repetir, sino al número de "pasos" por arista que efectuamos durante el paseo. Por comodidad, aceptaremos como paseo uno que tenga longitud 0, un "paseo vacío", en el que no se sale del vértice inicial.

En argumento posteriores resultará útil la siguiente observación, que el lector podrá comprobar sin dificultad: si dos vértices se pueden conectar en un grafo con n vértices, entonces existe un paseo entre ellos de longitud a lo sumo $n - 1$.

En ocasiones conviene precisar esta noción general de paseo pues, por ejemplo, podrían interesarnos paseos que no repitieran vértices, o quizás que no repitieran aristas, etc. De esto hablaremos en el apartado D, pero ahora nos damos un paseíto hasta la siguiente idea.

[15]No así en el caso de un multigrafo, pues entre dos vértices puede haber más de una arista, y quizás sea necesario registrar cuál se usa realmente.

B. Conexión y componentes conexas

Definición 10.7 *Un grafo $G = (V, A)$ se dice* ***conexo*** *si, dados cualesquiera dos vértices $v, w \in V$, existe un paseo que los conecta.*

Por convenio[16], diremos que un grafo con un único vértice es también conexo (estará conectado consigo mismo por un "paseo vacío", de longitud 0). Alarmará al lector una definición que incluye un "dados cualesquiera dos vértices…" Porque, ¿*cómo* se comprueba que un grafo particular es conexo? En los ejemplos habituales de los libros de texto, la conexión (o no) de un grafo suele ser evidente a simple vista. Pero, ¿y si nos las tenemos con un grafo con 1000 vértices? Habría que comprobar la existencia de paseos para las $\binom{1000}{2}$ parejas posibles. Afortunadamente, hay métodos eficaces para determinar si un grafo es conexo o no (por ejemplo, el método matricial que veremos en el apartado C o los algoritmos de la sección 10.2.2).

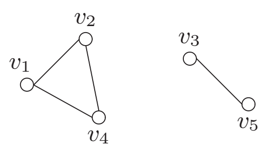
Si el grafo *no* es conexo, si tenemos vértices que no pueden ser conectados por paseos, nuestra intuición nos sugiere que el grafo estará formado por diversos "bloques" de vértices, cada uno de los cuales será un grafo conexo. La idea es muy natural, pero la definición formal requiere cierto lenguaje, que confiamos no "desconecte" al lector. Vamos con ella.

En el conjunto de los vértices de G definimos la siguiente relación $\mathcal{R}$: $u\mathcal{R}v$ si y sólo si u es "conectable" en G con v (esto es, existe un paseo en G que conecta u con v). Vamos a demostrar que esta relación es de equivalencia.

Para la reflexividad, no hay nada que probar. Para la simetría, observemos que si dos vértices (distintos) u y v cumplen que $u\mathcal{R}v$, será porque existe un paseo conectando u con v. Ese mismo paseo, "leído" al revés, conecta v con u, así que $v\mathcal{R}u$. Para la transitividad, digamos que tres vértices u, v, w, con $u \neq v$ y $v \neq w$, cumplen que $u\mathcal{R}v$ y $v\mathcal{R}w$. Como $u\mathcal{R}v$, hay un paseo que conecta u con v. Y como $v\mathcal{R}w$, habrá otro conectando v con w. "Uniendo" ambos, es decir, siguiendo el paseo de u a v y luego el de v a w, hallamos un paseo que conecta u con w. Así que $u\mathcal{R}w$.

Como la relación es de equivalencia, sabemos que el conjunto de vértices de G se parte en unas clases de equivalencia. Las **componentes conexas**, o simplemente componentes, de G son[17] los subgrafos inducidos por cada uno de estos conjuntos de vértices (esto es, dados los vértices de una clase de equivalencia, el subgrafo formado por esos vértices y todas las aristas que los unieran en G).

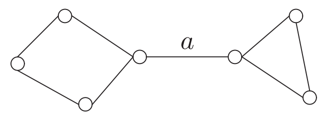
Las componentes conexas (que podrían constar de un único vértice, si es que éste era aislado) son grafos conexos. El grafo G es la unión de sus componentes conexas. El lector podrá comprobar que la matriz de vecindades de un grafo con varias componentes conexas se puede escribir, quizás permutando adecuada y simultáneamente las filas y las columnas, como una matriz por *cajas* (en el sentido habitual del Álgebra lineal). Véanse, en el ejemplo de grafo con dos componentes conexas que mostramos a la derecha, las correspondientes cajas con ceros.



	v_1	v_2	v_4	v_3	v_5
v_1	0	1	1	0	0
v_2	1	0	1	0	0
v_4	1	1	0	0	0
v_3	0	0	0	0	1
v_5	0	0	0	1	0

[16]Más bien por comodidad, para que todo cuadre; poderoso estímulo a la hora de diseñar definiciones.

[17]Una "definición" más visual: imagine que cada vértice es un *botón* sobre la mesa y cada arista un hilo que une dos botones. Tome uno de ellos y levántelo: todo lo que cuelgue de él será una componente conexa.



En ocasiones, como en la figura de la derecha, un grafo conexo deja de serlo al quitarle una arista particular (la arista a, en la figura). Reservamos para estas aristas "especiales" (que desempeñarán un papel relevante en argumentos relacionados con conexión y en algoritmos cuyo objetivo sea recorrer todo el grafo) un nombre especial: una arista a de un grafo G es un **puente** si el grafo $G \setminus \{a\}$ que se obtiene de G al quitar la arista a (y dejar los mismos vértices) tiene más componentes conexas que G. El lector puede comprobar que, en realidad, tendrá exactamente una componente conexa más que G. Así que si G es conexo y a es una arista puente, $G \setminus \{a\}$ consta de exactamente *dos* componentes conexas (véase el ejercicio 10.1.19).

Un grafo es conexo si podemos unir todos los vértices entre sí mediante paseos. Lo que sugiere que el grafo ha de contener un número "suficiente grande" de aristas. Pinte el lector n vértices e intente luego ubicar aristas, lo más económicamente posible, de manera que el grafo resultante sea conexo. ¿Cuántas ha usado? Al menos $n-1$, seguro. El siguiente resultado certifica esta intuición (véase también la generalización del ejercicio 10.1.15):

Proposición 10.2 *Si G es un grafo conexo, entonces $|A(G)| \geq |V(G)| - 1$.*

DEMOSTRACIÓN (por inducción en $|A|$, el número de aristas). Si $|A| = 0$, esto es, si no tenemos aristas, para que el grafo sea conexo, sólo puede haber un vértice. Si $|A| = 1$ y el grafo ha de ser conexo, sólo cabe la posibilidad de que sea el grafo L_2, que tiene dos vértices.

Supongamos cierto que si tenemos un grafo conexo con k aristas, para cualquier $k \leq m$, entonces $|V| \leq k+1$. Consideremos entonces un grafo conexo G con $|A(G)| = m+1$ y una arista a de G cualquiera. Llamemos H al grafo que se obtiene de G quitando la arista a: $H = G \setminus \{a\}$. Tiene los mismos vértices y una arista menos que G. Caben dos posibilidades:

(1) Si H sigue siendo conexo (es decir, si a no era arista puente en G), por hipótesis de inducción (tiene m aristas), como $|A(H)| = |A(G)| - 1$ y $V(G) = V(H)$, tendremos que

$$|A(H)| \geq |V(H)| - 1 \quad \Longrightarrow \quad |A(G)| \geq |V(G)|\,.$$

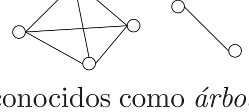
(2) Pero si a era puente en G, H ya no es conexo, sino que tiene dos componentes conexas; llamémoslas H_1 y H_2. Ambas son grafos conexos y tienen menos aristas que G (fijémonos en que estos subgrafos podrían constar de un único vértice). Teniendo en cuenta que

$$|A(H_1)| + |A(H_2)| = |A(H)| = |A(G)| - 1 \quad \text{y} \quad |V(H_1)| + |V(H_2)| = |V(H)|,$$

y con la hipótesis de inducción, terminamos la demostración:

$$\left.\begin{array}{l} |A(H_1)| \geq |V(H_1)| - 1 \\ |A(H_2)| \geq |V(H_2)| - 1 \end{array}\right\} \Longrightarrow |A(G)| - 1 \geq |V(G)| - 2 \Rightarrow |A(G)| \geq |V(G)| - 1\,.$$

■



Obsérvese que un grafo cuyo número de aristas cumpla que $|A| \geq |V| - 1$ no *tiene por qué ser conexo*. Véase a la derecha el dibujo de un grafo con 7 aristas y 6 vértices y que no es conexo. Los grafos conexos en los que se alcanza la igualdad $|A| = |V| - 1$, son conocidos como *árboles*, por razones pictóricas que quedarán claras más adelante; les dedicaremos atención especial en la sección 10.2.

C. Número de paseos, matriz de vecindades y conexión

Llamemos m_{ij} a las entradas de la matriz M de vecindades de un grafo G con vértices (ya ordenados) $(v_1, \dots, v_n)$. Los números m_{ii} son 0 para cada $i = 1, \dots, n$. Y si $i \neq j$, m_{ij} será 1 si v_i es vecino de v_j y 0 en caso contrario. Aunque por ahora no parezca muy interesante, podemos afirmar que cada m_{ij} "cuenta" el número de paseos de longitud 1 que hay entre el vértice v_i y el vértice v_j. Lo interesante es que esta interpretación se puede generalizar.

Teorema 10.3 *Si M es la matriz de vecindades de un grafo G, la entrada (i, j) de la matriz $M^l = M \times \overset{l \text{ veces}}{\cdots} \times M$, que denotamos como $m_{ij}^{(l)}$, cuenta el número de paseos de longitud l entre los vértices v_i y v_j.*

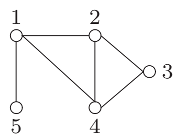
Demostración (por inducción en l). El caso $l = 1$ ha sido mencionado hace un momento. Consideremos todos los paseos de longitud l entre v_i y v_j cuyo penúltimo vértice visitado es v_k. Por hipótesis de inducción, $m_{ik}^{(l-1)}$ cuenta el número de paseos de longitud $l - 1$ entre v_i y v_k. Por otro lado, $m_{kj} = 0$ si es que v_k no es vecino de v_j y es un 1 en caso contrario. Así que, aplicando la regla del producto, hay $m_{ik}^{(l-1)} m_{kj}$ paseos de longitud l entre v_i y v_j cuyo penúltimo vértice visitado es v_k. Por la regla de la suma, el número total de caminos de longitud l entre v_i y v_j es

$$\sum_{k=1}^{n} m_{ik}^{(l-1)} m_{kj} \,.$$

Para completar la demostración, sólo queda observar que, como $M^l = M^{l-1} M$, las reglas de multiplicación de matrices nos dicen que $m_{ij}^{(l)}$ coincide con la suma anterior. ■

Así que, si queremos calcular el número de paseos de longitud l que hay entre dos vértices v_i y v_j, basta con calcular la potencia correspondiente M^l de la matriz de vecindades y consultar su entrada $m_{ij}^{(l)}$. Lo vemos en un ejemplo sencillo:

Ejemplo 10.1.2 *Consideremos el grafo G siguiente:*



La matriz de vecindades del grafo G, que denotamos por M, aparece debajo de estas líneas, junto con sus primeras potencias:

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad M^2 = \begin{pmatrix} 3 & 1 & 2 & 1 & 0 \\ 1 & 3 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 3 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad \text{y} \quad M^3 = \begin{pmatrix} 2 & 6 & 2 & 6 & 3 \\ 6 & 4 & 5 & 5 & 1 \\ 2 & 5 & 2 & 5 & 2 \\ 6 & 5 & 5 & 4 & 1 \\ 3 & 1 & 2 & 1 & 0 \end{pmatrix}.$$

Obsérvese, por cierto, que en la diagonal de M^2 aparecen los grados de cada vértice (éste es un hecho general, véase el ejercicio 10.1.23).

De la matriz M^3 deducimos, por ejemplo, que hay seis paseos distintos de longitud tres entre los vértices 1 y 2. Verifique el lector, sobre el dibujo del grafo, que son los siguientes: $(1, 5, 1, 2)$, $(1, 4, 1, 2)$, $(1, 2, 1, 2)$, $(1, 2, 3, 2)$, $(1, 2, 4, 2)$ y $(1, 4, 3, 2)$. ♣

El teorema 10.3 nos proporciona una manera de comprobar si un grafo G con n vértices y matriz de vecindades M es conexo o no. Ya sabemos que si dos vértices se pueden conectar en un grafo con n vértices, es seguro que lo podrán hacer utilizando un paseo cuya longitud sea a lo sumo $n-1$. Como la entrada (i, j) de la matriz

$$\widetilde{M} = I_n + M + M^2 + \cdots + M^{n-1}$$

(donde I_n significa la matriz identidad $n \times n$) contiene la información sobre el número de paseos de longitud a lo sumo $n-1$ que existen entre los vértices v_i y v_j, hemos probado que

Teorema 10.4 *Si alguna entrada de la matriz*

$$\widetilde{M} = I_n + M + M^2 + \cdots + M^{n-1}$$

es nula, entonces el grafo no puede ser conexo. Y viceversa, si todas las entradas de $\widetilde{M}$ *son positivas, entonces el grafo será conexo.*

Más aún, si el grafo no es conexo, la matriz $\widetilde{M}$ también contiene la información sobre las componentes conexas. Tomemos, por ejemplo, la primera fila de la matriz, etiquetada con el vértice v_1: las posiciones que no contengan ceros determinan los vértices de la componente conexa a la que pertenece v_1. El procedimiento se repetiría para la fila correspondiente al primer vértice no incluido en la componente anterior. Y así sucesivamente, hasta determinar todas las componentes conexas del grafo.

Ahora bien, este método exige calcular las $n-1$ primeras potencias de la matriz M, tarea en general muy costosa si n es grande. En la sección 10.2.2 revisaremos métodos más eficaces para determinar si un grafo es conexo o no.

D. Paseos eficientes

Muchos de los algoritmos y procedimientos que veremos en las páginas siguientes buscan maneras de recorrer las aristas o los vértices de un grafo de manera "económica".

Si los vértices del grafo representan ciudades y las aristas, carreteras entre ellas, nos puede interesar encontrar la manera de conectar dos de ellas utilizando el menor número de aristas posible. Esto supone considerar paseos en los que no se repitan aristas. O, más aún, en los que ni siquiera se repitan vértices (lo que, en particular, impide que se repitan aristas).

Aunque a veces, al tiempo que pretendemos ser económicos, también seremos ambiciosos, y nos interesará recorrer todas las aristas, o quizás visitar todos los vértices.

Estos diversos objetivos dan lugar, en la literatura de grafos, a una amplia nomenclatura para describir distintos tipos de paseos adaptados a cada situación particular: aquéllos en los que no se repiten aristas, en los que no se repiten vértices, que visitan todos los vértices, etc.

Es difícil encontrar los nombres adecuados para cada una de estas situaciones. Candidatos hay, y así los encontrará el lector en otros textos: caminos, caminos simples, circuitos, ciclos, sendas, etc., aunque no en todos con el mismo significado[18].

[18]Tampoco en la literatura anglosajona hay unanimidad, y el lector podrá encontrar términos como *walk*, *path*, *trail*, *circuit*, *cycle*, etc. (o *chemin*, *boucle*, *cheminement*, etc., en la literatura francesa) con diversos significados.

Así que, en alarde de continencia, no nos dejaremos llevar por la tentación taxonómica, y en lugar de bautizar cada tipo de paseo con un nombre especial, algo que despista, se olvida y obliga a repasar las definiciones en cada ocasión, optaremos[19] por utilizar circunloquios como "paseos que no repiten vértices", "paseos que no repiten aristas", "paseos que visitan todos los vértices sin repetir aristas", etc. Confiamos en que el lector sepa disculpar estos excesos perifrásticos. Aunque hay términos de uso tan frecuente y establecido que no podemos renunciar a ellos. Por ejemplo, el siguiente:

Definición 10.8 *Un **ciclo** en un grafo G es un paseo*

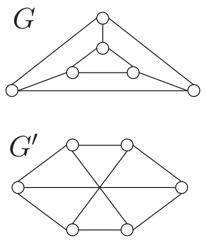
$$x_0, x_1, \ldots, x_l \qquad \textit{(con } l \geq 3\textit{)}$$

*donde $x_0 = x_l$ (esto es, el paseo es cerrado) y tal que los vértices son todos distintos (excepto el primero y el último, claro). La **longitud** del ciclo $x_0, x_1, \ldots, x_l$ (con $x_0 = x_l$) es justamente l.*

Recalcamos que un ciclo contiene al menos tres vértices. Además, si tenemos un paseo cerrado con al menos tres vértices, siempre podemos extraer de él un ciclo (véase el ejercicio 10.1.13). Obsérvese también que decir que un grafo G contiene un ciclo de longitud l es lo mismo que afirmar que contiene un grafo tipo C_l como subgrafo.

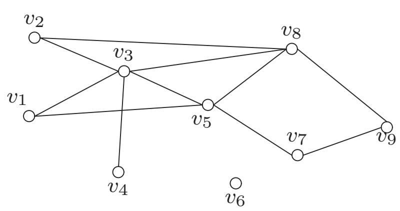
Si dos grafos G y G' son isomorfos mediante una biyección entre vértices ϕ y, por ejemplo, G contiene un ciclo $C = (v_0, \ldots, v_l)$, entonces $C' = (\phi(v_0), \ldots, \phi(v_l))$ ha de ser también un ciclo en G'. Esto nos conduce a otro importante invariante de un grafo:

Definición 10.9 *Si G es un grafo, se llama **cuello** del grafo G al mínimo de las longitudes de los ciclos de G. Si G no contiene ciclos, entonces convenimos que* $\text{cuello}(G) = +\infty$.



El valor del cuello de un grafo, un número entero ≥ 3 (o quizás $+\infty$), es un invariante (si G y G' son ϕ-isomorfos, entonces $\text{cuello}(G) = \text{cuello}(G')$) que, en ocasiones, permitir decidir si dos grafos son isomorfos o no. Véanse, por ejemplo, los que dibujamos a la izquierda. Tienen el mismo número de vértices y de aristas y todos los vértices son de grado 3. Parece difícil determinar si son isomorfos o no, sin recurrir a la comprobación de las $6! = 720$ posibles biyecciones. Pero fijémonos en que $\text{cuello}(G) = 3$, mientras que $\text{cuello}(G') = 4$. Así que no pueden ser isomorfos.

E. Distancias en un grafo (conexo)



Consideremos el grafo que dibujamos a la izquierda, que representa las posibles rutas existentes entre las ciudades $v_1, \ldots, v_9$. Un conductor tiene que elegir una ruta que conecte las ciudades v_1 y v_9. Hay muchos paseos que conectan los dos vértices. Pero nuestro conductor, que no tiene espíritu peripatético, desea hallar la ruta más corta, en términos del número de aristas que debe recorrer.

[19] Con dos salvedades: en la sección 11.1 elevaremos la categoría de ciertos paseos honrándolos con los ilustres nombres de Euler y Hamilton (lo que no es mala distinción).

Por simple inspección del grafo, concluimos que el paseo más corto emplea exactamente *tres* aristas. No hay un único "paseo más corto": por ejemplo, los paseos (v_1, v_3, v_8, v_9) y (v_1, v_5, v_7, v_9) recorren, ambos, tres aristas. Pero aunque disponga de varias posibilidades, y sea cual sea el paseo corto elegido, nuestro conductor asignará sin duda un número, el 3, a la "distancia" entre los vértices v_1 y v_9.

Lo que sigue es la definición formal de "distancia" en un grafo.

Definición 10.10 *En un grafo G conexo*[20]*, definimos la* ***distancia*** *entre dos vértices cualesquiera $u, v \in V(G)$ como*

$$d_G(v, w) = \min_{\gamma}\{\text{longitud}(\gamma)\}\,,$$

donde el mínimo se calcula sobre todos los paseos γ que conectan los vértices u y v en G. Obsérvese que $d_G(v, v) = 0$, para cada vértice $v \in V(G)$.

Consideremos un grafo conexo $G = (V, A)$ cualquiera. Tomemos cada par de vértices suyos y calculemos, con el procedimiento anterior, la distancia entre ellos. Cuando lo hayamos hecho para todos los pares de vértices, tendremos definida una función de $V \times V$ (los pares de vértices) en el conjunto de los enteros no negativos.

Esta función cumple unas determinadas propiedades (por ejemplo, la desigualdad triangular) que hacen que podamos referirnos a ella como una *distancia*, en el sentido matemático del término. Dejamos al lector que se ejercite con la comprobación (véase el ejercicio 10.1.25).

Una vez que lo haya hecho, y ésta es la gran ventaja de los conceptos abstractos en Matemáticas, estará "autorizado" a establecer analogías mentales entre esta noción de distancia en un grafo y otras sobre las que tendrá intuiciones más arraigadas, como la distancia euclídea en el plano. Y a trasladar a este contexto nociones como la de **geodésicas** entre u a v (paseos entre u y v cuya longitud[21] coincida con $d_G(u, v)$). O a construir su propia geometría con bolas[22], por analogía con las bolas euclídeas.

El lector de inquieta curiosidad puede ya entretenerse con el análisis de distancias en un caso particular especialmente relevante, como es el del grafo Q_n del cubo n-dimensional (cuyos vértices son las n-listas de ceros y unos). De su estudio, y de sus aplicaciones a la codificación, nos ocuparemos ampliamente en la subsección 16.3.4.

Obsérvese que la noción de "paseo más corto" que aquí hemos considerado se refiere a un grafo (conexo) sin pesos. En un modelo de la red de carreteras más realista, sería conveniente asociar a cada arista un *peso* que reflejara, por ejemplo, la distancia en kilómetros que separa sus dos extremos. El objetivo del conductor sería ahora establecer una ruta cuya longitud total (en kilómetros) fuera lo más corta posible. Podría ocurrir que el paseo "más corto", en este nuevo sentido, no fuera necesariamente el que menos aristas empleara. Sobre esta generalización reflexionaremos en la subsección 11.4.2.

[20] Aunque hemos limitado la definición a grafos conexos, siempre podríamos decidir que la distancia entre dos vértices no conectables en un grafo (es decir, que están en distintas componentes conexas) es $+\infty$. Obsérvese también que esta definición permite redefinir el concepto de vecindad en un grafo: dos vértices u y v de un grafo G serán vecinos si y sólo si $d_G(u, v) = 1$.

[21] En principio, puede haber varias geodésicas entre un par de vértices, y sólo en ciertas circunstancias la geodésica será única (véase, por ejemplo, el ejercicio 10.1.27).

[22] La bola de radio r sería el subgrafo formado por los vértices a distancia $\leq r$.

EJERCICIOS DE LA SECCIÓN 10.1

10.1.1 *Compruébese que se pueden formar hasta* $2^{n(n+1)/2}$ *grafos distintos, simples y con lazos, con los vértices* $\{v_1, \dots, v_n\}$

10.1.2 *Dados los vértices* $\{v_1, v_2, \dots, v_n\}$,

(a) ¿cuántos grafos simples distintos con m *aristas se pueden formar? ¿Y si permitimos lazos?*

(b) ¿Cuántos grafos dirigidos distintos con m *aristas se pueden formar? ¿Y si permitimos lazos?*

(c) Para todas las cuestiones de los dos apartados anteriores, ¿cuál es la respuesta si la condición es que los grafos contengan no más de m *aristas?*

10.1.3 *(a) ¿Cuántos multigrafos distintos con exactamente* m *aristas se pueden formar con* n *vértices? (b) ¿Y si permitimos (multi)lazos?*

10.1.4 *Una tarde de verano, alguien nos propone completar la siguiente (apasionante) tarea: dibujar todos los grafos (simples) con ocho vértices que hay. Para ello nos proporciona "innumerables hojas", en cada una de las cuales están ya pintados los vértices de un octógono regular. Tras dibujar unos cuantos, como buenos matemáticos que somos, nos mosqueamos, nos paramos e intentamos calcular cuánto tiempo nos llevará la tarea. Para ello, hacemos la siguiente estimación: nos lleva, digamos, un segundo, trazar una arista. Ahora, en media, ¿cuántas aristas tiene un gafo con ocho vértices? Completa el argumento y estima el tiempo necesario para concluir la tarea.*

10.1.5 *Dado un grafo* $G = (V, A)$, *el número*

$$\text{grado}(G) = \frac{1}{|V|} \sum_{v \in V} \text{grado}(v)$$

representa el ***grado medio*** *de los vértices del grafo* G. *Este número estará entre el grado mínimo y el máximo,*

$$\delta(G) \leq \text{grado}(G) \leq \Delta(G)\,.$$

El número grado(G) *es una medida del número de aristas que hay en el grafo* G *por cada vértice. Compruébese que*

$$\text{grado}(G) = \frac{1}{2}\,\frac{|A(G)|}{|V(G)|}\,.$$

10.1.6 *Sea* $G = (V, A)$ *un grafo. El* ***grafo complementario*** $\widehat{G} = (\widehat{V}, \widehat{A})$ *de* G *es el grafo cuyos vértices son los de* V *y cuyas aristas unen pares de vértices que no están unidas en* G. *Si* G *tiene* n *vértices, de grados* $g_1, g_2, \dots, g_n$, *¿qué grados tienen los vértices de* $\widehat{G}$*? ¿Cuánto vale la suma de los grados de los vértices de* $\widehat{G}$*?*

10.1.7 *Compruébese si en un multigrafo con lazos* G *se cumple que la suma de los grados coincide con (dos veces) el número de aristas.*

10.1.8 *Sea* G *un grafo con al menos dos vértices. (a) Compruébese que hay un número par (o cero) de vértices con grado impar. (b) Verifíquese que en* G *hay, al menos, dos vértices con el mismo grado.*

10.1.9 *Constrúyanse cinco grafos con* 8 *vértices, todos de grado* 3, *de forma que cada dos de esos grafos no sean isomorfos.*

10.1.10 *Pruébese que* C_n *es el único grafo conexo (salvo isomorfismos) con* n *vértices de forma que el grado de todos sus vértices es 2.*

10.1.11 *Demuéstrese que un grafo G es bipartito si y sólo si no contiene ciclos de longitud impar.*

10.1.12 *(a) ¿Cuántos grafos de tres vértices pueden construirse de manera que cada dos no sean isomorfos? (b) ¿Y cuántos con cuatro? (c) ¿Y con cinco?*

10.1.13 *Sea $(x_0, x_1, \dots, x_l)$, $l \geq 3$, un paseo cerrado (esto es, $x_0 = x_l$) en un grafo G. Compruébese que podemos extraer un ciclo de ese paseo.*

10.1.14 *Fijemos los vértices $\{1, \dots, n\}$ y sea G el grafo completo con esos vértices.*

(a) *¿Cuántos grafos isomorfos a un C_3 distintos se pueden formar que sean subgrafos de G?*

(b) *¿Cuántos grafos isomorfos a un C_k distintos se pueden formar que sean subgrafos de G?*

(c) *¿Cuántos grafos isomorfos a un K_r se pueden formar con los vértices $\{1, \dots, n\}$? Obsérvese que el apartado (a) es un caso particular de (b) y (c).*

(d) *La misma pregunta, pero para un $K_{r,s}$.*

10.1.15 *Sea $G = (V, A)$ un grafo con p componentes conexas. Compruébese que*

$$|V| - p \leq |A| \leq \binom{|V| - p + 1}{2}.$$

10.1.16 *Sea un grafo G y sean $\delta = \delta(G)$ y $c = \operatorname{cuello}(G)$ el grado mínimo y el cuello de G, respectivamente ($c(G) < +\infty$).*

(a) *Supongamos que c es un número impar, digamos $c = 2m + 1$. Pruébese que G tiene, al menos,*

$$1 + \delta + \delta(\delta - 1) + \delta(\delta - 1)^2 + \cdots + \delta(\delta - 1)^{m-1} \quad \textit{vértices.}$$

Así que, en particular, $|V(G)| \geq (\delta - 1)^{\frac{c-1}{2}}$.

(b) *Supongamos que c es un número par, $c = 2m$. Entonces, el número de vértices de G es, al menos, $2 + 2(\delta - 1) + 2(\delta - 1)^2 + \cdots + 2(\delta - 1)^{m-1}$.*

(c) *Compruébese que se tiene siempre que $|V(G)| \geq (\delta - 1)^{\left[\frac{c-1}{2}\right]}$. Obsérvese que estos resultados nos indican que si un grafo tiene δ grande (todos los vértices tienen muchos vecinos) y c grande (todos los ciclos son largos), entonces ha de tener muchos vértices[23]. Por ejemplo, si $\delta = 10$ y $c = 10$, entonces el grafo tiene al menos 132860 vértices.*

d) *De un cierto grafo sabemos que no tiene triángulos (C_3) ni cuadrados (C_4) y que tiene menos de 100 vértices. ¿Podemos deducir la existencia de al menos un vértice con grado "pequeño"?*

10.1.17 *Demuéstrese que si G es un grafo con n vértices y al menos $(k-1)n - \binom{k}{2} + 1$ aristas, donde $0 < k < n$, entonces hay un subgrafo de G con $\delta(H) \geq k$.*

10.1.18 *Sea G un grafo con $V(G) \subseteq \{1, \dots, n\}$ con $|V(G)| = v$ y $|A(G)| = a$. ¿Cuántos grafos distintos con vértices en $\{1, \dots, n\}$ contienen a G como subgrafo?*

10.1.19 *Compruébese que si G es un grafo conexo y a es una arista puente de G, entonces $G \setminus \{a\}$ tiene exactamente dos componentes conexas.*

10.1.20 *Sea G un grafo con n vértices y dos componentes conexas. (a) ¿Cuál es el número mínimo de aristas que G puede tener en esas condiciones? (b) Supongamos además que cada componente de G es un grafo completo. ¿Cuál es el número mínimo de aristas que G puede tener?*

[23] Sin embargo, esto ocurre cuando exigimos que ambos números sean altos simultáneamente: un grafo como el K_{11} tiene $\delta = 10$, y sólo tiene 11 vértices. Mientras que un C_{10} tiene $c = 10$ (y 10 vértices).

10.1.21 *Sea $G = (V, A)$ un grafo con n vértices. Demuéstrese que si* $\text{grado}(v) \geq (n-1)/2$ *para todo vértice $v \in V$, entonces la distancia entre cualesquiera dos vértices es ≤ 2. En particular, G es conexo.*

10.1.22 *En una reunión de 20 personas hay en total 48 pares de personas que se conocen.*

(a) Justifíquese por qué hay al menos una persona que a lo sumo conoce a cuatro personas.

(b) Supongamos que hay exactamente una persona X que conoce a lo sumo a cuatro; y supongamos que esta X conoce al menos a una. Verifíquese que las otras 19 conocen exactamente a cinco cada una. ¿A cuántos conoce X?

10.1.23 *Sea G un grafo con vértices $\{v_1, \dots, v_n\}$ y sea M la matriz de vecindades correspondiente. Llamemos $a_{i,j}$ a los registros de M y $a_{i,j}^{(2)}$ a los de la matriz M^2. Compruébese que $a_{i,i}^{(2)} = \text{grado}(v_i)$ para cada $i = 1, \dots, n$.*

10.1.24 *Sea G un grafo con matriz de vecindades M. Demuéstrese que G contiene un K_3 si y sólo si para algún par (i, j), las entradas (i, j) de las matrices M y M^2 son no nulas.*

10.1.25 *Sea $G = (V, A)$ un grafo conexo. Pruébese que la función distancia en G, $d_G : V \times V \to \mathbb{R}$, descrita en la Definición 10.10 es una distancia, en el sentido matemático del término. Es decir, que verifica las siguientes propiedades:*

1. *Para cada par de vértices $u, v \in V$, $d_G(u, v) \geq 0$. Además, $d_G(u, v) = 0$ si y sólo si $u = v$.*
2. *(Simetría): $d_G(u, v) = d_G(v, u)$ para cada par $u, v \in V$.*
3. *(Desigualdad triangular): $d_G(u, v) \leq d_G(u, w) + d_G(w, v)$ para cualesquiera $u, v, w \in V$.*

10.1.26 *Sea G un grafo conexo con matriz de vecindades M correspondiente a la ordenación de los vértices $(v_1, v_2, \dots, v_n)$. Llamemos $a_{i,j}^{(k)}$ a la entrada (i, j) de la matriz M^k. Compruébese que*

$$d_G(v_i, v_j) = \text{mín}\{k \geq 1 \,:\, a_{i,j}^{(k)} \neq 0\},.$$

10.1.27 *A los paseos entre u y v cuya longitud coincida con $d_G(u, v)$ se les llama* ***geodésicas*** *entre u y v. Puede haber varias geodésicas distintas entre dos vértices. Comprúebese, sin embargo, que si G es un grafo conexo con cuello $c(G)$ y $d(u, v) \leq c(G)/2$, entonces hay una única geodésica de u a v.*

10.1.28 *Calcúlense los autovalores (con sus multiplicidades) de las matrices correspondientes a los grafos K_n, C_n, $K_{r,s}$ y Q_n.*

10.1.29 *Algunos resultados básicos sobre los autovalores de un grafo. Sea $G = (V, A)$ y sea M su matriz de vecindades asociada. Sus autovalores (que son reales, pues la matriz es simétrica) son los números $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, ya ordenados de mayor a menor. Sean $\Delta(G)$ y $\delta(G)$, respectivamente, el máximo y el mínimo grado en el grafo G. El lector con especiales inclinaciones hacia el Álgebra lineal podrá demostrar los siguientes resultados:*

a) $\lambda_n < 0 < \lambda_1$;

b) $|\lambda_j| \leq \Delta(G)$, *para cada* $j = 1, \dots, n$.

c) $\delta(G) \leq \lambda_1 \leq \Delta(G)$.

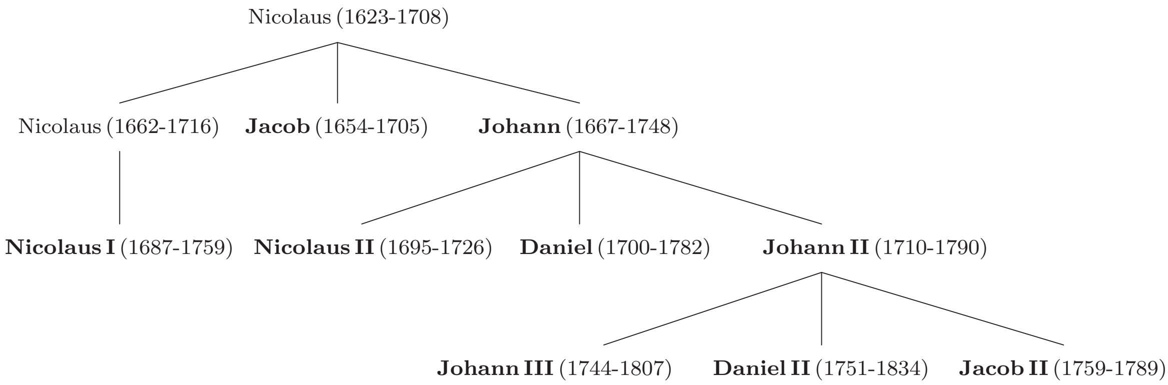
d) $\Delta(G)$ *es autovalor de G si y sólo si G es regular.*

e) Si $-\Delta(G)$ es autovalor de G, entonces G es regular y bipartito.

e) Si G es bipartito y μ es autovalor de G, entonces $-\mu$ también lo es (con la misma multiplicidad).

10.2. Árboles

La de los Bernoulli de Basilea es, quizás, la familia más famosa de la historia de las Matemáticas.



Famosa por la cantidad de excelentes matemáticos que "produjo" (hasta nueve, los que aparecen en negrita en el esquema anterior, en tres generaciones distintas) y, también, por la especial personalidad de algunos de ellos. De algunos de los más destacados ya hemos hablado en capítulos anteriores (Daniel en la página 468, Jacob en la página **??**; de Johann hablaremos en la página 919). A la derecha los vemos a los tres, coronando el precioso cartel anunciador del Primer Congreso Internacional de Matemáticos[24], celebrado en Zurich, en 1897. Por debajo de ellos (¡aunque sólo en el diseño del cartel!), y completando el Olimpo de matemáticos suizos, encontramos a Euler y a Steiner (de quien hablaremos en el capítulo 19). Pero no son estas consideraciones históricas las que aquí nos interesan, sino, precisamente, la manera en que hemos exhibido arriba la información sobre la familia Bernoulli, su *árbol genealógico*. Es un grafo cuyos vértices van etiquetados con los nombres de los componentes de la familia, que tiene una estructura muy particular.

Es muy habitual utilizar estas estructuras jerárquicas para representar información. El lector las encontrará por doquier: en el diseño y representación de los sucesivos pasos de un algoritmo, en la descripción que del contenido de un ordenador muestra el Explorador de Windows, etc. En esta sección estudiaremos las peculiaridades del tipo especial de grafos,

[24]Los matemáticos se reúnen cada cuatro años en un congreso en el que se exponen los últimos avances de las diversas disciplinas que integran el edificio de las Matemáticas. En cada congreso se anuncian los ganadores de las medallas Fields, el equivalente de los Nobel para las Matemáticas (con permiso de los recientemente creados Premios Abel). El Congreso Internacional de Matemáticos del 2006 se celebró en Madrid.

que llamaremos *árboles*, que nos permiten codificar cuestiones tan diversas. De sus variadas aplicaciones (problemas de optimización, diseño de algoritmos, análisis de juegos, etc.) nos ocuparemos en la sección 11.4.

A. Definición de árbol y caracterizaciones

La primera definición de la noción de árbol (de las varias que daremos) es la sugerida por el "aspecto" del grafo genealógico de la página anterior:

Definición 10.11 *Un **árbol** es un grafo **conexo** y **sin ciclos**.*

En el mismo tono botánico, se define un **bosque** como un grafo sin ciclos (si es conexo, será un árbol; si no lo es, sus componentes conexas serán árboles). Por ejemplo, los grafos lineales L_n son árboles, mientras que los circulares C_n o los completos K_n no lo son en cuanto $n \geq 3$. Los bipartitos completos $K_{r,s}$, que son siempre conexos, sólo son árboles si $s = 1$ ó $r = 1$ (puesto que, si $r \geq 2$ y $s \geq 2$, hay al menos un ciclo de orden cuatro).

Obsérvese que ser conexo exige tener "bastantes" aristas, para así poder conectar todos los vértices, mientras que no tener ciclos supone, en principio, que haya "pocas", para que no se formen ciclos. Los árboles están justo en el punto de equilibrio.

El siguiente resultado formaliza esta idea. Tiene dos apartados. El primero de ellos nos dice que los árboles son los conexos "más económicos", en el sentido de que tienen el número mínimo de aristas que permiten la conexión; en otros términos, un árbol es un grafo minimalmente conectado. El segundo apartado nos dice que los árboles son los más "eficientes" en cuanto a no tener ciclos; en otros términos, un árbol es un grafo maximalmente sin ciclos.

Proposición 10.5 *Sea G un grafo.*

i) El grafo G es un árbol si y solamente si es conexo y tiene la propiedad de que al eliminar una arista cualquiera el grafo deja de ser conexo.

ii) El grafo G es un árbol si y solamente si no tiene ciclos y, si le añadiéramos una arista cualquiera, se formaría un ciclo.

DEMOSTRACIÓN. i) Supongamos primero que tenemos un grafo G conexo y sin ciclos. Queremos probar que se desconecta al quitar una arista cualquiera.

Sea a una arista de G y formemos el grafo $G \setminus \{a\}$ eliminándola. Si $G \setminus \{a\}$ fuera conexo, podríamos conectar en $G \setminus \{a\}$ los vértices de la arista a. Pero añadiendo la arista a, se formaría un ciclo en G (contradicción). Así que $G \setminus \{a\}$ es no conexo (sea cual sea la arista a de G que elijamos).

En el otro sentido, supongamos que G es un grafo conexo que se desconecta si quitamos cualquier arista. Si el grafo contuviera un ciclo, siempre podríamos quitar una arista de ese hipotético ciclo sin que el grafo dejara de ser conexo, lo que supone una contradicción. Luego ese tal ciclo no puede existir[25].

[25] Como sabemos, véase el ejercicio 10.1.19, si quitamos una arista de un grafo conexo y éste se desconecta, lo hace en exactamente dos componentes conexas. En el caso de un árbol, al quitar una arista cualquiera se formará un bosque con dos componentes conexas. Obsérvese que lo que nos dice esta proposición es que toda arista de un árbol es un puente.

ii) Supongamos primero que G es un grafo conexo y sin ciclos, es decir, un árbol. Consideremos dos vértices cualesquiera que no sean vecinos en G. Por estar en un grafo conexo, existirá un paseo que los conecte en G. Al añadir una arista entre los vértices, tendremos un paseo cerrado (con al menos tres vértices), del que podremos extraer un ciclo.

En el otro sentido, sea G un grafo sin ciclos para el que añadir una arista cualquiera supone la formación de un ciclo. Supongamos que no fuera conexo. En este caso, al menos existirían dos vértices que no podríamos conectar en G. Pero entonces todavía podríamos añadir la arista que los une sin que se nos formara un ciclo, de nuevo una contradicción. ∎

Sabemos (recuérdese la proposición 10.2) que, en un grafo conexo, $|A(G)| \geq |V(G)| - 1$. La igualdad se alcanza justamente para los árboles, como nos dice el siguiente resultado:

Proposición 10.6 *Un grafo G es un árbol si y solamente si es conexo y $|A(G)| = |V(G)| - 1$.*

Demostración. En un sentido, vamos a proceder por inducción sobre el número de aristas $|A|$: la inducción es en sentido fuerte, es decir, la hipótesis de inducción para d aristas es que el resultado es cierto para grafos con un número de aristas $\leq d$.

- Si G es un árbol con una arista, $|A(G)| = 1$, sólo cabe la posibilidad de que sea un L_2, para el que $|V(G)| = 2$.
- Supongamos cierto que para todo árbol con $|A(G)| \leq d$ se tiene que $|A(G)| = |V(G)| - 1$. Consideremos un árbol cualquiera G con $|A(G)| = d + 1$. Si nos fijamos en una arista a de G, sabemos que
$$G \setminus \{a\} = G_1 \cup G_2,$$
donde G_1 y G_2 son árboles con no más de d aristas (pues $|A(G_1)| + |A(G_2)| = d$). A estos dos árboles les podemos aplicar la hipótesis de inducción para deducir lo que buscábamos. En efecto, por inducción
$$|A(G_1)| = |V(G_1)| - 1 \qquad \text{y} \qquad |A(G_2)| = |V(G_2)| - 1\,.$$
Sumando estas dos igualdades, obtenemos el resultado buscado: $|A(G)| - 1 = |V(G)| - 2$.

En el otro sentido, tomemos un grafo conexo G tal que $|A(G)| = |V(G)| - 1$. Si contuviera un ciclo, podríamos quitar una arista a de ese ciclo sin que el grafo se desconectara. Pero habríamos llegado a un grafo, $G \setminus \{a\}$, conexo con
$$|A(G \setminus \{a\})| = |A(G)| - 1 \qquad \text{y} \qquad |V(G \setminus \{a\})| = |V(G)|\,.$$
Utilizando que $|A(G)| = |V(G)| - 1$,
$$|A(G \setminus \{a\})| = |V(G)| - 2 = |V(G \setminus \{a\})| - 2\,;$$
y esto contradice (¡nos faltan aristas!) el que $G \setminus \{a\}$ sea conexo. ∎

El perspicaz lector echará en falta una caracterización similar en términos de la igualdad $|A| = |V| - 1$ y la ausencia de ciclos. Podrá satisfacer su curiosidad al respecto en el ejercicio 10.2.1 (véase también alguna otra caracterización en el ejercicio 10.2.2).

Resumamos las características que hacen de un grafo G un árbol: es conexo, sin ciclos, tiene $|A(G)| = |V(G)| - 1$ aristas: si quitamos una arista cualquiera, se desconecta; y si añadimos una arista cualquiera, se forma un ciclo. Es decir, como ya adelantábamos, es el grafo conexo más "económico" (en el sentido de que no sobra ni falta arista alguna) posible.

B. Sucesión de grados de un árbol

Son tantas las condiciones que imponemos a un grafo para poder calificarlo como árbol, que el lector no se sorprenderá por que la sucesión de grados de los vértices de un árbol sea peculiar. Ya sabemos que en cualquier grafo $G = (V, A)$ se tiene que

$$\sum_{v\in V} \text{grado}(v) = 2\,|A|\,.$$

Pero si además G es un árbol, como $|A| = |V| - 1$, se tendrá que

$$\sum_{v\in V} \text{grado}(v) = 2\,|V| - 2\,.$$

Obsérvese que ahora sólo tenemos un "grado de libertad", el número de vértices $|V|$, pues el de aristas ya queda fijado. De esta igualdad podemos deducir, por ejemplo, el siguiente resultado.

Proposición 10.7 *Todo árbol con $|V| \geq 2$ tiene, al menos, dos vértices de grado 1.*

Demostración. Supongamos que no hay vértices de grado 1, es decir, que $\text{grado}(v) \geq 2$, para todo $v \in V$. Entonces,

$$2|V| - 2 = \sum_{v\in V} \text{grado}(v) \geq \sum_{v\in V} 2 = 2\,|V|\,,$$

lo que resulta imposible. Pero tampoco puede ocurrir que haya un único vértice w de grado 1, porque tendríamos que

$$2|V| - 2 = \sum_{v\in V} \text{grado}(v) = \text{grado}(w) + \sum_{v\neq w} \text{grado}(v) \geq 1 + 2\,(|V| - 1) = 2|V| - 1$$

Así que al menos ha de haber dos de grado 1. ■

Ejemplo 10.2.1 *¿Cómo son los árboles con n vértices que tienen el menor y el mayor número posible de vértices de grado 1?*

Sabemos que el mínimo número de vértices de grado 1 es 2. Así que, si un árbol con n vértices tiene exactamente dos vértices, digamos w y u, de grado 1, se cumplirá que

$$2n - 2 = \sum_{v\in V} \text{grado}(v) = 1 + 1 + \sum_{v\neq u,w} \text{grado}(v).$$

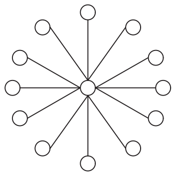
En la suma final tenemos $n-2$ términos, todos ellos mayores o iguales que 2; la única forma de conseguir la igualdad será que $\text{grado}(v) = 2$ para todo $v \in V$, $v \neq u, w$. Y esta configuración de grados es la del grafo lineal con n vértices, L_n.

En el otro extremo, es imposible que todos los vértices tengan grado 1, pues no se cumpliría la fórmula de los grados

$$\sum_{v\in V} \text{grado}(v) = 2|V| - 2\,.$$

Pero sí podría ocurrir que el árbol contuviera $n-1$ vértices de grado 1. El vértice restante, al que por ejemplo vamos a llamar w, necesariamente tendría grado $n-1$, porque la fórmula de los grados ha de cumplirse siempre:

$$2n-2=\sum_{v\in V}\text{grado}(v)=(n-1)+\text{grado}(w)\,.$$

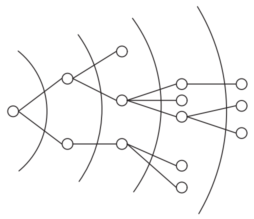


De donde se deduce que

$$\text{grado}(w)=n-1\,,$$

que, por cierto, es el máximo grado que puede tener un vértice en un grafo con n vértices. En este caso, tenemos el grafo estrellado de la izquierda. ♣

C. Árboles con raíz



En muchas ocasiones conviene señalar un vértice especial en un árbol. Lo que resulta es lo que llamaremos un **árbol con raíz**, donde la raíz, por supuesto, es ese vértice especial que sirve de origen de coordenadas. Seguro que el lector estará ya regocijándose con esta peculiar terminología. Hablamos, primero, de árboles, pero los dibujamos boca arriba, boca abajo, o apuntando en las direcciones más diversas. Y ahora nos permitimos la licencia de hablar de raíz, que normalmente situaremos en la parte superior de los esquemas (o, como en el dibujo, a la izquierda del todo). ¡Curiosa fidelidad a la Botánica!

Muchos de los algoritmos que veremos en estas páginas producen, de manera natural, un árbol con raíz, pues en ellos es habitual tomar un vértice como punto de partida, vértice que automáticamente queda etiquetado como la raíz.

En la sección 10.2.3 estudiaremos con detalle algunas aplicaciones de estos objetos y parte de su jerga propia. Por supuesto, cualquier árbol se convierte en uno con raíz en cuanto decidamos qué vértice actúa como tal. Como veremos más adelante, la elección de este vértice especial puede hacer cambiar las propiedades del árbol con raíz.

En un árbol con raíz, y ésta es una idea fundamental para muchos de los argumentos que desarrollaremos, los vértices se agrupan por **generaciones**: la primera contendría sólo a la raíz, la segunda estaría formada por todos los vértices vecinos de la raíz; la tercera, por los vecinos de estos últimos (salvo la raíz); la cuarta, por los vecinos de los de la tercera generación (excepto los que ya estaban en la segunda)... y así sucesivamente, de manera que los vértices de la generación k son aquéllos que están a distancia exactamente $k-1$ de la raíz. Cada vértice de un generación sólo puede estar unidos a vértices de las generaciones anterior y posterior, porque si no tendríamos ciclos.

Lo que la proposición 10.7 afirma es que al menos hay dos vértices "terminales" (pertenecientes a la última generación) en un árbol como éste. A estos vértices especiales los denotaremos más adelante como... ¡sí!, el lector ya lo estará sospechando, por la forma en que los dibujamos, y por el ya acreditado desbarajuste botánico de que estamos haciendo gala: ¡*hojas*!

10.2.1. Contando el número de árboles distintos

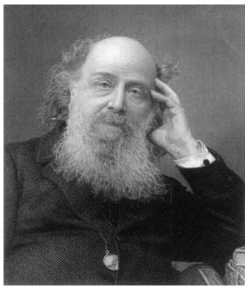


Figura 10.1: Sylvester

Cayley y Sylvester[26] estaban muy interesados en la cuestión de *enumerar* ciertos compuestos orgánicos, uno de los primeros intentos serios de matematización de la Química. Ahora que gran parte de la Química moderna se describe con modelos matemáticos, enternece el comentario que allá por 1830 hacía el filósofo positivista Auguste Comte:

> [...] cualquier intento de emplear métodos matemáticos en el estudio de la Química debe ser considerado como profundamente irracional y contrario al propio espíritu de la Química.

Trataron, por ejemplo, de enumerar los isómeros de hidrocarburos saturados, de fórmula $C_k\, H_{2k+2}$: los átomos de carbono tienen valencia 4, mientras que los de hidrógeno, valencia 1. Son grafos (conexos) con $3k+2$ vértices, cuya suma de grados es $4k+(2k+2)=6k+2$; de manera que tienen $3k+1$ aristas. Una menos que vértices: ¡son árboles!

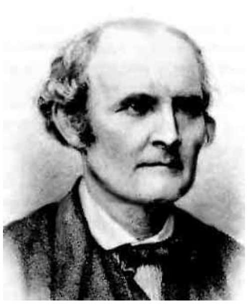


Figura 10.2: Cayley

Enumerar los grafos con ciertas propiedades (ser árboles, tener una determinada sucesión de grados, etc.) es, en general, una cuestión muy complicada. El lector que haya revisado (entregado en cuerpo y alma, claro) la discusión de la subsección 10.1.2 tendrá ya una idea cabal de esa dificultad. Aunque en ocasiones, sobre todo si etiquetamos los vértices, las cosas se simplifican. Un caso especialmente ilustrativo es el de los grafos distintos (esto es, etiquetados) que se pueden formar con un conjunto dado de n vértices, de los ya sabemos que hay $2^{\binom{n}{2}}$. En el caso que nos va a ocupar, el de los árboles con n vértices, como Cayley ya sabía[27], la respuesta es también una fórmula (sorprendentemente) sencilla. Empecemos nuestra investigación, como parece prudente y razonable, analizando ejemplos de árboles con pocos vértices.

[26] Las vidas de los ingleses Arthur Cayley (1821-1895) y James Joseph Sylvester (1814-1897) corren paralelas. Estudiaron ambos en Cambridge pero, por diversas razones, durante muchos años no ejercieron como matemáticos. Cayley fue abogado durante 14 años, aunque durante ese periodo de "matemático aficionado" llegó a publicar más de 200 artículos de investigación. En esa labor coincidió con Sylvester, que no había llegado a graduarse por negarse a realizar juramento de fidelidad a la Iglesia de Inglaterra (Sylvester era judío). Juntos gustaban de discutir sobre casos judiciales y teoremas (incluyendo algunos de Teoría de grafos). A partir de 1863, Cayley se dedicó en exclusiva a las Matemáticas, desde su puesto en Cambridge. Sylvester, por su parte, fue matemático, abogado... y poeta. A los 27 años fue contratado por la Universidad de Virginia. Se cuenta que allí, durante una lección, tuvo un serio incidente con un alumno que estaba leyendo un periódico en clase, al que Sylvester (a quien se describe como un hombre de fuerte carácter) golpeó. Creyéndolo muerto, tomó el primer barco de regreso a Inglaterra. En 1877 volvió a cruzar el charco, contratado por la Universidad John Hopkins. Un año después fundaría la primera revista norteamericana de matemáticas: *American Journal of Mathematics*. De vuelta a Inglaterra (sin más incidentes por medio, que se sepa), enseñó en Oxford hasta su muerte. Gran parte de los trabajos de Sylvester sobre invariantes algebraicos, matrices y determinantes fueron realizados en colaboración con Cayley, quien es considerado uno de los padres de la teoría de grupos. También son fundamentales sus aportaciones a la geometría no euclídea y la geometría proyectiva.

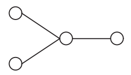
[27] Aparentemente, fue Sylvester el primero que propuso este resultado, en 1857. La primera demostración es de Borchardt, en 1860. Los trabajos de Cayley en este campo son de 1889, y en ellos enumeró grafos con diversas propiedades. Años después, Pólya desarrollaría una teoría enumerativa general (véase el capítulo 18) para abordar estas cuestiones.

Ejemplo 10.2.2 *Contemos el número de árboles distintos con* 2*,* 3*,* 4 *y* 5 *vértices.*

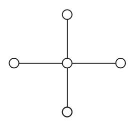
Como en la subsección 10.1.2, determinaremos primero las distintas almas, para luego, para cada una de ellas, contar los posibles etiquetados. Para ser sistemáticos, nos iremos guiando por las distintas sucesiones de grados posibles. En un grafo con n vértices, el grado de un vértice no puede mayor que $n-1$. Como además G es un árbol, no podrá haber vértices de grado 0 (pues es conexo) y la sucesión de grados debe cumplir que

$$\sum_{v\in V(G)} \text{grado}(v) = 2|V(G)| - 2\,, \quad \text{donde } 1 \leq \text{grado}(v) \leq |V(G)| \text{ para todo } v \in V(G).$$

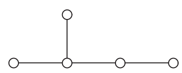
En el caso de dos vértices sólo cabe la posibilidad de que el árbol sea isomorfo a un L_2. Si el conjunto de vértices es $\{1,2\}$, hay también un único árbol (los dos "posibles" etiquetados de los vértices dan el mismo resultado). Si tenemos tres vértices, sólo tenemos un alma posible, la que corresponde al grafo lineal L_3. Para etiquetarlo con el conjunto $\{1,2,3\}$, basta con decidir qué símbolo va, por ejemplo, en la posición central (cuál es el vértice de grado 2). Esto se puede hacer de tres formas distintas, así que hay 3 árboles distintos con vértices $\{1,2,3\}$.



Vamos con el caso de cuatro vértices. El grado máximo es ahora 3. Un simple análisis, que recomendamos haga el lector, nos lleva a concluir que sólo puede haber dos sucesiones de grados admisibles, $(1,1,1,3)$ y $(1,1,2,2)$. La primera de ellas se corresponde con el (alma del) grafo que dibujamos a la izquierda y arriba, mientras que la segunda se traduce en el de debajo. Para etiquetar el primer grafo con $\{1,2,3,4\}$, basta con decidir el símbolo que va en la posición central; así que hay 4 maneras distintas de hacerlo. El etiquetado del otro es más delicado: elegimos primero las etiquetas de los dos vértices de grado 2 (se puede hacer de $\binom{4}{2}$ formas); y para cada elección de éstas, hay luego dos posibilidades para nombrar los vecinos. En total, 12 maneras distintas. En resumen, con cuatro vértices hay 2 árboles no isomorfos y 16 árboles distintos con vértices $\{1,2,3,4\}$.



Para el caso de cinco vértices, el grado ha de ser a lo sumo 4. Si hay de grado 4, la sucesión de grados debe ser $(1,1,1,1,4)$, a la que le corresponde un único árbol salvo isomorfismos, el que aparece a la derecha. Etiquetarlo con $\{1,2,3,4,5\}$ es muy sencillo, pues basta decidir qué situamos en el vértice central: en total, 5 posibilidades.



Si no hay de grado 4, pero sí de grado 3, la única sucesión de grados posible es $(1,1,1,2,3)$ (alma de la derecha). Para etiquetarla, fijamos el símbolo del vértice de grado 3 (5 posibilidades), luego el del de grado 2 (4 posibilidades) y, finalmente, elegimos (3 posibilidades) el vecino de grado 1 del vértice de grado 2 (o bien los dos vecinos de grado 1 del vértice de grado 3). En total, 60 posibilidades.

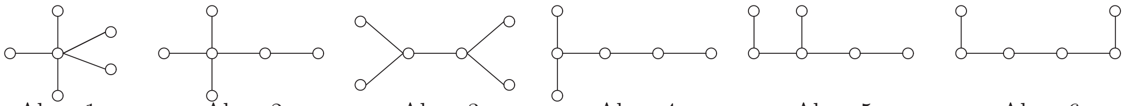
Por último, si no hay vértices de grado 3, entonces sólo podremos tener la sucesión de grados $(1,1,2,2,2)$, que corresponde a un L_5. Para etiquetarlo, elegimos el símbolo del vértice central (5 posibilidades), luego los otros dos de grado 2 ($\binom{4}{2} = 6$ posibilidades) y ya sólo quedan dos posibilidades para etiquetar los vértices finales. En total, $5 \times 6 \times 2 = 60$ formas distintas. Resumiendo, con 5 vértices hay tres árboles no isomorfos y 125 árboles distintos con vértices $\{1,2,3,4,5\}$. ♣

Ejemplo 10.2.3 *Un poco más difícil: árboles con seis vértices $\{v_1,\dots,v_6\}$.*

Los grados son ahora seis números entre 1 y 5 que suman 10. Si hay de grado 5, sólo podrá haber uno, y la sucesión de grados será $(1,1,1,1,1,5)$. El único grafo con estas características es el que aparece debajo de estas líneas como *Alma 1*. El etiquetado de los vértices con $\{v_1,\dots,v_6\}$ sólo requiere decidir el símbolo del vértice central: seis posibilidades.

Si no hay de grado 5, pero sí de grado 4, tendremos la sucesión $(1,1,1,1,2,4)$ (véase el *Alma 2*). El lector puede comprobar que hay 120 árboles distintos asociados a esta estructura.

Si no hay vértices de grado 4, pero sí de grado 3, las cosas se ponen interesantes. Por un lado, la sucesión de grados $(1,1,1,1,3,3)$ está asociada al *Alma 3*. El lector debería comprobar que hay 90 formas distintas de etiquetar sus vértices. Pero, por otro lado, y como ya vimos en la subsección 10.1.2, hay *dos* árboles no isomorfos (*Almas 4* y *5*) asociados a la sucesión de grados $(1,1,1,2,2,3)$. Es, de nuevo, un sencillo[28] ejercicio comprobar que hay 180 etiquetados distintos para la primera y 360 para segunda.



Alma 1 Alma 2 Alma 3 Alma 4 Alma 5 Alma 6

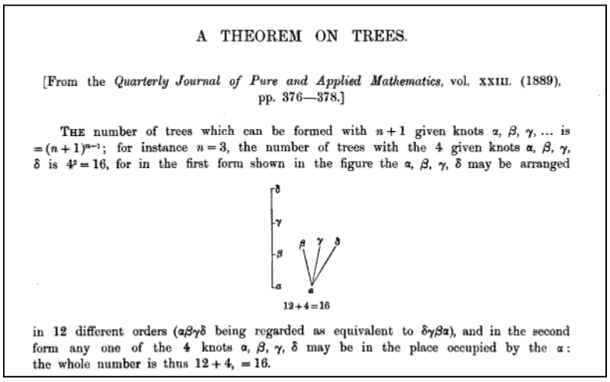
Por último, si no hay de grado 3, entonces la sucesión de grados es $(1,1,2,2,2,2)$ (véase el *Alma 6*, un L_6), que tiene 360 distintos etiquetados de sus vértices. En resumen, hay seis árboles no isomorfos con 6 vértices y 1296 árboles distintos con vértices $\{1,\dots,6\}$. ♣

Fue Cayley[29] quien, en 1889, probó que el patrón que se observa en los resultados obtenidos en el laborioso análisis hecho hasta aquí,

n	2	3	4	5	6
árboles distintos	$1=2^0$	$3=3^1$	$16=4^2$	$125=5^3$	$1296=6^4$

es completamente general:

Teorema 10.8 (Cayley) *El número de árboles distintos que se pueden formar con el conjunto de vértices $\{1,\dots,n\}$ es n^{n-2}.*



A THEOREM ON TREES.

[From the *Quarterly Journal of Pure and Applied Mathematics*, vol. XXIII. (1889), pp. 376—378.]

THE number of trees which can be formed with $n+1$ given knots $\alpha, \beta, \gamma, \dots$ is $=(n+1)^{n-1}$; for instance $n=3$, the number of trees with the 4 given knots $\alpha, \beta, \gamma, \delta$ is $4^2=16$, for in the first form shown in the figure the $\alpha, \beta, \gamma, \delta$ may be arranged

in 12 different orders ($\alpha\beta\gamma\delta$ being regarded as equivalent to $\delta\gamma\beta\alpha$), and in the second form any one of the 4 knots $\alpha, \beta, \gamma, \delta$ may be in the place occupied by the α: the whole number is thus $12+4, =16$.

A la derecha mostramos las primeras líneas de su artículo. Quizás el lector pueda entrever las dos estructuras que dibujábamos en el caso $n=4$. Existe un buen número de demostraciones distintas de este elegante resultado, como las que recogemos en los ejercicios 10.2.4 y 10.2.5. Esta última es especialmente interesante porque, además, proporciona un mecanismo para codificar la información que determina un árbol,

[28] ¿Sencillo?: clásica exageración que suele desesperar al lector de los libros de matemáticas. Lo es si somos cuidadoso y sistemáticos; pero la cuestión de etiquetar estructuras (árboles, en este caso) con simetrías variadas dista mucho de ser sencilla, y su análisis requiere del lenguaje de la Teoría de Grupos (capítulos 17 y 18).

[29] Aunque se dice que fue Borchardt, en 1860, quien se dio cuenta por primera vez.

mediante el llamado *código de Prüfer*. Si el lector no sintiera todavía saciada su curiosidad, puede revisar también otras demostraciones, con argumentos de características bien distintas, en el ejemplo 10.2.7 y en la sección 15.3).

La prueba que veremos aquí, debida a Zeilberger[30], es especialmente sencilla y sugerente.

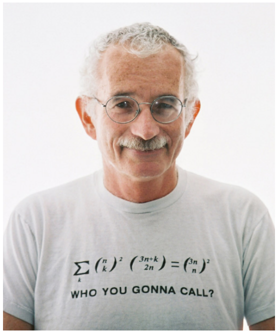


FIGURA 10.3: Zeilberger

DEMOSTRACIÓN. La escribiremos con el lenguaje (siempre simpático) que utiliza Zeilberger, aunque luego lo reinterpretaremos en términos de grafos. En una cierta empresa hay m jefes y k empleados, con sus respectivos nombres y apellidos. Los queremos organizar de manera que cada empleado tenga un *único* supervisor (que pudiera ser otro empleado o quizás un jefe). Los jefes no tienen supervisor (¿tautología?), y podrían no tener subordinados. Al número de formas distintas de organización lo llamaremos $P(m, k)$.

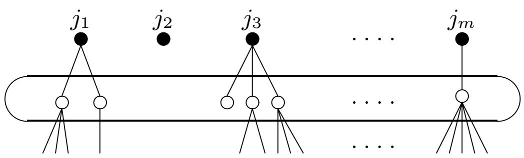
Digamos que $m \geq 1$ y que $k \geq 0$. Es claro que $P(m, 0) = 1$ para cada $m = 1, 2, \ldots$. En palabras, para obtener otros valores de $P(m, k)$, seguimos el siguiente proceso:

1) Distinguimos a los empleados cuyos supervisores son jefes. A estos empleados, de los que habrá un cierto número s, con $1 \leq s \leq k$, se les asigna, desde este momento, y como no podía ser de otro modo, el papel de *jefecillos*.
2) Decidimos, pues, qué s empleados son jefecillos; esto se puede hacer de $\binom{k}{s}$ maneras.
3) Y ahora asignamos el jefe ante el que responde cada jefecillo: m^s maneras.
4) Queda organizar a los restantes $k-s$ empleados. Pero, para éstos, los s jefecillos actúan como jefes. Así que los podremos organizar de tantas maneras como nos diga $P(s, k-s)$.

En total,

$$P(m, k) = \sum_{s=1}^{k} \binom{k}{s} m^s \, P(s, k-s)\,.$$

Esta regla de recurrencia, junto con los valores iniciales de antes, determina de manera única el valor de los $P(m, k)$, como el lector puede comprobar.



En el lenguaje de los grafos, $P(m, k)$ cuenta el número de bosques (con $m + k$ vértices) formados por m árboles con raíz, donde las raíces van etiquetadas con los nombres de los jefes. El argumento anterior, en estos nuevos términos, consiste en decidir qué vértices van en la segunda generación (por debajo de las raíces), y cómo se distribuyen en esa segunda generación. Nos interesa el caso $m = 1$ y $k = n - 1$: un árbol con raíz con n vértices, con la raíz ya etiquetada.

Atención ahora al ida y vuelta: si partimos de los símbolos $\{1, \ldots, n\}$, hay n maneras de etiquetar la raíz. Así que $nP(1, n-1)$ cuenta cuántos árboles con raíz podemos formar con

[30] Doron Zeilberger (1950-), nacido en Haifa (Israel), trabaja en la Universidad de Rutgers. Su página *web* `www.math.rutgers.edu/~zeilberg/` es una auténtica delicia que recomendamos al lector.

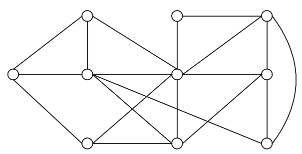
n vértices. Pero, por cada árbol con n vértices, hay n árboles con raíz distintos (pues hay n formas de elegir la raíz). Concluimos así que la cantidad que nos interesa, el número de árboles con n vértices, coincide con $P(1, n-1)$.

Queda resolver la recurrencia anterior para obtener el valor de $P(m, k)$. Aquí llega el ingrediente ingenioso de la prueba. La ecuación de antes nos recuerda vagamente a las sumas que se obtienen en el teorema del binomio. El lector podrá comprobar, utilizando este teorema, que la función $f(m, k) = m(m + k)^{k-1}$ cumple la misma regla de recurrencia que $P(m, k)$ y tiene los mismos valores iniciales. Así que $P(m, k) = f(m, k)$ para todo m y k. En particular,

$$P(1, n-1) = f(1, n-1) = 1 \times (1 + (n-1))^{(n-1)-1} = n^{n-2},$$

y el resultado de Cayley queda demostrado. Obsérvese que, de camino[31], hemos comprobado también que $P(m, k) = m(m + k)^{k-1}$. ■

10.2.2. Árboles abarcadores de un grafo



Al principio del capítulo planteamos la siguiente cuestión: construir una red que conecte una serie de puntos (por ejemplo, un sistemas de oleoductos, una red de ordenadores) de la forma más barata (en cuanto a número de conexiones) a partir de un diseño previo, como el de la figura. El objetivo es eliminar el mayor número posible de aristas de manera que el grafo siga siendo conexo. O, dicho de otra manera, quedarnos con el número mínimo de aristas que garantizan la conexión del grafo. Estamos buscando, en definitiva, un árbol que incluya a todos los vértices.

Definición 10.12 *Consideremos un grafo $G = (V, A)$. Diremos que un árbol H es* ***árbol abarcador***[32] *de G si cumple que:*

- *$V(H) = V(G)$ (tiene los mismos vértices que G).*
- *$A(H) \subseteq A(G)$ (tiene algunas —o todas— las aristas de G).*

Es decir, es un subgrafo abarcador del grafo inicial que, además, es un árbol. Asegurémonos primero de que tales árboles existen si, como es razonable, partimos de un grafo conexo.

Proposición 10.9 *Un grafo G es conexo si y sólo si tiene, al menos, un árbol abarcador.*

Demostración. En un sentido, si un grafo tiene un árbol abarcador, que por definición es conexo, es obvio que G también lo será.

En el otro: consideremos un grafo G conexo. Si es un árbol, ya hemos acabado (G es su propio árbol abarcador). Pero si no es árbol, como es conexo, podemos todavía quitar una cierta arista a sin que se desconecte (recordemos el carácter extremal de los árboles en este

[31] ¡Quizás un resultado más interesante que el propio de Cayley!

[32] *Spanning tree*, en la terminología anglosajona, *arbre de recouvrement* en la francesa. En castellano se utilizan términos como "árbol generador" o "árbol recubridor". Nuestra elección trata de reflejar que el árbol contiene (abarca) todos los vértices del grafo.

sentido). Así que el grafo $G \setminus \{a\}$ sigue siendo conexo. Si fuera un árbol, habríamos acabado; pero si no lo fuera, aún podríamos quitar una arista b de forma que $(G \setminus \{a\}) \setminus \{b\}$ seguiría siendo conexo. Y así, sucesivamente. En cada paso, los sucesivos grafos son conexos; cuando el número de aristas llegue a $|V(G)| - 1$, habremos llegado a un árbol abarcador. ■

Nótese que si dispusiéramos de algoritmos *eficaces* para hallar árboles abarcadores (o para decidir que no los hay), tendríamos una manera de determinar si un grafo es o no conexo.

De la demostración anterior, además, deducimos que si el grafo G es conexo y $|A(G)| = |V(G)| - 1 + k$, con $k \geq 0$, entonces podemos quitar k aristas (convenientemente escogidas para no perder la conexión) y quedarnos con un árbol abarcador. Se trata de ir "eliminando aristas" hasta quedarnos con el número adecuado de ellas. El criterio para eliminarlas sería el de ir identificando y "rompiendo" los ciclos del grafo.

Sin embargo, existen algoritmos mucho más eficaces para construir un árbol abarcador en un grafo (o para concluir que no los hay). En estos procedimientos, en lugar de ir eliminando aristas, se va "haciendo crecer" el árbol en pasos sucesivos, como veremos en un momento.

Estos algoritmos están diseñados, en principio, para buscar y etiquetar vértices de un grafo, y en realidad muchos de los algoritmos que veremos en la sección 11.4 son variaciones de éstos adaptadas a distintos problemas (paseos más cortos, árboles abarcadores de peso mínimo, etc.). Aquí veremos que, además, sirven para dar una respuesta (computacionalmente sencilla) a cuestiones como la de decidir si un grafo es conexo (o, en caso contrario, identificar componentes conexas), localizar ciclos, decidir si una arista es puente, etc.

A. Algoritmos de búsqueda en grafos y (su conexión con) conexión

Los dos procedimientos que vamos a describir, aunque siguen filosofías distintas (descritas adecuadamente con los respectivos nombres), dan como resultado árboles (con raíz) que incluyen unos cuantos vértices del grafo de partida (o quizás todos). Justamente en el caso en que el árbol producido incluya a todos los vértices del grafo (es decir, cuando sea un árbol abarcador), podremos afirmar que el grafo es conexo. Es ésta la razón fundamental por la que los traemos a colación en esta subsección. Nos limitaremos a describir la idea básica de cada uno de los procedimientos, y el lector con especiales inclinaciones algorítmicas podrá entretenerse en traducirlos en algoritmos más detallados.

A1. Algoritmo de Búsqueda a lo Ancho (BA)[33].

El procedimiento parte de un grafo $G = (V, A)$ y construye un árbol (con raíz) cuyos vértices están agrupados por generaciones en función de su distancia a la raíz.

Marcamos un cierto vértice inicial, digamos v_1. Localizamos entonces todos sus vecinos y los etiquetamos. Las aristas correspondientes quedan incorporadas al árbol. Una vez hecho esto y numerados los vértices vecinos del inicial, pasamos al que sea primero en esa ordenación y repetimos el proceso: localizamos todos sus vecinos nuevos y los añadimos al árbol. Y así sucesivamente, hasta que no podamos incorporar más vértices (ni aristas) al árbol.

[33] *Breadth-First Search* (BFS), en la terminología anglosajona.

Obsérvese que, en el proceso de “localización” de vértices nuevos, cada arista es examinada como mucho dos veces (una por cada uno de sus extremos), así que el número de pasos en este algoritmo es[34] $O(|A|)$. Si G tiene n vértices, el número de pasos será $O(n^2)$.

Si el grafo es conexo, el resultado final es un árbol abarcador (con raíz) de G. Si no lo fuera, se detendría al completar un árbol abarcador de la componente conexa de G en la que estuviera emplazado el vértice inicial v_1.

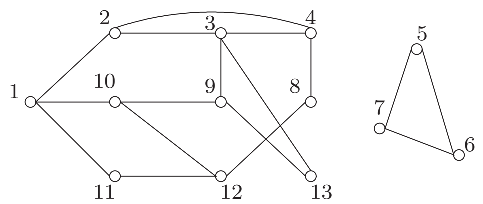
A2. Algoritmo de Búsqueda en Profundidad (BP)[35]

Ahora procedemos de la siguiente manera: empezamos de nuevo con un cierto vértice v_1 y localizamos sus vecinos. Se toma uno de ellos como v_2 e incorporamos la arista $\{v_1, v_2\}$ al árbol. Entonces **avanzamos** a v_2 y buscamos vecinos que no hayan sido ya visitados; tomamos cualquiera de ellos como v_3, y así sucesivamente. Cuando lleguemos a un vértice que no tenga vecinos nuevos que añadir, digamos v_k, **retrocedemos** al vértice anterior, v_{k-1}, y ahí repetimos el proceso (buscamos nuevos vecinos que añadir al árbol).

Si en algún momento del proceso nos encontramos en v_1 sin sitios nuevos a donde ir (quizás hayamos pasado antes varias veces por v_1, pero pudiendo añadir nuevos vértices), se termina el algoritmo. El árbol T así generado es un árbol abarcador de la componente conexa que incluye al vértice v_1. Como cada arista del grafo se utiliza, como mucho, dos veces en el algoritmo (una en el “avance”, otra en el “retroceso”), también el número de pasos del algoritmo será $O(|A|)$ (o quizás $O(|V| + |A|)$), lo que nos da de nuevo $O(n^2)$ si el grafo tiene n vértices.

Ilustramos las diferencias entres los dos algoritmos en el siguiente ejemplo.

Ejemplo 10.2.4 *Apliquemos los algoritmos BA y BP al siguiente grafo.*



Empecemos, por ejemplo, con el algoritmo BP, digamos partiendo del vértice 1 (que sería el v_1).

$$\begin{array}{ccccccccccc} v_1 & & v_2 & & v_3 & & v_4 & & v_5 & & v_6 \\ 1 & \longrightarrow & 2 & \longrightarrow & 4 & \longrightarrow & 3 & \longrightarrow & 9 & \longrightarrow & 13 \end{array}$$

De él pasamos al 2, al 4, al 3, al 9 y al 13. Una vez en el vértice 13 (que es v_6) vemos que todos sus vecinos ya han sido visitados, así que retrocedemos a v_5 (el vértice 9).

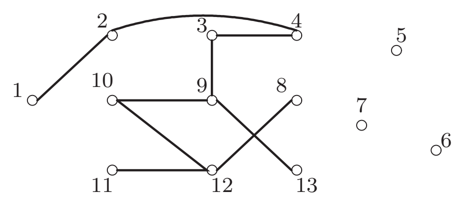
$$\begin{array}{ccccccc} & & v_7 & & v_8 & & v_9 \\ 9 & \longrightarrow & 10 & \longrightarrow & 12 & \longrightarrow & 11 \end{array}$$

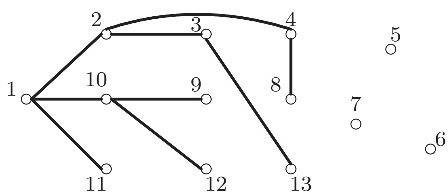
Desde allí podemos visitar nuevos vecinos (observemos que la arista entre 9 y 13 la recorremos en los dos sentidos, de ida y de vuelta). Por ejemplo, con el camino que mostramos a la derecha. Al no poder añadir vecinos desde el vértice 11 (v_9), retrocedemos a v_8, el vértice 12. Desde allí podemos incluir al vértice 8 como v_{10}. Y una vez en éste, nos vemos obligados a retroceder a v_9, v_8, etc., hasta volver a v_1.

[34] Para ser más precisos, es $O(|V| + |A|)$.

[35] *Depth-First Search* (DFS), en la terminología anglosajona.

Como ya no podemos añadir nuevos vértices, el algoritmo termina. A la derecha dibujamos el árbol resultante, que no incluye a todos los vértices del grafo (el grafo original no era conexo), sólo a los de la componente conexa del vértice 1. Nótese que en el proceso hemos recorrido cada arista del árbol en ambos sentidos.





Vamos ahora con el algoritmo BA. De nuevo etiquetaríamos el vértice 1 como v_1, y a sus vecinos 2, 10 y 11 como v_2, v_3 y v_4, respectivamente. Una vez considerados todos los posibles vecinos de v_1 (y añadidas sus correspondientes aristas al árbol), nos movemos a v_2, el vértice 2; ahí añadimos a 3 como v_5 y a 4 como v_6. Es hora de ir al vértice $v_3 = 10$ y añadir a 9 como v_7 y a 12 como v_8. En el vértice $v_4 = 11$ no hay vértices nuevos que considerar, pero sí en el v_5 (el vértice 3): etiquetamos el 13 como v_9. Desde $v_6 = 4$ añadimos el vértice 8 como v_{10}. Y al mirar en los vértices del v_7 al v_{10} comprobamos que no podemos ir más allá con el árbol. El algoritmo ha terminado y ha producido el árbol de la izquierda. De nuevo, el árbol sólo abarca a los vértices de la componente conexa a la que pertenecía el vértice 1. ♣

Ambos algoritmos tienen sus ventajas y sus desventajas, en función del problema que se esté tratando. Pero, para lo que aquí nos interesa, ambos permiten resolver de manera sencilla (desde el punto de vista computacional) la siguiente cuestión básica:

- *determinar si un cierto grafo G es conexo o no* (comprobando si el árbol que produce el algoritmo incluye todos los vértices del grafo; esto es, si es árbol abarcador o no).

Con ligeras variaciones, permiten también resolver las siguientes cuestiones:

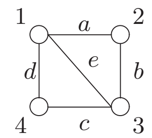
- *determinar componentes conexas de G* (arrancando con un vértice, identificando su componente conexa, y repitiendo luego el proceso para vértices que no estén en ella);
- *determinar si G tiene o no ciclos* (localizando las componentes conexas y contando las aristas de cada una de ellas; esto es, comprobando si son árboles);
- *decidir si una cierta arista es puente o no del grafo* (comparando el número de componentes conexas antes y después de eliminar la arista en cuestión).

O algunas otras, algo más sofisticadas, sobre las que no daremos los detalles:

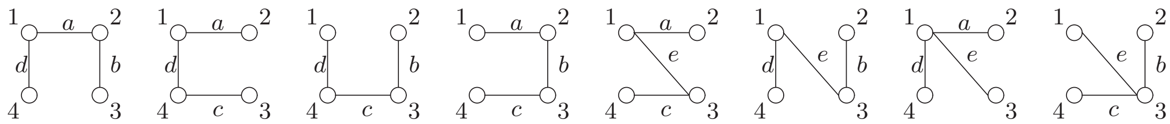
- *encontrar el ciclo más corto en el grafo G* (y, por tanto, determinar cuello(G));
- *determinar si un grafo G es bipartito o no* (utilizando la caracterización, en términos de la presencia de ciclos de longitud impar del ejercicio 10.1.11).

Por su propia estructura, además, el algoritmo BA permite determinar las distancias de un vértice a los demás del grafo. En la subsección 11.4.2 volveremos sobre este asunto, aunque allí en el contexto más general de los grafos con pesos. En otros apartados de la sección 11.4 aparecerán diversos algoritmos muy relacionados con BA y BP. Pero, para lo que sigue, quedémonos con la idea de que los problemas que hemos expuesto en esta página se resuelven de manera (computacionalmente) "sencilla" con este tipo de algoritmos.

B. El número de árboles abarcadores de un grafo



Que un grafo conexo tiene, en general, más de un árbol abarcador es bastante evidente. ¿Cuántos, exactamente? Dependerá, claro, del grafo considerado. En el que aparece dibujado a la derecha nos "sobran" dos aristas. Pero no cualesquiera: por ejemplo, si quitáramos las aristas a y b, desconectaríamos el grafo. Compruebe el lector que el grafo tiene ocho árboles abarcadores distintos:



Para algunas familias de grafos, la respuesta es bastante sencilla.

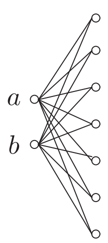
Ejemplo 10.2.5 *El número de árboles abarcadores de los grafos circulares, los grafos lineales, los grafos completos... ¡y de los propios árboles!*

Consideremos el *grafo circular* con n vértices, C_n, para el que $|A(C_n)| = |V(C_n)|$. Formar un árbol abarcador consiste, simplemente, en quitar una arista; y cualquiera de las n que hay vale para ello. Así que C_n tiene n posibles árboles abarcadores.

El *grafo lineal* con n vértices, L_n, es ya un árbol, así que es su propio árbol abarcador. Éste es un resultado general: un árbol G sólo tiene un árbol abarcador (él mismo).

El *grafo completo* K_n contiene todas las aristas posibles, así que hay tantos árboles abarcadores como árboles con n vértices haya: n^{n-2} (recuérdese el teorema 10.8 de Cayley). ♣.

Ejemplo 10.2.6 *¿Cuántos árboles abarcadores distintos contiene un $K_{r,s}$?*



Empecemos con un grafo bipartito completo $K_{2,s}$, $s \geq 2$, que tiene $2s$ aristas y $s+2$ vértices. Necesitamos quitar $s-1$ aristas sin que se desconecte el grafo. Obsérvese que un árbol abarcador del grafo contendrá a una serie de vértices de los de la derecha conectados al vértice a y otros conectados a b. Pero ha de haber al menos uno que se conecte a ambos, para que sea conexo. Y sólo uno, porque si dos (o más) vértices a la derecha conservaran sus dos aristas, se formaría un ciclo. Con esta información podemos argumentar como sigue:

Digamos que los vértices de la "derecha" son $\{1,\ldots,s\}$ y llamemos A y B a los conjunto de ellos que comparten arista, respectivamente, con a y b en el árbol abarcador. Así que elegir un árbol abarcador es *exactamente* lo mismo que escoger A y B de manera que $A \cup B = \{1,\ldots,s\}$ y $|A \cap B| = 1$. Para contar el número de maneras de elegir A y B, sigamos el siguiente proceso:

1. Elegimos el elemento de la intersección (hay s posibilidades).
2. Luego basta decidir si el resto de los elementos están en A ó en B. Equivalentemente, formar una $(s-1)$-lista con repetición permitida con dos símbolos ("estar en A" o "estar en B"). Esto se puede hacer de 2^{s-1} formas distintas.

Así que hay $s\,2^{s-1}$ árboles abarcadores de $K_{2,s}$. El resultado general afirma que un grafo $K_{r,s}$ tiene $s^{r-1}\,r^{s-1}$ árboles abarcadores distintos (véase en el ejercicio 10.2.9 el caso $K_{3,s}$). ♣

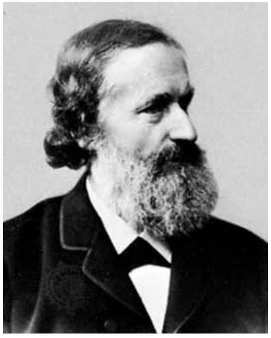


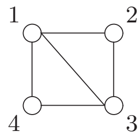
FIGURA 10.4: Kirchhoff

Si un grafo conexo G tiene n vértices, entonces el número de árboles abarcadores de G puede oscilar entre 1 (si G ya es un árbol) y n^{n-2} (si se trata del grafo completo). Pero el número exacto depende de la estructura particular del grafo.

Ésta no es una cuestión exenta de interés. Supongamos que un grafo (conexo) representa una red eléctrica; estamos interesados en la "fiabilidad" de nuestra red. Esto es, si puede ser capaz de soportar la pérdida de algunos tramos (aristas) sin que el fluido se interrumpa. El que el grafo tenga muchos árboles abarcadores sugiere que el sistema será bastante robusto, y que podrá sobreponerse a estas pérdidas. Estas cuestiones le interesaban, y mucho, a Kirchhoff[36], que en 1847 enunció el siguiente resultado, del que no incluiremos la demostración, que permite calcular algebraicamente el número de árboles abarcadores de un grafo arbitrario (véase un procedimiento alternativo en el ejercicio 10.2.11).

Teorema 10.10 (Kirchhoff) *El número de árboles abarcadores de un grafo G con matriz de vecindades M coincide con un cofactor*[37] *cualquiera de la matriz*[38] *diferencia $D - M$, donde D es la matriz diagonal cuyos registros son los grados de los vértices.*

Calculemos, usando este teorema, el número de árboles abarcadores del grafo que aparece a la derecha, el que abría este apartado y del que sabemos, tras la enumeración correspondiente, que tiene ocho árboles abarcadores. Las correspondientes matrices M y D son



$$M = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \text{ y } D = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}; \text{ así que } D - M = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 2 & -1 & 0 \\ -1 & -1 & 3 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix}.$$

Por ejemplo, el cofactor que se obtiene eliminando la primera fila y columna resulta ser

$$(-1)^{1+1} \begin{vmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{vmatrix} = 8\,.$$

Compruebe el lector que los demás cofactores llevan al mismo resultado. ♣

[36]Gustav Robert Kirchhoff (1824-1887) es otro de los padres de la Teoría de grafos. Nació en Königsberg, lo que no deja de sonar algo premonitorio. En sus estudios sobre circuitos eléctricos formuló las famosas **leyes de Kirchhoff**, que relacionan corrientes, voltajes y resistencias, generalizando así las leyes de Ohm. Kirchhoff se ocupó también de analizar otros fenómenos físicos, como la radiación del cuerpo negro, o el estudio de los espectros de los cuerpos celestes. Junto con el químico Bunsen, diseñó uno de los primeros espectroscopios, con el que, por ejemplo, se descubrirían elementos como el cesio y el rubidio.

[37]Dada una matriz A $n \times n$, el *menor* M_{ij} es el determinante de la matriz que se obtiene al quitarle a A la fila i y la columna j. El *cofactor* correspondiente es $C_{ij} = (-1)^{i+j} M_{ij}$.

[38]En la terminología habitual de la Teoría espectral de grafos, es la *matriz laplaciana*. Obsérvese que los registros de cada una de sus filas (o columnas) suman 0, así que es una matriz singular (de determinante 0).

Ejemplo 10.2.7 *Kirchhoff y Cayley.*

Queremos contar, de nuevo, cuántos árboles abarcadores tiene un grafo completo K_n.

$$M = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 0 \end{pmatrix} \quad \Longrightarrow \quad D - M = \begin{pmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ -1 & -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & n-1 \end{pmatrix}$$

La matriz de la derecha tiene dimensiones $n \times n$. Para calcular el número de árboles abarcadores de K_n, como nos dice el resultado de Kirchoff, debemos calcular uno cualquiera de sus cofactores. Dejamos al lector que se entretenga efectuando las operaciones de eliminación gaussiana para calcular el determinante correspondiente y que concluya que, una vez más, que el grafo K_n tiene n^{n-2} árboles abarcadores distintos (véase el ejercicio 10.2.12). ♣

El resultado de Kirchhoff reduce, pues, la cuestión de calcular el número de árboles abarcadores de un grafo a los procedimientos típicos del Álgebra lineal, como es el cálculo de determinantes. Para que el lector se asombre con la utilidad de este procedimiento, señalemos que permite resolver, por ejemplo, el caso del grafo del cubo Q_n. La sorprendente fórmula es

$$\#\{\text{árboles abarcadores de } Q_n\} = \prod_{k=2}^{n} (2k)^{\binom{n}{k}}$$

Aparentemente, no se conoce una prueba de carácter combinatorio de este resultado.

10.2.3. Representación de algoritmos con árboles con raíz

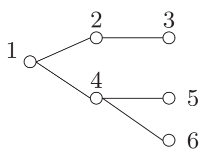
Muchos de los algoritmos básicos de la Computación están organizados en pasos sucesivos, en cada uno de los cuales se toma una decisión de entre varias posibilidades. Pensemos, por ejemplo, en un procedimiento que permita ordenar los elementos de un cierto conjunto con arreglo a determinado criterio basándose en *comparaciones* sucesivas entre dos (o más) elementos del conjunto.

Estos algoritmos se corresponden, de manera natural, con árboles (con raíz), cuyas características específicas dependerán del tipo de comparaciones realizadas.

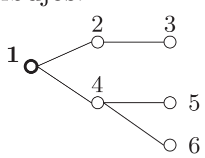
El objetivo de esta subsección es el de obtener, a partir del análisis de este tipo de árboles, cotas teóricas para el número de pasos de que debe constar un cierto algoritmo. Esto lo haremos en el apartado A. En el apartado B revisaremos un cálculo que resultará fundamental en ciertos problemas de codificación (relacionados con los llamados *códigos de Hufmann*, véase el capítulo 16).

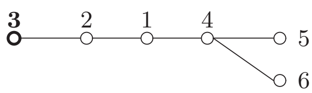
Para todo esto, necesitaremos introducir cierta terminología sobre árboles con raíz, objetos que ya presentamos al comienzo de esta sección dedicada a los árboles. Un árbol con raíz no es más que un árbol en el que designamos un vértice especial, la **raíz**, que sirve de origen de coordenadas. Los vértices del árbol se agrupan en niveles:

$$\begin{array}{rcl}
\text{Nivel } 0 & = & \{\text{raíz}\} \\
\text{Nivel } 1 & = & \{\text{vecinos de la raíz}\} \\
\text{Nivel } 2 & = & \{\text{vecinos de los vértices del nivel } 1\} \setminus \{\text{raíz}\} \\
\text{Nivel } 3 & = & \{\text{vecinos de los vértices del nivel } 2\} \setminus \{\text{vértices del Nivel } 1\} \\
\vdots & & \vdots \\
\text{Nivel } j & = & \{\text{vecinos de los vértices del nivel } j-1\} \setminus \{\text{vértices del Nivel } j-2\}
\end{array}$$



Llamaremos a, la **altura** del árbol, al máximo nivel no vacío. Es importante recordar que el valor de a depende de la raíz elegida. Por ejemplo, si partimos del árbol que aparece dibujado a la izquierda, cualquiera de sus vértices puede servir como raíz. Elegir, por ejemplo, el vértice 1 o el 3 lleva a que la altura del árbol sea 2 ó 4, como se aprecia en los siguientes dibujos:





Para ir distinguiendo los distintos papeles de los vértices en el árbol, diremos que

- los **descendientes** de un vértice v son los vértices del nivel siguiente al de v que sean vecinos suyos (al vértice v se le dice **progenitor** de sus descendientes).
- Un vértice es **hoja** de un árbol con raíz si no tiene descendientes.
- Un árbol con raíz será q**-ario** si cada progenitor tiene exactamente q descendientes (es decir, el número de descendientes es 0 si el vértice es hoja y q si es progenitor). Será **casi** q**-ario** si el número de descendientes de cada vértice está comprendido entre 0 y q.

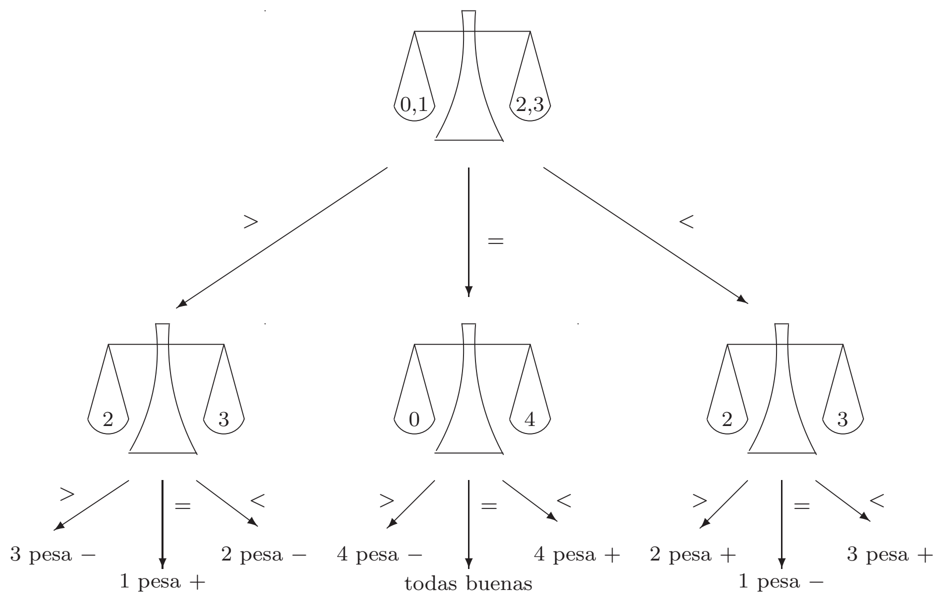
Así que los parámetros que manejaremos en un árbol con raíz serán

- el número de vértices, n;
- la altura del árbol, a;
- el número de hojas, h;
- y el tipo de árbol, definido por el entero positivo q (podrá ser q-ario o casi q-ario).

La importancia de estos árboles con raíz radica en que se utilizan, como ya hemos comentado, para representar algoritmos en los que intervienen operaciones binarias (o q-arias) sucesivas. Veamos algunos ejemplos.

Ejemplo 10.2.8 ***Un árbol de decisión***. *Tenemos 4 monedas,* $\{1,2,3,4\}$*, y, a lo sumo, una de ellas no tiene el peso correcto (no es legal). Disponemos además de una moneda patrón, la* 0*, con el peso correcto. Contamos además con una balanza, que tiene tres resultados posibles: o bien se inclina hacia un lado, o bien hacia el otro, o bien se mantiene en equilibrio. Queremos averiguar, de la manera más económica posible (con menos usos de la balanza), cuál es la moneda no legal.*

Podemos, desde luego, comparar sucesivamente la moneda patrón con las otras cuatro. En el peor de los casos, emplearemos cuatro pesadas para obtener la respuesta (aunque a veces la podamos obtener con menos). El siguiente esquema presenta un procedimiento alternativo:

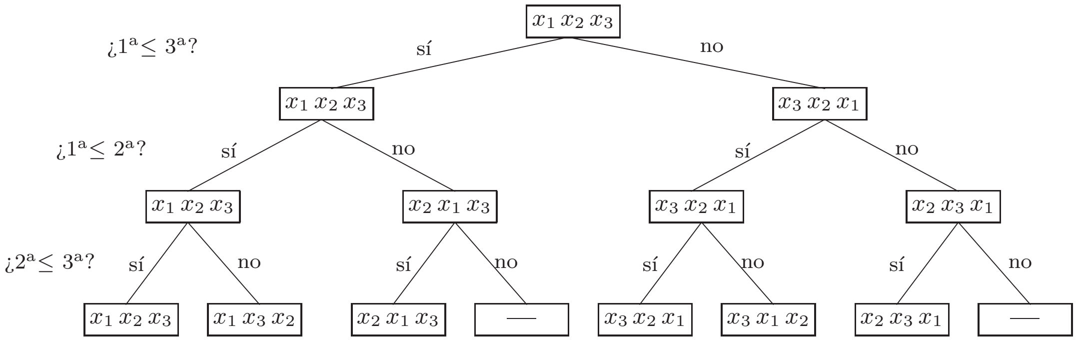


Los símbolos se explican por sí mismos: $<$ significa que la balanza se vence hacia la derecha, $>$ hacia la izquierda; y el símbolo $=$, que se queda equilibrada. Observemos que este procedimiento da la respuesta a nuestra pregunta con únicamente dos usos de la balanza.

El árbol que hemos diseñado es ternario ($q = 3$), tiene altura $a = 2$ (el número de pesadas) y número de hojas $h = 9$. Y la observación fundamental es que el algoritmo "funciona" porque el árbol tiene tantas hojas, nueve, como número posible de respuestas hay (1 pesa más, 2 pesa menos, etc). ♣

EJEMPLO 10.2.9 ***Un algoritmo de ordenación***. *Dada una lista de tres números,* (x_1, x_2, x_3), *¿qué número mínimo de comparaciones binarias requeriremos para ordenarlos?*

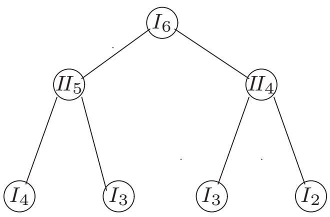
Diseñamos el siguiente algoritmo: en el primer paso, comprobamos si el primer elemento es menor o igual que el tercero. Si la respuesta es "sí", los dejamos tal como están; si es "no", los permutamos. En el segundo, miramos si el primero es menor o igual que el segundo, y de nuevo procedemos como antes, dependiendo de la respuesta. Por último, en el tercer paso investigamos si el segundo es menor o igual que el tercero (observamos que aquí hay respuestas que son incompatibles con las obtenidas anteriormente). Si representamos este proceso en un árbol, obtenemos



El árbol así construido es casi binario, con $a = 3$ y $h = 6$, que de nuevo coincide con el número de resultados posibles (las $3! = 6$ posibles ordenaciones). Podríamos preguntarnos si se podrían ordenar estos tres números, con comparaciones binarias, en menos pasos, por ejemplo dos. Veremos que no, porque en un árbol casi binario se cumplirá que $h \leq 2^a$. Así que, si $a = 2$, obtenemos que $h \leq 4$, y con ese número de hojas no podríamos cubrir todos los resultados posibles, que recordamos que eran 6. ♣

EJEMPLO 10.2.10 ***El juego de Nim con seis monedas***. *De un montón inicial de seis monedas, dos jugadores, que llamaremos I y II, van retirando, alternativamente, una o dos monedas del montón. Gana el jugador que retira las últimas monedas de la mesa.*

Podemos describir los posibles desarrollos de la partida con árboles: a cada configuración de monedas le corresponderá con un vértice, que irá etiquetado con I ó II (dependiendo de cuál sea el jugador al que le toca jugar) y con un número, para recordar cuántas monedas quedan en el montón. Por ejemplo, el primer vértice vendrá etiquetado con I_6, para recordar que juega I y que hay seis monedas en el montón.

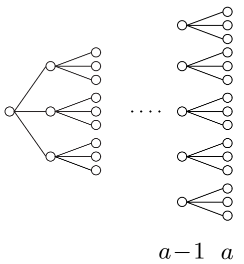


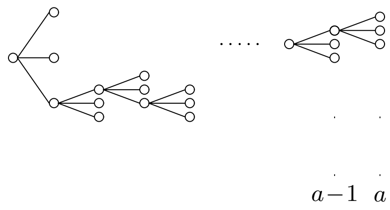
El jugador I tiene dos opciones, quitar una o dos monedas; estas dos acciones vendrán representadas por aristas: hacia la derecha si quita dos monedas, hacia la izquierda si quita una. Para cada una de estas elecciones, el jugador II encuentra dos configuraciones distintas y en cada una de ellas puede tomar, a su vez, dos decisiones. Véase a la izquierda el árbol hasta este punto. El lector podrá encontrar el árbol completo en el que se codifican *todas las posibles partidas* en la subsección 10.2.4, donde aprenderemos a analizar juegos de estas características utilizando representaciones arbóreas. ♣

A. Relación entre a y h en un árbol q-ario

En los ejemplos 10.2.9 y 10.2.8, la clave para que los algoritmos (representados por árboles q-arios, o casi q-arios) funcionaran era que el número de hojas cubriera todas las posibles respuestas. Pero estos dos parámetros, como el lector sospechará, dada la rigidez de este tipo de estructuras, no son independientes.

Supongamos fijados a y q. Queremos estimar el número de hojas que puede tener un árbol con raíz con esas características. Da la impresión de que la configuración con mayor número de hojas es aquélla en la que *todas* las hojas están en el último nivel. En el otro extremo, el árbol q-ario con menor número de hojas (para a fijo) sería aquél en el que las hojas van apareciendo lo antes posible. Los dos siguientes dibujos representan ambas situaciones:

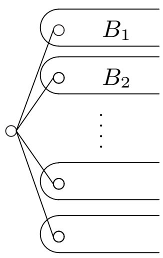




Esto nos sugiere, por un lado, que $h \le q^a$. Para la otra situación extrema, como en cada nivel, desde el 1 hasta el $a-1$, aparecen $q-1$ hojas nuevas y en el nivel a hay q hojas, sospechamos que $h \ge (a-1)(q-1)+q = (q-1)a+1$.

Proposición 10.11 *En todo árbol con raíz casi q-ario, $h \le q^a$.*

Demostración. Nótese que hemos relajado las condiciones sobre el tipo de árbol, basta con que sea casi q-ario (por supuesto, un árbol q-ario es también casi q-ario; pero no necesariamente al revés). Probaremos el resultado por inducción en a, la altura del árbol.



Si $a = 1$, es claro que $h \le q$. Sea entonces un árbol A casi q-ario con raíz y llamemos B_i a los árboles que tienen como raíz a los vecinos de la raíz del árbol A. Los B_i son más pequeños (en altura) que A y son todos árboles con raíz casi q-arios. La hipótesis de inducción nos diría que para los B_i se tiene, llamando a_i y h_i a la altura y el número de hojas de cada sub-árbol B_i, respectivamente, que

$$h_i \le q^{a_i} \quad \text{(la misma } q \text{ para todos).}$$

Nótese que hay a lo sumo q sub-árboles B_i, y que $a_i \le a-1$, para cada i (el peor caso correspondería a los sub-árboles que se extendieran hasta altura $a-1$). Así que

$$h = \sum_i h_i \le \sum_i q^{a_i} \le q q^{a-1} = q^a\,. \qquad \blacksquare$$

En el otro sentido, se puede probar la siguiente cota:

Proposición 10.12 *En todo árbol con raíz q-ario, $h = s\,(q-1)+1$, donde s es el número de vértices interiores del árbol (esto es, con descendientes).*

Como $a \geq s$ (porque en cada nivel, desde el 0 hasta el $a-1$, ha de haber al menos un vértice interior), deducimos que $h \geq a\,(q-1)+1$, como afirmábamos antes. La prueba de este resultado la dejamos como ejercicio (véase el ejercicio 11.4.1).

Ahora que disponemos de las cotas teóricas, en especial la que nos da la Proposición 10.11, podemos aplicarlas a los ejemplos que hemos presentado anteriormente.

Ejemplo 10.2.11 *Analicemos el problema de la balanza, pero ahora con r monedas (en lugar de cuatro), de las cuales a lo suma una es falsa, y la moneda patrón.*

El número de posibles resultados es $2r+1$, porque para cada moneda hay dos posibles (pesa más o menos de lo legal) y hay un resultado extra, que es que todas sean legales. Cualquier algoritmo que permita detectar la moneda falsa debe recoger todos estos resultados. Recordemos, por cierto, que con r pesadas siempre lo podemos hacer.

Si representamos uno de esos algoritmos con un árbol ternario, como corresponde a una balanza, la altura será el número máximo de pesadas necesarias para alcanzar todos los posibles resultados; y el número de hojas debe poder cubrir todos los resultados. Como $h \leq 3^a$, necesitaremos que se cumpla que

$$2r+1 \leq 3^a; \quad \text{esto es, que} \quad a \geq \log_3(2r+1).$$

Así que el número mínimo de usos de la balanza es del orden de $\log(r)$, un ahorro considerable frente al procedimiento trivial de las r pesadas. Por ejemplo, si $r=4$, como en el ejemplo que proponíamos, tendremos que

$$r=4 \quad \Longrightarrow \quad a \geq \log_3 9 = 2\,,$$

así que al menos se requieren dos pesadas (justo las del diseño del ejercicio 10.2.8). De la misma manera obtendríamos que para r entre 3 y 13 necesitaríamos al menos tres pesadas. Y que se requerirían cuatro pesadas, al menos, para cualquier r entre 14 y 40.

Ahora bien, conviene señalar que éstas son cotas teóricas y que cosa bien distinta es, por supuesto, diseñar un algoritmo que permita hacerlo con ese número mínimo de pesadas. ♣

Ejemplo 10.2.12 *Estudiemos el algoritmo para ordenar n números mediante comparaciones binarias.*

El número de resultados posibles es $n!$, así que necesitaremos que

$$2^{\#\,\text{pasos}} \geq n! \quad \text{en otras palabras,} \quad \#\text{ pasos} \geq \log_2(n!)\,.$$

Por ejemplo,

$$\begin{array}{llll}
\text{para } n=3\text{:} & \longrightarrow \;\#\text{ pasos} \geq \log_2(3!) = \log_2(6) \approx 2{,}585 & \Longrightarrow & \#\text{pasos} \geq 3; \\
\text{para } n=4\text{:} & \longrightarrow \;\#\text{pasos} \geq \log_2(4!) = \log_2(24) \approx 4{,}585 & \Longrightarrow & \#\text{pasos} \geq 5; \\
\text{para } n=5\text{:} & \longrightarrow \;\#\text{pasos} \geq \log_2(5!) = \log_2(120) \approx 6{,}907 & \Longrightarrow & \#\text{pasos} \geq 7.
\end{array}$$

El lector especialmente interesado en las estimaciones asintóticas precisas puede revisar la subsección 2.4.4 y comprobar que

$$\#\text{ pasos} \geq \log_2(n!) > \log_2\left(\left(\frac{n}{e}\right)^n\right) = n\,\log_2\left(\frac{n}{e}\right)\,.$$

♣

Ejemplo 10.2.13 *Contemos el número de hojas en el juego de Nim.*

Si el lector completa el árbol del juego de Nim con seis monedas (si no le viene bien en este momento hacer tamaño esfuerzo, puede consultar la subsección 10.2.4), descubrirá que el árbol tiene exactamente 13 hojas. Cada hoja representa un posible final del juego y, de hecho, a cada hoja le corresponde un desarrollo del juego distinto (el único paseo desde el vértice raíz a la hoja).

Si el juego de Nim parte de n monedas, la altura del árbol (que es casi-binario) es siempre n, porque el paseo de mayor longitud en el árbol es aquél en el que vamos quitando una sola moneda en cada turno (la rama de la izquierda), así que podemos estimar

$$\#\{\text{ posibles partidas en un Nim con } n \text{ monedas}\} = \#\{\text{hojas del árbol}\} \leq 2^n \,.$$

Pero, puesto a precisar, también podemos observar que el último nivel sólo tiene una hoja, así que podemos mejorar la estimación. Si llamamos P_n al número de hojas que tiene el árbol del juego de Nim con n monedas tendremos que

$$P_n \leq 2^{n-1} + 1 \,.$$

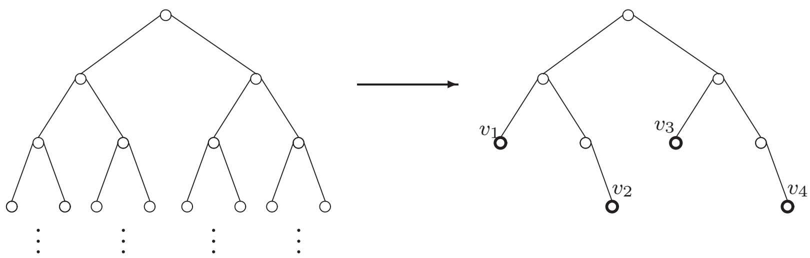
Aunque en este caso podemos calcular explícitamente el número de hojas que hay. Para ello, fijémonos en que un árbol de Nim con n monedas se compone de un árbol de Nim con $n-1$ monedas (el árbol que encontramos a partir del descendiente de la raíz que está a la izquierda) y otro con $n-2$ monedas (el de la derecha). De acuerdo, los papeles de I y II están intercambiados, pero eso no afecta al número de hojas que tenga cada árbol. Por supuesto, el número de hojas del árbol total es la suma del número de hojas que tenga cada subárbol, es decir,

$$P_n = P_{n-1} + P_{n-2} \quad n \geq 2 \,,$$

nuestra conocida ecuación de Fibonacci. Los valores iniciales son $P_1 = 1$ y $P_2 = 2$; así que el número de hojas de un juego de Nim con n monedas es justamente el número de Fibonacci F_{n+1} ($F_7 = 13$ hojas para el caso $n = 6$). ♣

B. La poda de un árbol binario

Vamos ahora a tratar una cuestión distinta. Tomemos *el* árbol binario (con raíz) *infinito*. En él vamos a seleccionar k vértices de manera que *ninguno de ellos sea antecesor de ningún otro*. Llamemos $v_1, \dots, v_k$ a estos vértices señalados, que estarán a alturas (generaciones) $h_1, \dots, h_k$. Observemos que esta elección de vértices se corresponde con una *poda* del árbol en la que las hojas superivientes están a alturas $h_1, \dots, h_k$. En la siguiente figura hemos escogido cuatro vértices, $v_1, \dots, v_4$:

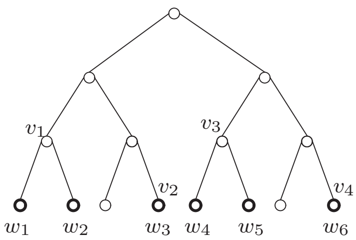


Exigir esta condición (ningún vértice ha de ser antecesor de otro) hace pensar que los vértices seleccionados no pueden estar todos muy “arriba” en el árbol, es decir, que los h_j no pueden ser arbitrariamente pequeños. De manera más precisa, y esto es lo que nos interesa comprobar, en una poda cualquiera, los h_j correspondientes han de cumplir que

$$\sum_{j=1}^{k} 2^{-h_j} \leq 1\,.$$

Veámoslo: hemos seleccionado unos vértices $v_1, \dots, v_k$ a alturas $h_1, \dots, h_k$. Llamemos $H = \text{máx}\{h_1, \dots, h_k\}$ a la mayor altura a la que encontramos alguno de los vértices señalados.

Vamos ahora, a partir de esta elección inicial, a seleccionar un conjunto de vértices en el árbol infinito con el siguiente procedimiento: tomamos el vértice v_1 y escogemos sus 2^{H-h_1} descendientes que están en la generación H (quizás solo el propio v_1, si es que es uno de los que vive en esa máxima generación). Esto lo hacemos, sucesivamente, con el resto de los vértices $v_2, \dots, v_k$. Nótese que los vértices que seleccionamos así son todos distintos.



En el dibujo que mostramos a la izquierda tenemos un ejemplo de este procedimiento. Los vértices escogidos en primer lugar son v_1, v_2, v_3 y v_4: dos de la segunda generación y otros dos de la tercera. La máxima altura es $H = 3$. En trazo más grueso hemos señalado los vértices de esta tercera generación que seleccionamos en segundo lugar, que nombramos como $w_1, \dots, w_6$. Si sumamos los números 2^{-h_j} para cada vértice v_j obtenemos $1/4+1/4+1/8+1/8 = 3/4$, un número no mayor que 1. El cálculo análogo, pero ahora para los vértices w_j seleccionados en segundo lugar, nos lleva a 6/8, el mismo resultado.

Llamemos A a la elección inicial de vértices y A' a la nueva, y sea $D(v_j)$ el conjunto de vértices descendientes de v_j que escogemos. Entonces

$$\sum_{v\in A'} 2^{-h(v)} = \sum_{j=1}^{k} \sum_{v\in D(v_j)} 2^{-h(v)} = \sum_{j=1}^{k} 2^{-H} \underbrace{|D(v_j)|}_{=2^{H-h_j}} = \sum_{j=1}^{k} 2^{-h_j} = \sum_{v\in A} 2^{-h(v)}\,.$$

El valor de la suma es el mismo para A que para A'. Pero claro, en A' hay, a lo sumo, 2^H vértices, así que

$$\sum_{j=1}^{k} 2^{-h_j} = \sum_{v\in A'} 2^{-h(v)} = |A'|\, 2^{-H} \leq 2^H\, 2^{-H} = 1\,.$$

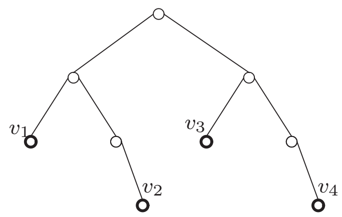
Lo interesante del asunto es que el resultado es cierto en el otro sentido: si damos unos números $h_1, \dots, h_k$ tales que $\sum 2^{-h_j} \leq 1$, entonces en el árbol binario (infinito) podemos señalar k vértices $v_1, \dots, v_k$ de alturas respectivas $h_1, \dots, h_k$ y de manera que ninguno sea antecesor de ningún otro. Llamamos, como antes, $H = \text{máx}\{h_1, \dots, h_k\}$. Y observamos que

$$1 \geq \sum_{j=1}^{k} 2^{-h_j} = 2^{-H} \sum_{j=1}^{k} 2^{H-h_j} \quad \Longrightarrow \quad 2^H \geq \sum_{j=1}^{k} 2^{H-h_j}\,.$$

Ahora, en el árbol binario infinito, en la generación H vamos señalando, de izquierda a derecha y sucesivamente, 2^{H-h_1} vértices, luego 2^{H-h_2}, etc.

La desigualdad anterior nos dice que en esa generación encontramos suficientes vértices como para completar el proceso. El paso final consiste en señalar a v_1 como el vértice de la generación h_1 que tiene a los primeros 2^{H-h_1} vértices seleccionados como descendientes en la generación H, a v_2 como el que tiene a los 2^{H-h_2} siguientes, etc.

El lector puede entretenerse, por ejemplo, siguiendo el procedimiento anterior para la lista de alturas $(2,3,4,4)$, que cumple las condiciones requeridas. Y. si lo desea, podrá también comprobar que el argumento funciona de manera análoga para el caso de un árbol q-ario general (sustituyendo el 2 por q en la desigualdad).

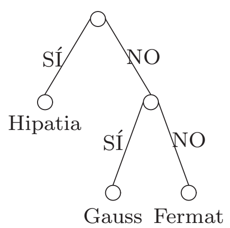


Vamos ahora a interpretar este resultado etiquetando con listas de ceros y unos los vértices seleccionados. Digamos que elegir, en el árbol binario infinito, la rama de la izquierda es un 1, y elegir la de la derecha un 0. Seleccionar un vértice de altura $h \geq 1$ no es, entonces, sino dar una lista de ceros y unos de longitud h. En el ejemplo que hemos estado considerando, al vértice v_1 le correspondería la lista $(1,1)$, al vértice v_2 la lista $(1,0,0)$, y a los vértices v_3 y v_4, las listas $(0,1)$ y $(0,0,0)$, respectivamente. Obsérvese que la elección de vértices con la restricción de que unos no sean antecesores de otros se traduce en que ninguna de las listas correspondientes puede coincidir con el comienzo de ninguna otra.

Más adelante (en la sección 16.2) llamaremos a esta restricción *condición de prefijo* y veremos que será fundamental a la hora de construir ciertos códigos. Veamos ahora otro ejemplo en el que esta condición de prefijo es relevante.

Ejemplo 10.2.14 *El problema de los cuestionarios.*

Tenemos k objetos que debemos identificar con una sucesión de preguntas cuya respuesta es un SÍ o un NO. El árbol asociado a las posibles listas de respuestas se obtiene seleccionando k vértices en el árbol binario infinito. Pero, para que el cuestionario permita realmente identificar los k objetos, los vértices deben cumplir que ninguno de ellos sea antecesor de algún otro vértice. O, si codificamos las sucesivas respuestas con una lista de ceros y unos, que todas las listas cumplan la condición de prefijo.



Por ejemplo, podríamos tener tres personajes, tres ilustres matemáticos como Hipatia de Alejandría[39], Gauss y Fermat. Preguntamos primero si el personaje es mujer, y posteriormente si es alemán. El resultado es el árbol que aparece a la izquierda, en el que se puede comprobar que no hay vértices que sean antecesores de otros. Con la identificación habitual de 1 para el SÍ y 0 para el NO, a Hipatia le corresponde la lista (1), a Gauss la $(0,1)$ y a Fermat la $(0,0)$. Las tres listas cumplen, como el lector puede observar la condición de prefijo. ♣

[39]Hypatia of Alejandría (370-415) es una de las pocas mujeres que se mencionan en los libros de Historia de las Matemáticas. Filósofa, matemática y astrónoma, y ahora popular gracias a la película "Ágora", iluminó un breve renacimiento del pensamiento clásico en su Alejandría natal, pocos años antes de que el Imperio Romano se desmoronara definitivamente y diera paso a unos cuantos siglos de completa oscuridad cultural. Las versiones sobre su muerte a manos de turbas cristianas varían de un historiador a otro, no así los detalles algo espeluznantes sobre el cómo se produjo, detalles que ahorraremos al lector.

10.2.4. Estrategias para ganar

En los sucesivos movimientos de una partida de un juego de estrategia, como el juego del ajedrez o el juego de Nim o las tres en raya, los jugadores han de elegir alternativamente entre las distintas opciones que se le presentan. Como ya hemos visto, los árboles, o más precisamente los árboles con raíz, permiten codificar eficientemente todas las alternativas sucesivas. El árbol codifica de golpe todas, en serio, todas, todas las partidas que pudieran darse. Recuerde el lector el ejemplo 10.2.10 del juego del Nim con seis monedas. En general, el número de partidas posibles, a poco que el juego tenga un mínimo de riqueza, será bien grande, y, correspondientemente, el árbol que lo representa deberá ser apropiadamente grande.

Una vez que disponemos del árbol del juego, podemos responder a preguntas tales como: ¿cuántas posibles partidas (diferentes sucesiones de movimientos) hay?; ¿cómo de largas son esas partidas? Se trata, en un caso, de contar el número de hojas del árbol, y en el otro, la longitud de los paseos que, sin repetir vértices, conectan la raíz con cada una de las hojas.

El lector comprensivo estará pensando, sin duda (¡no lo niegue, lector!), que esos ejercicios de aplicación de técnicas combinatorias están muy bien: nos permiten saber de antemano (aunque tras un meticuloso análisis) si nos enfrentamos a un juego potencialmente largo, o por el contrario a un juego que se resuelve en unas pocas jugadas. No está mal; información útil.

Figura 10.5: Zermelo

El lector competitivo nos espetará, sin embargo, que aunque hay quien dice que sólo busca entretenimiento, en realidad todo el mundo juega, a qué negarlo, para ganar; y que no nos gusta perder ni siquiera al parchís en una tarde de verano jugando contra los sobrinos. Clamará, elevando la voz, que los que afirman que lo importante es participar son unos perdedores, o unos hipócritas, o unos pardillos... En suma, el lector nos exigirá[40] que nos dejemos de zarandajas y nos demandará si esto de analizar los juegos con árboles sirve o no para ganar.

La demanda del lector es razonable. Hemos pergeñado un procedimiento para representar todas las partidas. Así que si, por ejemplo, el lector es el jugador que hace el primer movimiento, tendrá ahí delante tanto las partidas en las que ganaría como aquéllas en las que perdería. Ahí hay mucha información de la que se debería poder extraer, quizás con arduo esfuerzo, estrategias sobre cómo jugar. Nos ponemos a ello[41], sin más dilación, dóciles y sumisos ante las perentorias demandas del lector.

[40] Con buen tono y maneras, tras sosegarse, porque el lector será todo un ganador campeón, pero también un caballero o una señora, oiga.

[41] Fue el matemático alemán Ernst Zermelo (1871-1953), a quien además debemos gran parte de la fundamentación actual de las Matemáticas, quien desarrolló los argumentos que permiten analizar juegos como el ajedrez o el Nim. Zermelo renunció a su cátedra en Friburgo por oposición al nazismo. El húngaro John von Neumann (del que hablamos en la página 523) y el austriaco Oskar Morgenstern fueron los primeros en formalizar lo que hoy conocemos como Teoría de Juegos, en su clásico *Theory of Games and Economic Behavior* (1944). Otro gran nombre asociados a esta teoría es el de John Nash (1928-), tan de actualidad con motivo de la oscarizada película *Una mente maravillosa* (dirigida por Ron Howard en 2001 y protagonizada por Russell Crowe). Nash recibió el Premio Nobel de Economía en 1994 precisamente “por su pionero análisis del equilibrio en la teoría de los juegos no cooperativos”. Aunque quizás sus aportaciones matemáticas de mayor calado se encuentran en las ecuaciones en derivadas parciales y en el estudio de las variedades riemannianas.

A. ¡A jugar. . . !

Juegos hay de muchas clases, pero creemos que interpretamos en su justo punto los intereses del lector si nos centramos en los llamados juegos **estrictamente competitivos** entre **dos** jugadores. Este término, como cualquier otro en el contexto matemático, requiere una definición precisa, que daremos más adelante. Pero baste por el momento apuntar que significa que los intereses de los participantes en el juego son opuestos y que no habrá intención (en realidad, ni posibilidad) de colaborar entre ellos.

Supondremos además que los jugadores disponen en todo el juego de **información perfecta**; esto es, cuando un jugador tiene que mover, dispone (por ejemplo, sobre la mesa) de toda la información que necesita para tomar la decisión. Observe el lector que el mus no cumple esta propiedad, porque cada jugador desconoce las cartas de los demás.

Finalmente, exigimos que en el juego **no intervenga el azar**; esto descarta, por ejemplo, al parchís, porque los movimientos dependen del valor que obtenga al tirar unos dados[42].

El ajedrez, las damas, el juego del Nim o las tres en raya son juegos que reúnen estas tres características: son estrictamente competitivos, toda la información está sobre el tablero (o en las pilas o en la tabla) y no hay sorteos de ningún tipo. Para empezar, y fieles al espíritu competitivo que nos inflama, estudiaremos juegos en los que únicamente pueden darse dos resultados: **victoria o derrota**.

B. El análisis sobre el árbol

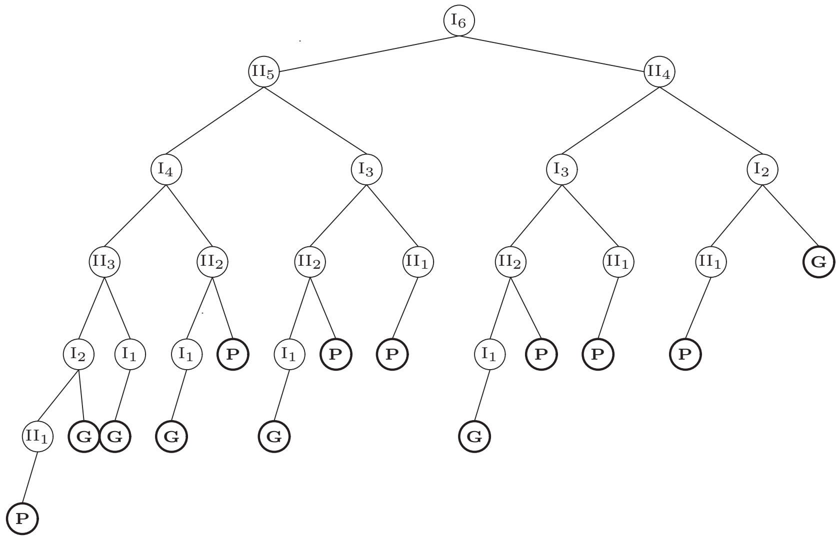
Nuestro análisis comienza traduciendo el juego (sus reglas y posiciones) en un árbol con raíz. El vértice raíz recoge la configuración inicial. Los vértices del primer nivel recogen las configuraciones posibles del juego tras la primera jugada. Así que las aristas que conectan la raíz con los vertices del primer nivel recogen las posibles jugadas que puede hacer el primer jugador (jugador I). En cada vértice del primer nivel tiene turno el segundo jugador, II. Sus jugadas posibles en ese momento conectan vértices del primer nivel con sus descendientes del segundo nivel. Así, sucesivamente, se construye el árbol.

En cada vértice anotamos la configuración del juego en ese punto y etiquetamos con el nombre (I ó II) de quien le toca jugar. Las aristas serán las posibles elecciones o jugadas que tiene el jugador al que le toque mover; las etiquetaremos con el nombre de la jugada que realiza. Las hojas representarán los posibles finales del juego y las etiquetaremos con los posibles resultados del juego desde la perspectiva del *primer jugador*: G (gana) ó P (pierde). Una partida será un paseo que no repita vértices desde la raíz hasta una hoja.

Completamos ahora el árbol del juego del Nim con seis monedas, o Nim-6 (véase el ejemplo 10.2.10), en el que recogemos todas las partidas posibles con el sistema de etiquetas anterior. Por ejemplo, una etiqueta I_3 en uno de los vértices significa que juega el jugador I y que hay tres fichas en el montón. Para simplificar el dibujo, no ponemos etiquetas en las aristas, pero convenimos en que las aristas hacia la derecha representan la acción de quitar *dos* monedas, y las que van hacia la izquierda, quitar *una*.

[42]El lector perspicaz argumentará que en cualquier juego pueden producirse accidentes azarosos, como despistes o errores. Concedido, pero no es a ese azar al que nos referimos, sino al azar como mecanismo para generar nueva información en el juego.

A las hojas las diferenciamos de los otros vértices con trazo más oscuro y con un símbolo que representa si el jugador I gana (G) o pierde (P) en ese momento final de la partida.



Ya tenemos toda la información sobre el juego recogida en el árbol. Es hora de comenzar su análisis. Nuestro objetivo es investigar las posibles estrategias para (intentar) ganar. Una estrategia para un jugador es una prescripción de qué hacer en aquellos vértices en los que le corresponda turno. En el Nim, una estrategia del jugador I le dicta si escoger izquierda o derecha en cada vértice etiquetado con I, es decir, en los vértices de los de los niveles, 0, 2, 4. . . El jugador I quiere ir escogiendo izquierda o derecha para acabar en una hoja etiquetada con G. Por supuesto, al jugador II le gustaría ir escogiendo, en sus turnos, izquierda o derecha para intentar acabar en una hoja etiquetada con P (que significa victoria para él).

Ahora vamos a *(re)etiquetar* el resto de los vértices del árbol con los símbolos G ó P (siempre desde el punto de vista de I). Lo haremos de abajo hacia arriba, desde los potenciales finales hacia el comienzo, partiendo de las hojas (ya etiquetadas) y luego utilizando las etiquetas que vayamos asignando. Bien podría el lector repetir, en hoja aparte, el dibujo del árbol e ir colocando etiquetas siguiendo las explicaciones que damos a continuación.

Empezamos con el vértice II_1 de la penúltima generación. Ahí le toca mover al jugador II y sólo queda una ficha. La única posibilidad es que II la retire y, por tanto, gane el juego. Así que es razonable asignar P a ese vértice (pues gana II). Ahora todos los vértices de la penúltima generación van distinguidos con G ó P.

Nos vamos a la generación anterior (en la que le toca mover a I), empezando con el vértice I_2 más a la izquierda (según mira el lector la página). Quedan dos fichas y hay dos opciones para I: retirar una ficha (arista hacia la izquierda), dejando una configuración II_1 (que acabamos de etiquetar con P), o bien retirar dos (arista hacia la derecha) para ganar el juego. Si, como hemos dado por supuesto, el carácter de I es ganador, elegirá la línea de juego que le lleva a la victoria, que en este caso consiste en retirar dos fichas de la pila. Así que deberemos etiquetar este vértice con G, pues, llegados a esta situación de la partida, el jugador I puede ganar (y el propio árbol nos dice cómo hacerlo). Los otros cuatro vértices de esta generación corresponden a una configuración I_1, a la que el lector asignará sin dudar la etiqueta G.

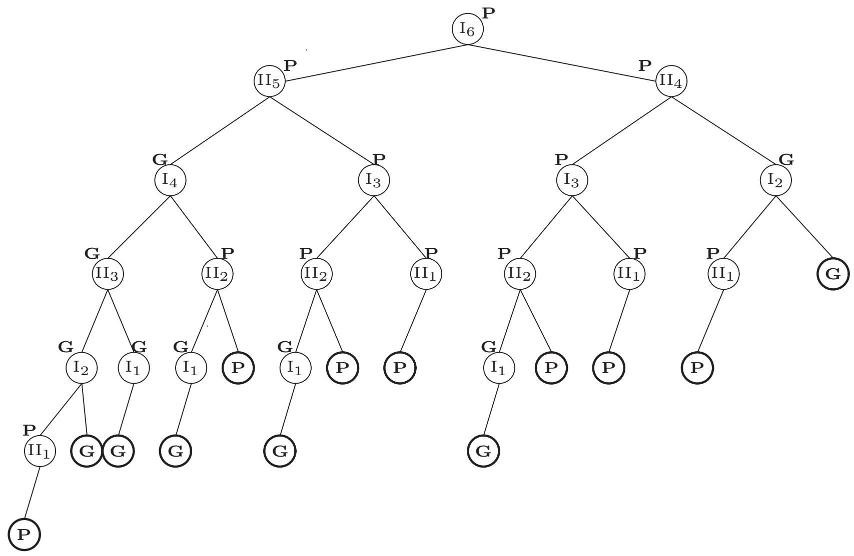
En los vértices de la generación anterior ha de jugar II. Uno de ellos es un II_3, y sus dos descendientes ya han sido etiquetados (ambos) con G. Así que II sabe que, haga lo que haga, llegará a un vértice en los que, como ya se ha argumentado antes, I tiene líneas de juego ganadoras. No le quedará más que resignarse y aceptar que ese vértice debe recibir la etiqueta G. Los vértices que llevan II_2, sin embargo, tienen un descendiente G y uno P, y el jugador II elegirá, sin duda, la línea de juego que le lleva al P. Por últimos, los vértices II_1 son, como ya hemos visto antes, directamente ganadores para II, y por tanto llevarán P.

Para etiquetar los vértices de la generación anterior (en los que mueve I), el lector encontrará situaciones ya vistas: en concreto, vértices I_4 y vértices I_2, que tienen un descendiente P y otro G (que nos lleva a etiquetar los I_2 y los I_4 con G). Y una situación nueva: los vértices I_3, cuyos descendientes son de tipo P; así que… P para estos vértices.

Ya está en disposición el lector de completar el (re)etiquetado completo del árbol, pues en los argumentos anteriores ya nos hemos encontrado con las cuatro posibilidades que resumimos a continuación, para que sirvan de oportuna referencia para las momentáneas dudas que pudieran surgir en el proceso. Estas cuatro reglas conforman el conocido como **algoritmo de Zermelo**, que parte de un árbol (de un juego) cuyas *hojas* están etiquetadas con G ó P dependiendo de si en ellas el jugador I gana o pierde:

1. Si *todos* los descendientes de un vértice II están etiquetados con G, etiquetamos ese vértice con G (observemos que en esta situación, haga lo que haga II, el jugador I gana).
2. Si *todos* los descendientes de un vértice I están etiquetados con P, etiquetamos ese vértice con P. Observemos que, en esta situación, nada puede hacer I para ganar.
3. Si *algún* descendiente de un vértice II está etiquetado con P, etiquetamos ese vértice con P (en esta situación, I nada puede hacer contra el mejor juego de II).
4. Si *algún* descendiente de un vértice I está etiquetado con G, etiquetamos ese vértice con G (porque I puede elegir una línea de juego en la que gana).

Aplicando estas instrucciones al árbol del Nim-6, obtenemos lo siguiente:



La conclusión es que el vértice raíz, que se corresponde con la configuración inicial del Nim-6, acaba con una etiqueta P (diremos más adelante que el juego tiene un *valor*, en concreto P). Lo que significa que existe una línea de juego "ganadora" para el jugador II (haga lo que haga I). Así que siempre podremos ganar en el juego de Nim-6 ... ¡si nos dejan mover en segundo lugar! Observemos además que cada vértice interior del árbol determina un subjuego: el árbol que tiene a ese vértice como raíz y todo lo que le sigue. Cada uno de estos subjuegos ha sido también etiquetado (con G ó P) durante el proceso.

Pero si aceptamos que el juego es ganador para II, nos gustaría saber además *cómo* debe mover para ganar. Esto supone prescribir una *estrategia*, en la que se especifique la acción de un jugador (en este caso, II) en cada configuración en la que se juegue. Insistimos en que ha de ser una *especificación completa*, una regla que dice qué arista va a escoger en cada vértice en el que le toque jugar. Esta estrategia ganadora (para II) se puede leer directamente del árbol anterior: si el primer jugador retira una ficha, entonces II retira dos: entra así en un subjuego que tiene etiqueta P y en el que habrá una estrategia ganadora. Si, por el contrario, el jugador I retira dos fichas en el primer movimiento, entonces II ha de retirar una sólo, para de nuevo adentrarse en un subjuego a su favor.

El árbol anterior también contiene la información sobre juegos de Nim más pequeños. Si el lector analiza los subjuegos que en él aparecen (prestando la debida atención a quién es el "primer" jugador en cada uno de ellos), concluirá que el jugador que empieza gana en los Nim con 1, 2, 4 y 5 fichas, mientras que pierde en el Nim-3 (véase también el ejercicio 10.2.13).

El siguiente resultado despejará las dudas del aquéllos lectores que aún estén preocupados por la posibilidad de que este algoritmo de etiquetado no funcione *siempre*:

Lema 10.13 *Consideremos un árbol finito cuyas* ***hojas*** *están etiquetadas con G ó P. Entonces, el algoritmo de etiquetado hacia atrás descrito anteriormente produce con un etiquetado de los vértices del árbol que es* ***completo*** *(todos los vértices se etiquetan) y* ***consistente*** *(ningún vértice recibe dos etiquetas distintas).*

Demostración. La haremos por inducción en h, la altura del árbol. Si $h = 0$, sólo tenemos la raíz, que ya está etiquetada. Supongamos entonces que el lema es cierto para cualquier árbol de altura $h = n$. Consideremos ahora un árbol de altura $n + 1$, con hojas etiquetadas con los símbolos G ó P, y sea v un vértice con altura n. Si v es hoja, ya estará etiquetado. Si, por el contrario, no es hoja, entonces tiene algún descendiente en el nivel $n + 1$; todos ellos serán hojas y, por tanto, estarán etiquetadas. El vértice v puede ser del tipo I o del tipo II:

- Si v es del tipo II, entonces: si todos sus descendientes llevan G, lo etiquetamos con G. Y si alguno lleva P, lo etiquetamos con P.
- Si es de tipo I, entonces. o bien todos sus descendientes llevan P, en cuyo caso etiquetamos con P; o bien alguno lleva G, y entonces lo etiquetamos con G.

Como cubrimos todas las posibilidades, conseguimos etiquetar con G ó P todos los vértices del nivel n que no estuvieran ya etiquetados (por ser hojas del árbol). Borramos ahora todas las hojas del nivel $n+1$ y obtenemos un árbol de altura n con *todas* las hojas etiquetadas. La hipótesis de inducción nos permite completar el etiquetado del resto de los vértices de este árbol. Al recuperar las hojas del nivel $n + 1$, obtenemos el etiquetado total. ■

Recapitulemos. Sea cual sea el juego finito[43] representado con su árbol con raíz, si clasificamos las hojas como G o P podemos, mediante el algoritmo de Zermelo, el vértice raíz queda etiquetado como G o P. Es decir, o bien el jugador I tiene una estrategia para ganar o bien el jugador II tiene estratega para ganar.

Pero, ¡espere!, dirá el lector: ¿todo este trabajo para saber que o bien gana el primer jugador o bien gana el segundo? No, no es esto[44]. Lo que hemos probado es que o bien el primer jugador tiene una estrategia para ganar *siempre*, haga lo que haga el segundo jugador, o bien el segundo jugador dispone de una estrategia para ganar *siempre*, haga lo que haga el primer jugador. ¡Ah!, esto es distinto, reconocerá el lector. En el Nim-6, el segundo jugador ganará siempre, si se atiene a la estrategia que se destila de analizar el árbol.

Y esto es general, ocurre en todos los juegos finitos (estrictamente competitivos, con información completa y sin azar) que acaban con victoria o derrota, sin empates. En *todos los juegos*, por muy largos que sean. O hay estrategia para que gane el primer jugador o hay estrategia para que gane el segundo. En un juego relativamente sencillo como el del Nim, además, podemos decir quién de los dos tiene esa estrategia ganadora: si por ejemplo hay 213 monedas, el segundo jugador tiene estrategia ganadora; pero si hay 214 ó 215, es el primer jugador quien tiene la estrategia.

Los juegos con muchas alternativas en cada situación, como el ajedrez, que analizaremos en un momento, dan lugar a árboles muy ramificados, inmensos[45], que no podemos etiquetar hoy por hoy con los más potentes ordenadores. Pero, en principio, esto es una cuestión de tiempo y de espacio. En abstracto, matemáticamente hablando, todo juego (que se ajusta a las condiciones anteriores) se *puede* analizar de manera completa, precisa y determinada.

C. ¿Y si cupiera también el empate?

El análisis de otros juegos interesantes exige considerar más resultados. En el ajedrez, sin ir más lejos, debemos considerar G (el jugador I gana), P (I pierde) y E (empatan). El algoritmo de Zermelo nos permite también abordar esta situación. Considere el lector el árbol de un juego de éstos con las hojas etiquetadas con G, E, y P.

Reetiquetemos las hojas. Ponemos G si había una G y ponemos la etiqueta (nueva) NG si había E ó P. Ahora sólo hay dos etiquetas. El algoritmo de Zermelo nos dice que, o bien hay estrategia para que I fuerce G, o bien para que II fuerce NG. Una u otra.

Un reetiquetado alternativo. Ponemos P si había P y ponemos NP si la etiqueta era G ó E. De nuevo, como sólo hay dos etiquetas, el algoritmo de Zermelo nos dice que o bien hay estrategia para que II fuerce P o bien para que I fuerce NP. Una u otra.

[43] Aquí siempre consideraremos juegos finitos. En el caso del ajedrez, o de las damas, esto supone incluir algún protocolo en el que se declaren tablas en caso de que no se pueda alcanzar la victoria, o que se limite el número de movimientos.

[44] Que clamaría Ortega y secundaría Gasset.

[45] Obsérvese que los vértices del árbol del ajedrez deberían ir etiquetados con el jugador que se dispone a mover, junto con una "foto" del tablero que recoja la posición de todas las fichas. Del vértice raíz parten ya 16 aristas (las posibles maneras de mover los peones blancos) a otros tantos vértices, y de cada uno de estos, otras 16 aristas a vértices nuevos (los correspondientes movimientos de peones negros). Y en la siguiente generación. . . ¡uff!, dejamos al lector que se entretenga (bastante tiempo) contando posibilidades.

Tanto análisis nos ha llevado a discutir cuatro estrategias. Obsérvese que una de las dos estrategias fruto del primer etiquetado ha de existir y *también* existe una de las que han surgido en el segundo reetiquetado. Desde luego, si I tiene estrategia para forzar G, la conclusión es clara. Lo mismo sucede si II tiene estrategia para forzar P.

Nos queda el caso en que ninguna de estas dos estrategias tan definitivas existe. Entonces, el análisis anterior nos lleva a concluir que II tiene una estrategia que, haga lo que haga I, lleva a una hoja etiquetada con NG, *al tiempo* que I tiene una estrategia que, haga lo que haga II, conduce a una hoja con NP. Usando cada uno la suya, la competitividad de I y II conducirá irremediablemente a una hoja etiquetada a la vez con NG y NP; es decir, con E.

Conclusión: o bien I tiene una estrategia con la siempre gana, o bien II dispone de una estrategia con la que siempre gana; o bien I y II disponen de sendas estrategias de manera que todas las partidas acaban en empate.

Lo que resta es averiguar, para cada juego en particular, cuál de las tres situaciones es la que sucede en realidad. Para un juego como el ajedrez, como ya hemos comentado antes, el análisis en el árbol queda más allá de la potencia computacional de los ordenadores actuales. . . ¡tranquilícense (por ahora) los practicantes de este noble arte! Pero la conclusión alcanzada de que siempre deben ganar blancas, o siempre deben ganar negras, o siempre debe haber empate, es asombrosa, ¿no?

D. Valores

Envalentonados por el resultado anterior, e imbuidos de vocación de investigadores del abstracto matemático, nos preguntamos ahora qué pasa si hay más resultados. Pongamos que hay 10 resultados, que numeramos de 10 a 1 en orden de preferencia e interés para I. El jugador II prefiere, por el contrario, un 1 a un 2, y éste a un 3, y así hasta el 10, que desde la óptica de II es el peor resultado (los juegos de antes, con tres posibles resultados G, E y P, se codificarían ahora con 3, 2, 1).

Siguiendo el esquema del análisis con tres resultados, reetiquetamos primero con 10 las hojas que tienen un 10 y con el nuevo símbolo "9–1" las que tienen cualquiera de los demás, de 1 a 9. Si hay estrategia para que I fuerce 10, tenemos una conclusión clara. Si resulta que, por el contrario, es II quien tiene estrategia para forzar "9–1", seguimos analizando.

Reetiquetamos ahora con "10–9" las hojas que tienen 10 ó 9 y con "8–1" las que tienen cualquier resultado desde 8 a 1. Si I tiene estrategia para forzar "10–9", también tenemos una conclusión clara. ¿Sí? Veamos. Recuerde que como II tiene estrategia para forzar "9–1", la combinación competitiva de las estrategias de I y II forzará un resultado de . . . 9, siempre.

Complete el lector el análisis. La conclusión es que, o bien I tiene estrategia para forzar 10, o bien II tiene estrategia para forzar 1; o bien hay un valor v (entre 2 y 9) de manera que I y II tienen estrategias tales que el resultado acaba siempre en v. Resumiendo, no sabemos qué situación se dará, pero hay un *único* valor v entre 1 y 10 de manera que todas las partidas *deben* acabar en ese valor v. A ese v, cómo no, se le llama **valor** del juego. Todo juego, de las características consideradas tiene un valor, aunque no sabemos, en general, cuál.

Acabamos aquí esta discusión, no sin antes animar al lector proclive a la formalización a que revise los ejercicios 10.2.14 y 10.2.15. Y, ¿por qué no?, a que analice con esta tecnología su juego favorito.

EJERCICIOS DE LA SECCIÓN 10.2

10.2.1 *Pruébese que un grafo G es un árbol si y sólo si no tiene ciclos y $|A(G)| = |V(G)| - 1$.*

10.2.2 *Pruébese que, en un árbol, y dados dos vértices suyos cualesquiera, existe un único paseo que no repite vértices que los conecta.*

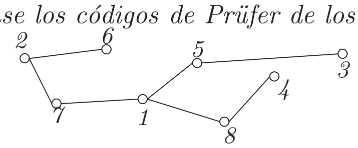
10.2.3 *Si G es árbol con p vértices de grado 1, q vértices de grado 4 y ningún otro vértice, ¿qué relación hay entre p y q? ¿Hay árboles con esos grados?*

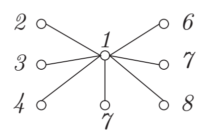
10.2.4 *Denotemos por $N(d_1, d_2, \dots, d_n)$ el número de árboles distintos que se pueden formar con el conjunto de vértices $\{1, 2, \dots, n\}$, donde $gr(j) = d_j + 1$.*
(a) Obsérvese que si $\sum_{j=1}^n d_j \neq n-2$ entonces $N(d_1, d_2, \dots, d_n) = 0$.
(b) Pruébese la siguiente fórmula de recurrencia:

$$N(d_1, d_2, \dots, d_{n-1}, 0) = N(d_1-1, d_2, \dots, d_{n-1}) + N(d_1, d_2-1, \dots, d_{n-1}) + \cdots + N(d_1, d_2, \dots, d_{n-1}-1),$$

donde en la suma anterior el término i-ésimo no aparece si $d_i = 0$.
(c) Dedúzcase que si $\sum_{j=1}^n d_j = n-2$, entonces $N(d_1, d_2, \dots, d_n) = \binom{n-2}{d_1, d_2, \dots, d_n}$.
(d) Dedúzcase finalmente la fórmula de Cayley.

10.2.5 *Sea T un árbol con n vértices etiquetados con los números $\{1, \dots, n\}$. Localizamos el vértice de grado 1 con menor etiqueta, que llamamos b_1, y anotamos quién es su único vecino, a_1. Borramos entonces b_1 y su arista. En el nuevo árbol, repetimos el procedimiento: localizamos el vértice de grado 1 con menor etiqueta, b_2, anotamos quién es su vecino a_2, y borramos b_2 y su arista. Y así, sucesivamente, hasta quedarnos con sólo un vértice. Lo que hemos ido anotando forman una lista $(a_1, \dots, a_{n-1})$, el* ***código de Prüfer*** *del árbol T. Esta lista puede tener símbolos repetidos y $a_{n-1} = n$.*
(a) Hállense los códigos de Prüfer de los árboles siguientes:





(b) Compruébese que, a partir del código de Prüfer $(a_1, \dots, a_{n-1})$ de un cierto árbol, podemos recuperar el árbol en sí. Es decir, la sucesión $(b_1, \dots, b_{n-1})$ de los vértices borrados por el algoritmo.
(c) Demuéstrese que, dada una lista de números $(a_1, \dots, a_{n-1})$, con $1 \leq a_j \leq n$ para cada $j = 1, \dots, n-2$ y $a_{n-1} = n$, existe un único árbol cuyo código de Prüfer es, precisamente, esa lista.
(d) Dedúzcase la fórmula de Cayley.

10.2.6 *a) Hállese el número de árboles distintos que se pueden formar con los vértices $\{1, 2, \dots, n\}$ si (i) $n = 6$ y cuatro vértices tienen grado 2; (ii) $n = 5$ y exactamente tres vértices tienen grado 1.*
b) ¿Cuántos árboles distintos se pueden formar con un conjunto de ocho vértices, $\{1, 2, 3, \dots, 8\}$, de manera que 2 de los vértices tengan grado 4 y los 6 restantes tengan grado 1?

10.2.7 *¿Cuántos árboles con n vértices tienen los grados de todos sus vértices ≤ 2?*

10.2.8 *Sean G y H dos grafos con conjunto de vértices $\{1, \dots, n\}$. Compruébese que si $A(G) \subseteq A(H)$, entonces el H tiene, al menos, tantos árboles abarcadores como G.*

10.2.9 *Calcúlese el número de árboles abarcadores distintos de un grafo isomorfo a $K_{3,s}$.*

10.2.10 *Consideremos el grafo que se obtiene al tomar n triángulos con exactamente un vértice común (hay $2n+1$ vértices y $3n$ aristas) ¿Cuántos árboles abarcadores tiene?*

10.2.11 *Consideremos un grafo G y llamemos $\tau(G)$ al número de árboles abarcadores que tiene. Fijémonos en una arista a del grafo G. Llamemos $G \setminus \{a\}$ al grafo que queda al eliminar esa arista y G_a al que se obtiene al identificar los vértices extremos de la arista a. Compruébese que*

$$\tau(G) = \tau(G \setminus \{a\}) + \tau(G_a)$$

(puede el lector puede consultar la discusión sobre el cálculo del polinomio cromático de un grafo de la sección 11.3.3 para entender la manera de utilizar esta fórmula como regla de cálculo de $\tau(G)$).

10.2.12 *Queremos calcular el determinante de dimensiones $(n-2)\times(n-2)$ que aparece debajo de estas líneas, más a la izquierda. Las habituales operaciones de eliminación gaussiana no cambian el valor de este determinante. Sustitúyase cada columna, de la segunda a la última, por la resta de ella misma con la primera para el paso (1) y sáquense factores comunes para el paso (2):*

$$\begin{vmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & n-1 & -1 & \cdots & -1 \\ -1 & -1 & n-1 & \cdots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & \cdots & n-1 \end{vmatrix} \overset{(1)}{=} \begin{vmatrix} n-1 & -n & -n & \cdots & -n \\ -1 & n & 0 & \cdots & 0 \\ -1 & 0 & n & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \cdots & n \end{vmatrix} \overset{(2)}{=} n^{n-2} \begin{vmatrix} n-1 & -1 & -1 & \cdots & -1 \\ -1 & 1 & 0 & \cdots & 0 \\ -1 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \cdots & 1 \end{vmatrix}$$

Finalmente, súmesele a la primera columna las restantes para comprobar que el determinante más a la derecha vale exactamente 1.

Juegos

10.2.13 *Dos jugadores, I y II, disputan una partida en un juego del Nim (en cada turno se retiran una o dos fichas del montón, y gana quien se lleve las últimas) en el que partimos de un montón con n fichas. ¿Cuál es el valor del juego, desde el punto de vista del jugador I?*

10.2.14 *En un cierto juego (con información perfecta y sin azar) se pueden producir los siguientes resultados: $\{r_1, \dots, r_k\}$. Podemos suponer que el jugador I no queda indiferente ante ningún par de posibles resultados. Digamos que su orden de preferencias sobre todos los posibles resultados es el siguiente:*

$$r_1 \prec_1 r_2 \prec_1 r_3 \cdots \prec_1 r_k\,,$$

*donde el símbolo $r_i \prec_1 r_j$ significa que el jugador I prefiere el resultado r_j antes que el r_i. Para el jugador II utilizaremos el símbolo $\prec_2$. Diremos que el juego es **estrictamente competitivo** si*

$$r_i \prec_1 r_j \quad \Longleftrightarrow \quad r_j \prec_2 r_i$$

para cada par de posibles resultados r_i y r_j. Es decir, los intereses de los jugadores son contrapuestos (lo bueno para uno es malo para el otro). Esto supone que las preferencias de II quedan ordenadas de la siguiente forma:

$$r_1 \succ_2 r_2 \succ_2 r_3 \cdots \succ_2 r_k\,.$$

*Pruébese el **teorema de Zermelo**: sea A un conjunto de posibles resultados de un juego finito estrictamente competitivo entre dos jugadores, con información perfecta en cada paso y en el que no hay movimientos que dependan del azar. Entonces, o bien el jugador I puede forzar un resultado de los del conjunto A, o bien el jugador II puede forzar uno en el conjunto A^c (el complementario de A en el conjunto de posibles resultados del juego).*

10.2.15 *Un resultado r es el **valor** de un juego finito estrictamente competitivo entre dos jugadores si y sólo si el jugador I puede forzar un resultado en el conjunto $G_r = \{s : s \succeq_1 r\}$ y el jugador II puede forzarlo en el conjunto $P_r = \{s : s \succeq_2 r\}$. Pruébese que cualquier juego finito, estrictamente competitivo entre dos jugadores con información perfecta en cada paso y sin azar tiene un **valor**.*

10.3. Grafos dirigidos

En muchas de las aplicaciones del lenguaje de los grafos, resulta natural asociar un *sentido*[46] u *orientación* a cada una de las aristas. Piense el lector en una red de calles de una ciudad, algunas de las cuales tienen un único sentido de circulación, mientras que por otras se puede circular en los dos sentidos. O quizás en Internet, que podemos representar con un enorme grafo en el que los vértices son las distintas páginas *web* y las aristas los enlaces entre unas y otras (enlaces que, por ejemplo, podrían ir de una página a otra, pero no al revés).

Esta sección está dedicada a la formalización de la noción de grafo dirigido (o digrafo) y a describir sus similitudes y diferencias con los grafos habituales. Se trata de la generalización del concepto de grafo que se obtiene si asociamos un sentido a una arista:

en lugar de $u \underset{a}{\text{———}} v$ consideramos $u \xrightarrow[a]{\quad} v$ o bien $u \xleftarrow[a]{\quad} v$

(¡o quizás incluyamos ambas aristas!) Observe el lector que dar un sentido a una arista requiere decidir qué extremo es el vértice inicial, es decir, *ordenar* el par de vértices.

Definición 10.13 *Un **grafo dirigido** $D = (V, A)$ está formado por un conjunto V de vértices y un conjunto A de aristas extraído de la colección de **pares ordenados** formados con los elementos de V. Una arista de D es, pues, una lista (a, b), con $a, b \in V$, $a \neq b$.*

Note el lector la diferencia con la definición 10.1 de la noción habitual de grafo. Aquí, las aristas son *pares ordenados*, es decir, listas, en lugar de los 2-subconjuntos habituales.

Para contar el número de grafos dirigidos (sin lazos) que hay con vértices $V = \{1, 2, \dots, n\}$, observamos primero que hay tantos pares ordenados con los n vértices como

$$\#\,\{\text{listas sin repetición de longitud 2 extraídas de } \{1, \dots, n\}\} = n(n-1)\,.$$

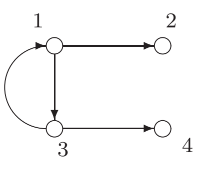
Así que el número de grafos dirigidos (sin lazos) con V como conjunto de vértices es igual a

$$\#\,\{\text{subconjuntos de } \{1, \dots, n(n-1)\}\} = 2^{n(n-1)}\,.$$

Muchos más, claro, que los $2^{\binom{n}{2}}$ grafos sin dirigir (una cantidad es el cuadrado de la otra). Si permitiéramos lazos (diremos que un lazo admite una única orientación), el lector podrá comprobar que habría 2^{n^2} grafos dirigidos (compárese con el resultado del ejercicio 10.1.1).

Si, por ejemplo, partiéramos de $V = \{1, 2, 3, 4\}$, el conjunto de posibles aristas sería

$$P = \{(1,2), (1,3), (1,4), (2,1), (2,3), (2,4), (3,1), (3,2), (3,4), (4,1), (4,2), (4,3)\}.$$



El de posibles lazos, si es que interesara considerarlos, sería $L = \{(1,1), (2,2), (3,3), (4,4)\}$. Para formar un grafo dirigido tomamos algunos elementos de P (si además permitiéramos lazos, algunos de L). La elección de aristas $A(D) = \{(1,2), (1,3), (3,1), (3,4)\}$ da lugar al grafo dirigido que aparece a la izquierda.

[46]Recuerde el lector que en castellano es habitual usar los términos "dirección" y "sentido" indistintamente. Y aunque "sentido" es más correcto, en algunas ocasiones utilizaremos también el otro. Por ejemplo, en el propio nombre de los objetos que estamos analizando, "grafos dirigidos". Pero es que, de lo contrario, caeríamos en la tentación de hablar de "grafos con y sin sentido"... ¡y eso ya no!

	v_1	v_2	v_3	$\cdots$	v_n
v_1	0	1	1	$\cdots$	1
v_2	0	0	1	$\cdots$	1
v_3	0	1	0	$\cdots$	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
v_n	0	0	0	$\cdots$	0

Si hay n vértices, digamos $\{v_1, \dots, v_n\}$, podremos representar cada grafo dirigido por su *matriz de vecindades*, la matriz de dimensiones $n \times n$ en la que: (1) en las posiciones (v_i, v_i), para cada $i = 1, \dots, n$, aparecen ceros (si permitiéramos lazos, podrían aparecer unos); y (2) en la posición (v_i, v_j), con $i \neq j$, ponemos un 1 si $(v_i, v_j) \in A$, y un 0 en caso contrario. Observemos que, como en el ejemplo que mostramos a la izquierda, el símbolo de la posición (v_i, v_j) no tiene por qué coincidir con el de la posición (v_j, v_i).

Así que los grafos dirigidos (con lazos) son una forma de representar relaciones binarias *generales* (revísese la sección 3.5). Ésta es una de las razones que justifica su estudio. Recordemos que los grafos habituales se corresponden con relaciones *simétricas*. Permítanos el lector insistir sobre el asunto en el siguiente esquema:

Grafo simple (sin lazos)	$\Longleftrightarrow$	Matriz simétrica de ceros y unos con ceros en la diagonal	$\Longleftrightarrow$	Relación simétrica entre los elementos del conjunto V
Grafo dirigido simple (sin lazos)	$\Longleftrightarrow$	Matriz de ceros y unos con ceros en la diagonal	$\Longleftrightarrow$	Relación entre los elementos del conjunto V

Si permitimos lazos, entonces el lector ha de eliminar la condición “con ceros en la diagonal” en los dos tipos de matrices. Recuerde también que la relación será reflexiva si el grafo contiene todos los lazos posibles. En el apartado B de esta sección utilizaremos el lenguaje de los grafos dirigidos para obtener, por ejemplo, un procedimiento que permite decidir si una relación es transitiva o no.

Si ahora nos fijamos en un vértice v de un grafo dirigido, encontraremos unas cuantas aristas que parten de él, y otras que llegan a él. Vamos distinguir, entonces, entre el grado “entrante” al vértice v (el número de aristas que en las que v es el segundo elemento del par),

$$\text{grado}_{\text{in}}(v) = \#\{a \in A \,:\, a = (w, v) \text{ para cierto } w \in V\}$$

y el grado “saliente” (el número de aristas que en las que v es el primer elemento del par):

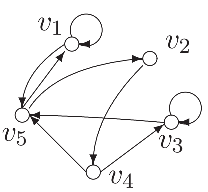
$$\text{grado}_{\text{out}}(v) = \#\{e \in A \,:\, a = (v, w) \text{ para cierto } w \in V\}\,.$$

(disculpe, por cierto, el lector el uso de los tipográficamente económicos subíndices *in* y *out* en las definiciones anteriores[47]).

Ahora, la relación que liga el número de aristas del grafo con los grados de los vértices es

$$2|A| = \sum_{v \in V} \text{grado}_{\text{in}}(v) + \sum_{v \in V} \text{grado}_{\text{out}}(v)\,.$$

En el grafo dirigido que dibujamos a la derecha, el vértice v_1 tiene grado entrante 2 (el número de unos que hay en su columna): le llega una arista desde v_5, más su propio lazo; y grado saliente también 2 (el número de unos de su fila). Obsérvese que un lazo suma un 1 tanto al grado entrante como al saliente. El grafo tiene nueve aristas, incluyendo dos lazos.



	v_1	v_2	v_3	v_4	v_5
v_1	1	0	0	0	1
v_2	0	0	0	1	0
v_3	0	0	1	0	1
v_4	0	0	1	0	1
v_5	1	1	0	0	0

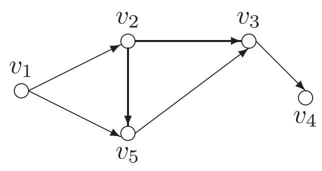
[47] Pero, ¿es que *in* y *out* no son términos castellanos? ¿Cómo que no?, véase cualquier lavadora.

A. Conexión en grafos dirigidos

Un **paseo** de v_0 a v_m en un grafo dirigido es una sucesión finita de vértices de V:

$$v_0, v_1, v_2, \ldots, v_m \qquad \text{(pueden repetirse)}$$

de forma que $(v_{i-1}, v_i) \in A$, para cada $i = 1, \ldots, m$. Nótese que exigimos que cada par *ordenado* (v_{i-1}, v_i) sea una arista del grafo dirigido. No tendrá dificultad el lector en formalizar las definiciones análogas de paseos "especiales" (ciclos dirigidos, paseos dirigidos que no repiten vértices, que no repiten aristas, etc.) en este contexto.



El paseo de arriba "conecta" el vértice v_0 con el vértice v_m, pero ahora, a diferencia de lo que ocurría con los grafos habituales, no tiene por qué "conectar" v_m con v_0. No son reversibles, estos paseos, qué le vamos a hacer. En el digrafo de la derecha podemos apreciar este contratiempo con claridad. Hay varios paseos que conectan v_1 con v_3, pero no hay manera de conectarlos en el otro sentido.

Esto nos obliga a definir más cuidadosamente la noción de conexión en un grafo dirigido. De hecho, a distinguir dos tipos de conexión. La primera de ellas se va a ceñir estrictamente a la definición de paseo que dimos anteriormente:

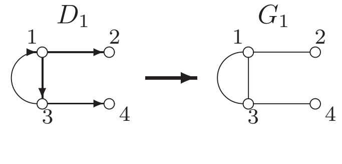
• Un grafo dirigido será **fuertemente conexo** si, para cada par de vértices v, w, existe un paseo de v a w.

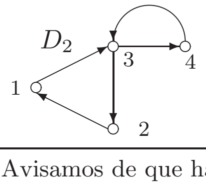
La segunda forma de "conexión" no es tan exigente como la anterior y sólo atiende, para un grafo dirigido, al grafo asociado que se obtiene "quitando las direcciones" (que, en general, será un multigrafo, pues al borrar los sentidos pueden quedar aristas dobles):

▪ Un grafo dirigido será **débilmente conexo** si el (multi)grafo subyacente es conexo.

Dependiendo de la cuestión que nos ocupe, será conveniente o necesario utilizar una u otra. Si, por ejemplo, queremos determinar si en una cierta red de calles se puede ir de cada ciudad al resto, entonces estaremos preguntándonos por la conexión fuerte. Pero, por ejemplo, cuando hablemos de grafos dirigidos "eulerianos" (véase la subsección 11.1.1), nos bastará con considerar la versión débil.

Los adjetivos "débil" y "fuerte", que en tantas ocasiones adornan nociones en Matemáticas, no se usan aquí gratuitamente. Si un grafo dirigido es fuertemente conexo, entonces es débilmente conexo (quite el lector los sentidos[48] de las aristas que forman los paseos que le han garantizado la conexión fuerte, y lo tendrá).





Pero no necesariamente al revés. La condición de ser fuertemente conexo es, no podía ser de otro modo, más "fuerte" (exigente) que la otra. El grafo dirigido D_1 que dibujamos a la izquierda (arriba) es débilmente conexo, porque el grafo G_1 asociado es conexo. Pero no es fuertemente conexo (por ejemplo, no hay paseo que conecte el vértice 4 con el vértice 1). Sin embargo, el grafo dirigido D_2 sí que es fuertemente (y, por tanto, débilmente) conexo.

[48] Avisamos de que hay cosas mucho más placenteras que este trivial ejercicio para "quitar el sentío".

B. Matriz de vecindades de un grafo dirigido y transitividad de relaciones

Como un grafo dirigido es esencialmente lo mismo que una relación (binaria) arbitraria, no debería sorprender que de la matriz de vecindades del grafo dirigido podamos extraer, de manera sencilla, la información relevante sobre las propiedades de la relación.

Para empezar, la relación será reflexiva si y sólo si en el grafo están presentes todos los lazos posibles; esto es, si todas las entradas de la diagonal de la matriz de vecindades son 1. Por otro lado, la simetría o antisimetría de la relación se corresponde exactamente con la simetría[49] o la antisimetría de la matriz. Pero lo más interesante de esta nueva forma de representar relaciones es que nos va a permitir obtener, también, un criterio "sencillo" (aunque no tanto como la simple inspección de la matriz que nos permitía decidir las otras propiedades) para decidir si una relación es transitiva o no.

Todo parte de la observación siguiente: llamemos M a la matriz de vecindades correspondiente a la ordenación $(v_1, v_2, \ldots, v_n)$ de los vértices del grafo dirigido, y sean los números m_{ij} sus entradas. Llamemos también $m_{ij}^{(k)}$ a las entradas de la matriz M^k. Entonces, $m_{ij}^{(k)}$ coincide con el número de paseos de longitud k entre los vértices v_i y v_j.

Esto ya lo vimos para los grafos habituales (teorema 10.3). El lector podrá comprobar sin esfuerzo, siguiendo las líneas de la demostración que allí expusimos, que este resultado sigue siendo válido en el caso de grafos dirigidos (en realidad es también válido si consideramos lazos, o si tenemos un multigrafo). La aplicación que aquí nos interesa es la siguiente: consideramos un grafo dirigido D con vértices $V = \{v_1, \ldots, v_n\}$; su matriz de vecindades es M, cuyas entradas son los números m_{ij}. Las entradas de la matriz M^2 las nombraremos como $m_{ij}^{(2)}$. Llamamos $\mathcal{R}$ a la relación en el conjunto V (la dada por: si $v_i, v_j \in V$, entonces $v_i \mathcal{R} v_j$ si y sólo si (v_i, v_j) es una arista de D).

Teorema 10.14 *$\mathcal{R}$ es transitiva si y solamente si se cumple que*

$$m_{ij}^{(2)} \neq 0 \implies m_{ij} \neq 0.$$

Así que, para determinar si la relación es transitiva, debemos simplemente comprobar que la matriz M^2 no tiene entradas no nulas "nuevas" con respecto a M. Si $\mathcal{R}$ es transitiva, entonces a cada entrada m_{ij} de M que sea 0 le corresponde una entrada $m_{ij}^{(2)}$ en M^2 también nula.

DEMOSTRACIÓN. Comprobaremos las dos implicaciones simultáneamente. La relación $\mathcal{R}$ es transitiva si y sólo si, para cualesquiera v_i, v_j se cumple que

$$\text{si existe } v_k \text{ tal que } v_i \mathcal{R} v_k \text{ y } v_k \mathcal{R} v_j, \text{ entonces } v_i \mathcal{R} v_j.$$

Lo de la izquierda equivale a "si existe algún camino de longitud 2 entre los vértices v_i y v_j", o también a "si $m_{ij}^{(2)} \neq 0$". Por otra parte, lo de la derecha es equivalente a que $m_{ij} \neq 0$. ■

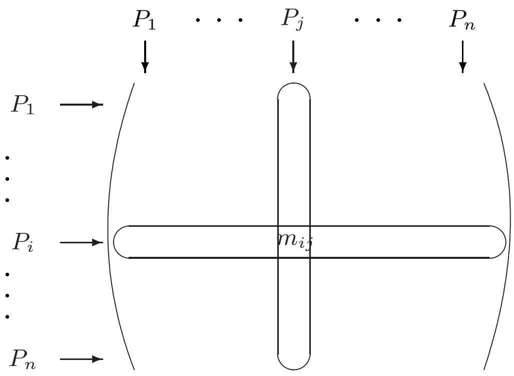
[49] Si la matriz es simétrica, entonces en posiciones simétricas de la matriz (con respecto a la diagonal) debe aparecer el mismo símbolo, bien un 0, bien un 1. Si es un 1, entonces tenemos las dos aristas posibles. Esta situación la describíamos, cuando trabajábamos con grafos, con una única arista (sin dirigir).

C. La matriz de vecindades, el Álgebra lineal y el buscador Google

Veamos un magnífico ejemplo de cómo se usa el lenguaje matemático (en este caso, teoría de grafos y Álgebra lineal) para modelar un problema real, analizarlo y extraer conclusiones. Se trata de un **problema de ordenación**, con aplicación al diseño de buscadores en la red.

Planteamos ya los ingredientes de la cuestión que nos interesa. Internet está formado por un enorme número (¿miles de millones?) de páginas, en cada uno de los cuales tenemos una serie de contenidos. Toda esta información está almacenada, de una forma muy eficiente, en las bases de datos del buscador. Cuando se introduce un término, el motor de búsqueda localiza las páginas cuyos contenidos están relacionadas, de una manera u otra, con el término buscado. El problema que nos interesa es cómo, *en qué orden*, deben ser expuestos esos resultados. Lo que uno desearía es que (casi siempre) la o las páginas que contienen información relevante aparecieran, digamos, entre las diez primeras que se muestren.

Así que queremos asignar a cada uno de las páginas de la red $P_1, \dots, P_n$ unas **importancias** $x_1, \dots, x_n$ de manera que si el resultado de una búsqueda son, digamos, cien páginas, baste consultar esa lista de importancias para poder mostrarlas ordenadas.



Disponemos de la siguiente información: para cada página P_i, conocemos qué páginas tienen un enlace hacia ella. Todos estos datos están codificados en una (gigantesca) matriz M, de dimensiones $n \times n$, formada por ceros y unos, cuya entrada m_{ij} vale 1 si es que hay un enlace desde la página P_j a P_i, y vale 0 en caso contrario. M es la traspuesta de la matriz de vecindades del enorme grafo (dirigido) de la red, donde cada vértice es una página de la red, y las aristas (orientadas) representan los enlaces entre ellas. Tal como está dispuesta, la suma de las entradas de una fila i es el número de páginas que apuntan a la página P_i, esto es, el grado entrante al vértice P_i en el grafo. La suma de la columna j da el grado saliente del vértice P_j.

Podríamos, por ejemplo, decidir que la importancia x_i de una página P_i fuera proporcional *al número de páginas* que enlazan con P_i. Esto es, a su grado entrante. Pero con este modelo podríamos asignar poca importancia páginas poco citadas, pero desde páginas muy importantes (por ejemplo, desde Amazon, o desde Microsoft).

Buscamos una asignación que adjudique alta importancia a páginas que sean citadas desde muchos lugares, pero también a páginas que lo sean desde pocos, pero a su vez importantes. Para conseguir este compromiso, podemos decidir que la importancia de una página sea proporcional *a la suma de importancias de las páginas* desde donde sea citado. Así, si la suma tiene muchos sumandos, aunque pequeños, el resultado puede ser un número alto. Pero también puede serlo, aunque haya pocos sumandos, si es que éstos son grandes. Matemáticamente, esto supone que, para cada $i = 1, \dots, n$,

$$x_i = K \sum_{j=1}^{n} m_{ij}\, x_j\,, \quad \text{(donde } K \text{ es cierta constante de proporcionalidad).}$$

Como m_{ij} es 0 si no hay enlace de P_j a P_i, sólo sumamos las importancias de las páginas que realmente enlazan con P_i. En términos matriciales, tenemos

$$\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = K \begin{pmatrix} & & \\ & M & \\ & & \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \qquad \text{esto es,} \qquad M\mathbf{x} = \lambda\,\mathbf{x}\,,$$

si llamamos $\mathbf{x}$ al vector columna de importancias y $\lambda = 1/K$.

La expresión nos es familiar: se trata de un problema de autovalores y autovectores, en el que nos interesa hallar $\mathbf{x}$, un autovector de la matriz M. Si pretendemos obtener de $\mathbf{x}$ una ordenación, convendría que sus entradas fueran números no negativos, o al menos del mismo signo (recuérdese que si $\mathbf{x}$ es un autovector, $-\mathbf{x}$ también lo es). Y, sobre todo, nos interesaría que hubiera un "único" autovector con esa propiedad. Aquí es donde entran en juego nuestros conocimientos de Álgebra lineal[50].

Veamos primero un ejemplo sencillo de la situación que hemos planteado hasta ahora, que muestra que se puede aplicar a contextos algo diferentes del de los buscadores de la red.

Ejemplo 10.3.1 *Seis equipos, $E_1, \dots, E_6$, disputan una competición, que consta de 21 partidos en total. Están divididos en dos grupos, E_1, E_2, E_3 por un lado, y E_4, E_5, E_6 por otro. Cada equipo juega 6 partidos contra los de su mismo grupo, y 3 contra los del otro. Una vez conocidos los resultados de los enfrentamientos, ¿cómo podemos ordenarlos?*

El lector familiarizado con la liga de baloncesto de la NBA (o las competiciones universitarias) sabrá que los equipos se agrupan en "conferencias", que a su vez constan de "divisiones". Cada equipo disputa un número fijo de partidos, pero no el mismo número contra cada equipo (se juegan más contra equipos de la propia conferencia). Podría ocurrir que un equipo hubiera acumulado muchas victorias simplemente por encontrarse en una división formada por equipos muy débiles. Para la ordenación que decidirá qué equipos pasan a las eliminatorias por el título, ¿deben valer todas las victorias por igual? ¿Cuál sería la ordenación más justa?

	E_1	E_2	E_3	E_4	E_5	E_6	
E_1	–	3/21	0/21	0/21	1/21	2/21	$\to 6/21$
E_2	3/21	–	2/21	2/21	2/21	1/21	$\to 10/21$
E_3	6/21	4/21	–	2/21	1/21	1/21	$\to 14/21$
E_4	3/21	1/21	1/21	–	2/21	2/21	$\to 9/21$
E_5	2/21	1/21	2/21	4/21	–	2/21	$\to 11/21$
E_6	1/21	2/21	2/21	4/21	4/21	–	$\to 13/21$

Formemos una matriz con los resultados de los partidos disputados, donde cada entrada a_{ij} es el cociente entre el número de victorias del equipo E_i sobre el E_j y el número de partidos disputados por E_i. A la izquierda mostramos una posible matriz de resultados. Sumando por filas, obtenemos la proporción de victorias de cada equipo. Mirando estos números, podríamos ordenarlos $E_3 \to E_6 \to E_5 \to E_2 \to E_4 \to E_1$. Pero observemos que, por ejemplo, E_3 ha acumulado muchas victorias contra E_1, que es el peor equipo.

Con ayuda de algún paquete matemático de cálculo, el lector podrá obtener los autovalores y autovectores de la matriz. De los seis autovalores distintos que tiene, uno de ellos, $\lambda = 9.97$, es el mayor (en módulo). Y el autovector asociado, $\mathbf{x} = (0.509, 0.746, 0.928, 0.690, 0.840, 1)$, es

[50] En las aplicaciones habituales, es habitual y conveniente modificar M (normalizando sus columnas) para obtener una matriz M' en la que la suma de las entradas de cada columna sea 1 (una **matriz de Markov**).

el *único* cuyas entradas son todas números reales y no negativos. Pues ya tenemos la respuesta que buscábamos: el orden que nos sugiere este cálculo es $E_6 \to E_3 \to E_5 \to E_2 \to E_4 \to E_1$, que difiere del anterior en los dos primeros (ahora E_6 es el mejor equipo). ♣

La idea parece prometedera: ha resuelto el problema de ordenación y, si las cosas funcionaran siempre como en el ejemplo (si existiera un único autovector con entradas positivas), podría aplicarse a la ordenación de los resultados de búsquedas en la red[51]. La respuesta a esta inquietud está contenida en el siguiente teorema, que tiene múltiples aplicaciones:

Teorema 10.15 (Perron-Frobenius) *Si A es una matriz cuyas entradas son números no negativos, entonces, asociado a un autovalor positivo, existe un autovector $\mathbf{v}$ cuyas entradas son no negativas. Si además A es* ***irreducible***[52]*, entonces ese autovector es único y corresponde a un autovalor simple, positivo y mayor que el módulo de cada uno de los otros.*

No daremos aquí la demostración, aunque el lector interesado y con cierta familiaridad con estas cuestiones puede consultar el ejercicio 10.3.1 para una versión sencilla del resultado.

No hemos acabado todavía: aunque estemos en las condiciones en las que el teorema de Perron-Frobenius garantiza la existencia del autovector (único) que resuelve el problema de ordenación, como la matriz puede ser gigantesca, podría ocurrir que fuera extremadamente complicado calcular numéricamente ese autovector. Pero no, utilizando métodos numéricos como el **método de las potencias** (véase el ejercicio 10.3.2 para los detalles), el cálculo suele resultar sencillo (en el caso de Google apenas se requieren unas horas de ordenador).

EJERCICIOS DE LA SECCIÓN 10.3

10.3.1 *Sea A una matriz 3×3 cuyas entradas son no negativas. Vamos a comprobar la existencia de un autovector $\mathbf{r}$ con entradas no negativas (lo que denotaremos, con cierto abuso de notación, como $\mathbf{r}\geq 0$). Obsérvese que si $\mathbf{x}\geq 0$, entonces $A\mathbf{x}\geq 0$ (A manda el octante positivo de $\mathbb{R}^3$ en sí mismo).*

(a) Consideremos la aplicación lineal $B(\mathbf{x}) = A\mathbf{x}/\|A\mathbf{x}\|$. Comprúebese que manda el conjunto $\{\mathbf{x}\in\mathbb{R}^3 : \mathbf{x}\geq 0, \|x\| = 1\}$ en sí mismo.

(b) Dedúzcase, utilizando el teorema del punto fijo de Brower, que existe $\mathbf{r}\geq 0$, con $\|\mathbf{r}\| = 1$, para el que $B(\mathbf{r}) = \mathbf{r}$. Compruébese, finalmente, que $\mathbf{r}$ es un autovector de A con entradas no negativas, que corresponde a un autovalor positivo.

10.3.2 *Los autovalores de una matriz A son $\lambda_1,\dots,\lambda_n$. El autovalor λ_1 es simple y mayor, en módulo, que todos los demás (en la jerga, el autovalor dominante). Sea $\mathbf{r}$ el autovector asociado. Pruébese que*

$$\lim_{k\to\infty}\frac{A^k\mathbf{r}_0}{\|A^k\mathbf{r}_0\|} = \mathbf{r} \qquad \text{para un cierto } \mathbf{r}_0.$$

[51] Una de las razones del éxito del buscador Google es, justamente, que emplea esta metodología en sus algoritmos de ordenación de resultados, lo que los diseñadores del buscador, Sergey Brin y Lawrence Page, denominan *PageRank*. Google fue creado en 1998 en la Universidad de Stanford, donde estudiaban ambos.

[52] A se dice irreducible si, por ejemplo, no existe ninguna permutación que transforma A en una matriz del tipo $\left(\begin{array}{c|c} A_{11} & A_{12} \\ \hline 0 & A_{22}\end{array}\right)$, donde A_{11} y A_{22} son matrices cuadradas.