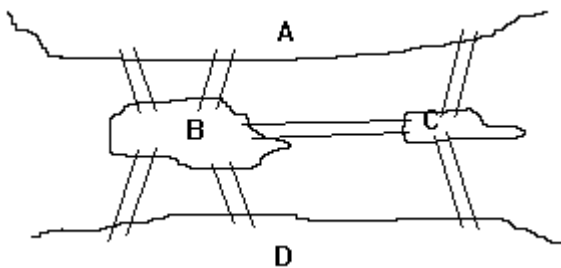


### 3. GRAFOS

#### 3.1 Introducción

El nacimiento del concepto GRAFOS se puede situar, por el año 1730, cuando **Euler** (matemático) se convirtió en el **padre** de la Teoría de Grafos al modelar un famoso problema no resuelto, llamado el "**problema de los puentes de Königsberg**".

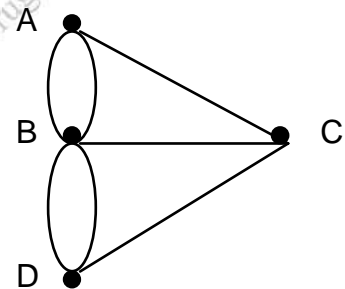


Un río con dos islas atraviesa la ciudad. Las islas están unidas, entre sí y con las orillas, a través de siete puentes. El problema consistía en establecer un recorrido que pasara una y solo una vez por cada uno de los siete puentes, partiendo de cualquier punto y regresando al mismo lugar.

Para probar que no era posible, Euler substituyó cada zona de partida por un punto y cada puente por un arco, creando así un grafo, el **primer grafo**, diseñado para resolver un problema.

Mostrar que el problema no tiene solución equivale a mostrar que el grafo no puede ser recorrido según criterios determinados.

Problema genérico: dado un grafo (con múltiples líneas entre pares de puntos) encontrar un camino que recorra el grafo pasando por cada arista exactamente una vez.



Solución: El grafo debe ser conexo, y en cada punto deben incidir un número par de líneas. Esta condición es suficiente para definir lo que se llama un **ciclo euleriano**.

A partir de Euler el modelado mediante grafos fue desarrollando esta metodología hasta convertirse en la actualidad, en una herramienta de trabajo para ciencias tan diferentes como la Física, la Química, la Sociología, la Economía, la Lingüística, etc. La teoría de grafos está íntimamente relacionada con varias ramas de la Matemáticas como por ejemplo la Teoría de Conjuntos, el Análisis Numérico, Probabilidad, Topología, etc. y es la base conceptual en el tratamiento de problemas combinatorios.

La eficacia de los grafos se basa en su **gran poderío de abstracción y la muy clara representación** de cualquier relación (de orden, precedencia, etc) lo que facilita enormemente tanto la fase de modelado como de resolución del problema. Gracias a la Teoría de Grafos se han desarrollado una gran variedad de algoritmos y métodos de resolución eficaces que nos permiten tomar una mejor decisión.

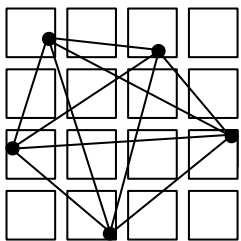
No se debe confundir el grafo con el sistema real al que está asociado. El grafo es una **estructura** que admitimos adecuada en lo concerniente a las propiedades que nos

interesan, donde luego **aplicamos** las deducciones y reglas matemáticas para obtener datos y poder decidir.

Una aplicación frecuente de la teoría de grafos es la del **método de camino hamiltoniano óptimo** para decidir el camino a seguir por un cobrador, de tal modo de economizar sus energías, las suelas de sus zapatos y su bolsillo.

El **objetivo** es hallar un camino que pase por todas las casas una y solo una vez y que nos de el costo menor en distancia. Dicho de otro modo, se deben buscar las permutaciones de las casas de forma tal que la distancia recorrida total sea mínima.

Se conoce la distancia entre cada par de casas, según si las calles son flechadas o no se orientarán o no las conexiones entre pares de casas.



Obsérvese que si se hicieran todas las permutaciones, suponiendo un caso muy reducido de diez casas, se tendrían más de 3 millones de permutaciones (10!).

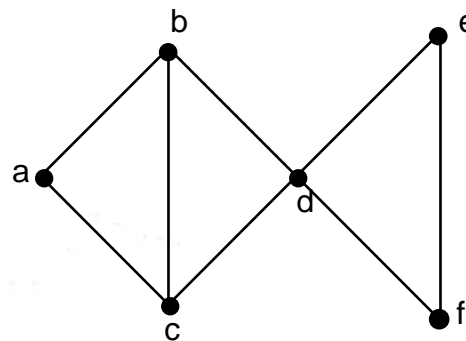
Si cada casa es representada por un vértice y cada camino entre par de casas por una arista ponderada por la distancia mínima entre pares de casas, tendremos un grafo completo y simétrico (cuando no hay calles flechadas).

El problema se reduce entonces, a obtener un camino hamiltoniano óptimo. Todo algoritmo conocido para encontrar ciclos hamiltonianos requiere al menos un tiempo exponencial de cálculo, o factorial en el peor de los casos.

Otro **ejemplo** para el que grafos provee un natural modelo matemático :

Supongamos que el siguiente grafo representa una red de líneas de teléfonos (o de comunicaciones). Estamos interesados en la vulnerabilidad respecto a interrupciones accidentales.

**Problema 1:** identificar esas líneas y centros de conexiones que deben permanecer en servicio para evitar la desconexión de la red. No existe ninguna línea que eliminada desconecte el grafo (red), pero hay un vértice, el vértice **d**, cuya desaparición (ruptura) desconecta el grafo.



**Problema 2:** encontrar un conjunto minimal de aristas necesarias para conectar los 6 vértices. Hay varios conjuntos mínimos posibles. Uno de ellos es el conjunto minimal:  $\{(a,b),(b,c),(c,d),(d,e),(d,f)\}$ .

Podemos enunciar el siguiente **resultado general:** dado un grafo  $G$  de  $n$  vértices, el conjunto mínimo de conexión de  $G$  (si existe) siempre tiene  $n-1$  aristas.

### 3.2 Definiciones Basicas

**Grafo:** Un grafo  $G$  es una dupla  $G = (X, U)$ , donde  $X$  es un conjunto finito y no vacío de elementos llamados **vértices** y  $U$  es el conjunto cuyos elementos se componen de subconjuntos de  $X$  de cardinalidad dos (2), llamados **aristas**.

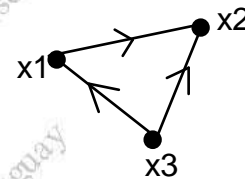
Los vértices de  $X$  se llaman usualmente  $x_1, x_2, x_3, \dots, x_n$  y se representan como puntos, las aristas  $u_1, u_2, u_3, \dots, u_m$  se dibujan como líneas.

**Grafo orientado:** Un grafo  $G^*$  es **orientado**, cuando sus aristas tienen asignadas direcciones, o sea cuando existe una relación de precedencia entre los elementos. Sus puntos se llaman **nodos**, y sus líneas **arcos**. En estos casos  $U$  es una familia de pares ordenados resultantes del producto cartesiano de  $X$ .

$$U \subseteq X \times X = \{ u_i = (x_k, x_j) : 1 \leq i \leq |U|, 1 \leq j, k \leq |X|, x_k, x_j \in X \}$$

Ejemplo:

$$G^* = (\{x_1, x_2, x_3\}, \{(x_1, x_2), (x_3, x_1), (x_3, x_2)\}) .$$



En realidad, no existen dos especies de grafos, orientados y no orientados, sino que **todos los grafos son orientados**, pero por razones conceptuales, es poco cómodo considerar las líneas orientadas para los problemas de naturaleza no orientada.

Cada vez que apliquemos un concepto orientado en un grafo  $G = (X, U)$  ese concepto deber ser considerado como aplicable de hecho, en un grafo orientado  $G^*$  al que le corresponde la orientación en los dos sentidos de cada arista.

**Orden** es el número de vértices del grafo, el cardinal del conjunto  $X$  de vértices:  $|X|$

#### Arcos incidentes a un nodo

Si un vértice  $x$  es extremidad inicial de un arco  $u = (x, y)$  y  $x \neq y$ , diremos que el arco es incidente a  $x$  hacia el exterior.  $I^+(x) = \{ y / (x, y) \in U \}$ .  $I(x) = \{ y / (y, x) \in U \}$

El número de los arcos incidentes hacia el exterior se llama **semigrado exterior de  $x$**  y se nota  $d^+(x) = |I^+(x)|$

De igual forma se define arco incidente a  $x$  hacia el interior y **semigrado interior de  $x$** . Este último se nota como  $d^-(x) = |I^-(x)|$ .

**Grado de  $x$** , es la suma del semigrado exterior e interior de  $x$ . O sea, es el número de arcos con una extremidad en  $x$ .

$$d(x) = d^+(x) + d^-(x)$$

Si todos los vértices tienen el mismo grado, el grafo al que pertenecen se llama **grafo regular**.

### 3.2.1 Recorrido de grafos.

**Cadena** (concepto no orientado):

*Es una secuencia de aristas de  $G$ , tal que cada arista de la secuencia tiene un extremo común con el arco precedente y otra con el siguiente.*

**Largo de una cadena**, es el número de aristas de la secuencia.

**Cadena elemental**, es aquella que no repite vértices.

**Cadena simple**, es aquella que no repite aristas.

**Camino** (concepto orientado)

*Es una cadena  $\mu = \{u_1, u_2, \dots, u_q\}$  en la que para todo  $u_i$  (con  $i < q$ ) el extremo terminal de  $u_i$  coincide con el extremo inicial de  $u_{i+1}$ .*

Las definiciones de largo de un camino, camino elemental y camino simple son análogas a las de cadenas, con la salvedad de la orientación.

**Sendero**, es un camino elemental (que no repite nodos).

**Vía**, es un camino cuyos arcos se pueden recorrer en su sentido directo o contrario.

**Ejemplo:** roblema del camino entre dos puntos. El siguiente es un ejemplo de como modelar una porción del universo, su problemática y como resolverla.

Supongamos que un hombre debe pasar a la otra orilla de un río llevando consigo una oveja, un repollo y un lobo. El inconveniente que se le plantea es que sólo puede cruzar con uno de ellos a la vez y sospecha que si deja solos a la oveja con la repollo ó con el lobo, la oveja se comerá al repollo ó el lobo se comerá a la oveja. Teniendo en cuenta estas restricciones, el sujeto dibuja sobre la arena de la orilla un grafo y aplicando alguna heurística o algún algoritmo conocido, encuentra el camino que debe seguir para llegar a la otra orilla con su carga intacta.

Utilice el siguiente procedimiento:

0) Dibuja el grafo: Existen 4 elementos que determinan las situaciones en cada orilla, ellos son:

H – Hombre   C – Repollo   L – Lobo   O – Oveja

1) Enumera las situaciones en una de las orillas comenzando por H,C,L,O.

2) Luego las ordena considerando:

a) se encuentra el hombre en esa orilla o no: H vs noH.

b) pasaje o secuencia de una situación a otra. (obs. que no se puede pasar de una situación en la que esté el hombre a otra en la que también esté)

3) Por último busca en el grafo un camino del estado inicial al estado final.

### 3.2.2 Clasificación De Grafos

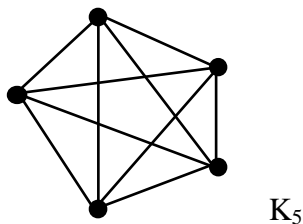
**Multigrafo**, es un grafo no orientado con múltiples aristas entre pares de nodos.

**Grafo simple**, es un grafo sin bucles, sin múltiples aristas entre pares de vértices.

#### Grafo completo

Para todo par de vértices de  $G$ , existe por lo menos una arista que los une. Por lo tanto, un grafo completo de  $n$  vértices es aquel que tiene sus  $n$  vértices mutuamente adyacentes.

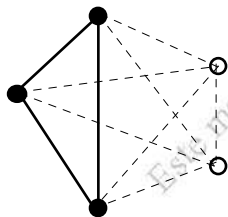
**$n$ -clique**, es un grafo completo simple de  $n$  vértices. Se nota  $K_n$ .



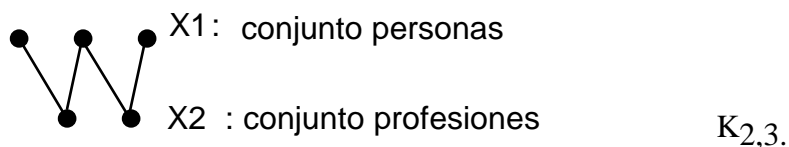
**Subgrafo de  $G = (X,U)$  engendrado por el conjunto  $A \subset X$** , es un grafo cuyos vértices pertenecen al conjunto  $A$  y cuyas aristas son aquellas de  $G$  que tienen las dos extremidades en  $A$ .

**Grafo parcial de  $G = (X,U)$  engendrado por  $V \subset U$** , es el grafo  $G' = (X,V)$  cuyos vértices son los mismos de  $G$  y cuyas aristas son las que conforman el conjunto  $V \subset U$ .

**Subgrafo parcial de  $G$** , es un subgrafo de un grafo parcial de  $G$ .



**Grafo bipartito**, es un grafo cuyo conjunto de vértices puede ser particionado en dos clases  $X_1$  y  $X_2$  de tal forma que dos vértices de la misma clase no sean jamás adyacentes. Se nota  $G = (X_1, X_2, U)$



**Grafo bipartito completo**, es aquel en el que para todo elemento de  $X_1$  y todo elemento de  $X_2$  existe por lo menos un arco que los liga.

Un grafo simple bipartito completo con  $p$  elementos en  $X_1$  y  $q$  elementos en  $X_2$  se nota  $K_{p,q}$ .

**Grafo Regular**, es aquel en el que todos sus vértices tienen el mismo grado.

**Grafo Ponderado**  $G = (X, U, W)$  donde  $(X, U)$  es un grafo y  $W$  es una función  $W: U \rightarrow Z^+$  ( $Z^+$ : enteros positivos).

Si  $u \in U$ ,  $w(u)$  es llamado el peso de la arista  $u$ . Estos pesos corresponden, según la aplicación, a costos, capacidades u otras propiedades de las aristas o arcos.

Cuando se desea asignar valores negativos o reales a los pesos de las aristas, se debe tener especial cuidado en la elección de los algoritmos ya que la correctitud de los mismos puede depender de la restricción a  $Z^+$ .

**Grafo Conexo**, es aquel en el que para cada par de vértices de  $G$ , existe una cadena que los une.

**En grafos orientados se definen 2 conceptos**

a) **Débilmente conexo**: si existe una cadena (sin tener en cuenta la orientación) que une cada par de nodos distintos.

b) **Fuertemente conexo**: si para cada par ordenado de nodos  $x$  e  $y$ , existe un camino que va de  $x$  a  $y$ .

Una componente conexa de un grafo  $G$ , es un subgrafo de  $G$  engendrado por los vértices que pueden unirse a un vértice  $x_i$  dado, mediante una cadena (puede ser todo el grafo  $G$ ).

### 3.2.3 Ciclos y Circuitos

**Ciclo**, es una cadena simple, cuyos dos vértices extremos, inicial y terminal, coinciden (no tiene en cuenta la orientación).

Si queremos describir la orientación en un ciclo designamos como:

$$u^+ = \{u_i : u_i \text{ orientada en el sentido del ciclo}\}$$

$$u^- = \{u_i : u_i \text{ orientada en el sentido contrario al ciclo}\}$$

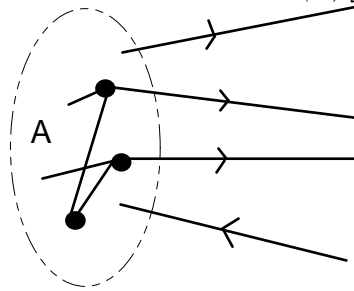
**Ciclo elemental**, es un ciclo donde no se repite ningún vértice (salvo el primero que coincide con el último). Lo notamos  $u^E = (u_1, \dots, u_n)$ .

**Propiedad 1:** Todo ciclo  $u^C$  es una suma de ciclos elementales sin aristas comunes.

**Propiedad 2:** Un ciclo es elemental si y solo si es un ciclo minimal (es decir que no se pueden deducir otros ciclos por supresión de aristas).

**Seudociclo**, es una cadena donde los extremos coinciden pero que una misma arista puede figurar más de una vez (también consecutivamente).

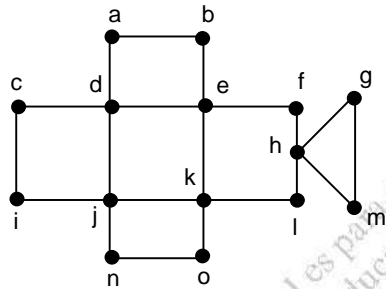
**Cociclo del conjunto de vértices A**, es el conjunto de aristas incidentes a A, del tipo  $I^+(A)$  y  $I^-(A)$ .



**Ciclo Euleriano** es aquel que incluye todas las aristas del grafo una sola vez, conteniendo cada vértice por lo menos una vez.

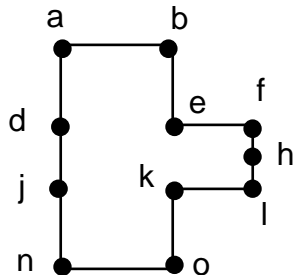
**Cadena Euleriana**, es aquella que recorre todas las aristas una sola vez (= simple) tocando todos los vértices del grafo.

Todo **multigrafo** que posee un **ciclo Euleriano** es **conexo** y todos sus vértices tienen **grado par**.



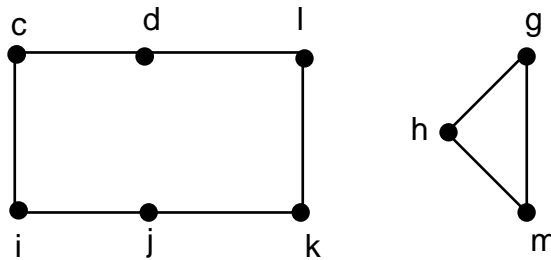
A partir del siguiente ejemplo daremos una idea del mecanismo utilizado por Euler para demostrar que la conexidad y el grado par de todos los vértices de un multigrafo, son condiciones necesarias y suficientes para garantizar la existencia de un ciclo Euleriano. Tenemos este grafo que es conexo y sus vértices tienen grado par.

1) Primero se comienza por trazar un camino simple desde un vértice, p. ej a. Supongamos que recorremos a-d-j-n-o-k-l-h-f-e-b-a. Volvimos a a.



La propiedad del grado par, significa que siempre podemos abandonar cada vértice al que entramos, excepto a. Es decir que cualquier cadena que trazamos desde a debe volver a a, formando un ciclo.

2) Las restantes aristas del grafo inicial, conforman un grafo no conexo, pero todos sus vértices mantienen el grado par, ya que al retirar el ciclo encontrado, se redujo cada grado en una cantidad par.



Cada subgrafo conexo posee un ciclo Euleriano:  $d-c-i-j-k-e-d$  y  $h-g-m-h$ .

3) Estos dos ciclos pueden ser insertados en el ciclo encontrado en 2) en los vértices comunes  $d$  y  $h$  respectivamente, originando un ciclo Euleriano  $a-d-c-i-j-k-e-d-j-n-o-k-l-h-g-m-h-f-e-b-a$ , en el grafo original.

**Teorema E.1:** *Un multigrafo (no orientado)  $G = (X,U)$  posee un ciclo Euleriano sii  $G$  es conexo y todos sus vértices son de grado par.*

Una Cadena Euleriana es una cadena que recorre todas las aristas del grafo una sola vez incluyendo todos los vértices.

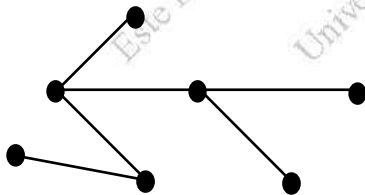
**Corolario E.2:** *Un multigrafo posee una cadena Euleriana, sii es conexo y tiene exactamente dos vértices de grado impar.*

Se puede demostrar observando lo que sucede al agregarle una arista cuyas extremidades sean los dos vértices de grado impar. El concepto de ciclo Euleriano es utilizado en la planificación de redes de alta tensión entre varias ciudades.

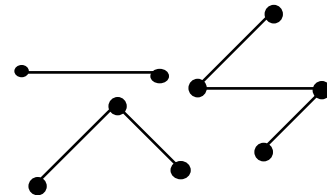
### 3.2.4 Árboles y Algoritmos de Búsqueda

Refiriéndonos al concepto de grafo no orientado en el estudio de grafos finitos, se introduce el concepto de un tipo especial de grafo, usado en una vasta variedad de aplicaciones.

**Árbol**, es un grafo finito, conexo, sin ciclos y con por lo menos 2 vértices.

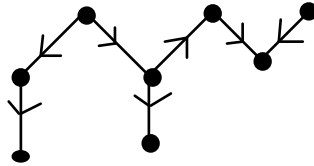


**Bosque**, es un grafo donde cada componente conexa es un árbol, es decir es un conjunto de árboles no conexos entre sí. Además es un grafo sin ciclos por estar compuesto por árboles.



**Arborescencia**, es un árbol dirigido con un nodo llamado **raíz**, tal que existe un único camino desde la raíz a cualquier otro nodo del árbol. Ese camino es elemental y simple.

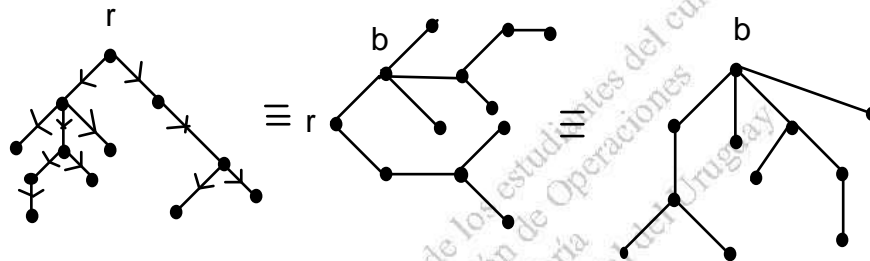




**Proposición:** Una arborecencia posee una sola raíz.

Es importante notar que la definición de arborecencia es válida para un árbol no orientado, ya que, cualquier vértice del árbol puede ser raíz y además un árbol no posee ciclos. Al determinar una raíz, se determina la existencia de una sola cadena que conecta la raíz con cada uno de todos los vértices restantes.

Entonces, un árbol lo puedo convertir en arborecencia tomando cualquier vértice como raíz y asignándole direcciones a las aristas desde el nodo raíz. Por lo tanto, designaremos como árbol, indistintamente a un árbol o arborecencia.



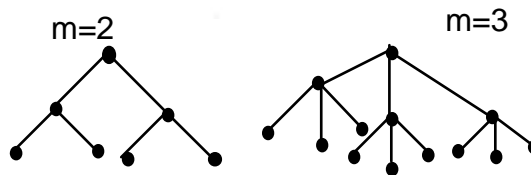
La manera standard de dibujar una arborecencia es colocando la raíz a en la cima de la figura. Así podemos definirle niveles a los vértices del grafo ( la raíz tiene nivel 0 ).

El **número asignado al nivel** de un vértice  $x$  del árbol ( $T = (X,U)$ ) corresponde al largo de la cadena (único, elemental) que une a  $r$  con  $x$ .

Se llama **padre** de un vértice  $x$ , al vértice  $y$ , próximo anterior a  $x$  en la cadena elemental y simple que va de la raíz a  $x$ , siendo  $x$  **hijo** de  $y$  dos vértices con el mismo padre se llaman **hermanos**. La relación padre-hijo se extiende a **ascendiente** y **descendientes** de un vértice  $x$ . Además cada vértice  $x$  de  $T$  es la raíz de un subárbol de  $x$  y sus descendientes. Los vértices sin hijos se llaman **hojas**, y todos los otros vértices que tengan hijos se llaman **vértices internos** de  $T$ .

$T$  se llama **árbol m-ario**, si cada vértice interno del árbol (arborecencia)  $T$  tiene  $m$  hijos. En particular si  $m = 2$ ,  $T$  es un **árbol binario**.

Ejemplos:



**Teorema T.1:** Un árbol con  $n$  vértices tiene  $n-1$  aristas.

**Altura de un árbol (o arborecencia)**, es el número de aristas del camino más largo, es decir, el número de nivel más alto de cualquier vértice.

**Arbol balanceado**, es aquel árbol de altura  $h$ , en el cual todas sus hojas se encuentran en el nivel  $h$  ó  $h-1$ . Ver el ejemplo anterior en el que la cantidad de comparaciones a realizar se minimiza.

Los árboles se utilizan también en muchas aplicaciones para verificar la conectividad, o la cantidad de ocurrencias de ciclos o subgrafos completos, en la búsqueda de esas propiedades o estructuras, muchos algoritmos utilizan lo que llamamos un esqueleto o árbol de cubrimiento.

**Esqueleto o árbol de cubrimiento (spanning tree) de un grafo G:** es un subgrafo que es un árbol y que contiene todos los vértices de  $G$ .

El esqueleto de un grafo puede ser construido ya sea por Búsqueda Primero en Profundidad: BPP (**depth -first search: DFS**) o por Búsqueda Primero a lo Ancho: BPA (**breadth-first-search: BFS**).

**Algoritmo de Búsqueda Primero en Profundidad**

Sea  $G = (X,U)$  ,  $x, v$  pertenecen al conjunto  $X$

**DFS(x)**

Visite y marque  $x$

**Mientras** exista un vértice  $v$  no marcado adyacente a  $x$

DFS( $v$ )

**fin mientras**

**fin**

Usando **BPP (o DFS)** , se toma algún vértice como raíz y se comienza a construir una cadena desde la raíz. La cadena continúa hasta que no se puede continuar mas abajo sin repetir un vértice que ya está en la cadena. El vértice donde esa cadena debe terminar es una hoja. Entonces se vuelve atrás (backtrack), se retrocede un nivel al padre de esa hoja, y se trata de construir una cadena desde ese padre en otra dirección u otro vértice no incluido en la cadena anterior, y así sucesivamente...

**Algoritmo de Búsqueda Primero a lo Ancho**

Sea  $G = (X,U)$  ,  $x, v, s$  pertenecen al conjunto  $X$ ,  $Q$  es una cola o lista FIFO.

**BFS(x)**

Visite y marque  $x$ .

Inserte  $x$  en  $Q$

**Mientras**  $Q$  no esté vacía **realice**

Saco el primer elemento  $s$  de  $Q$

**Para** cada vértice  $v$  no marcado adyacente a  $s$

visite y marque  $v$

inserte  $v$  en  $Q$

**fin para**

**fin mientras**

**fin**

Usando **BPA** ( o **BFS**), se toma cualquier vértice  $x$  como raíz, insertando en la cadena todas las aristas incidentes a ese  $x$ . Luego, sucesivamente, se van agregando las aristas incidentes a los nodos adyacentes a  $x$ , siempre que no estén incluidos en la cadena que estamos construyendo, hasta que todos los nodos hayan sido visitados .

Es **importante** notar que si el grafo no es conexo, entonces no existe ningun esqueleto que lo recorra.

#### Algoritmo para verificar que un grafo es conexo

- 1) Use BPP ( o BPA) para tratar de construir un esqueleto del grafo.
- 2) Si todos los vértices del árbol son alcanzados en la búsqueda, entonces se ha encontrado un esqueleto del grafo y por lo tanto el grafo es conexo.
- 3) Si la búsqueda no recorrió todos los vértices, entonces el grafo no es conexo.

### 3.3 Conectividad. Clausura Transitiva.

Un grafo se puede representar para su manipulación de las siguientes maneras :

- a) Gráficamente. Mediante puntos que representan los nodos y líneas que representan las aristas del grafo.
- b) Notación de conjuntos. Ya sea por extensión (enumerando vértices y aristas) o por comprensión.
- c) Estructuras de datos, como por ejemplo listas encadenadas, stacks, etc.
- d) Matrices

#### 3.3.1 Representación Matricial

Una relación binaria en un conjunto  $X$  finito:  $X = \{x_1, x_2, \dots, x_n\}$ ,  $|X| = n$ , es el subconjunto  $R = \{(x_i, x_j); x_i R x_j\}$ , compuesto por el conjunto de pares ordenados resultantes del producto cartesiano de  $X \times X$ . Si  $(x_i, x_j) \in R$ , decimos que  $x_j$  esta relacionado con  $x_i$  y notamos  $x_i R x_j$ .

La relación  $R$  puede ser representada como una matriz  $n \times n$  cuyos elementos son:

$$R = [r_{ij}] = \begin{cases} 1 & \text{si } x_i R x_j \\ 0 & \text{si no} \end{cases}$$

Podemos afirmar que todo grafo  $G$  es orientado, por lo que existe una relación binaria entre pares de elementos de  $G = (X, U)$ , que es la relación de **adyacencia**  $A$ .

Entonces, dado un grafo  $G=(X,U)$ , de orden  $n$ , se puede representar la relación de adyacencia , mediante una matriz  $A=[a_{ij}]_{n \times n}$ , llamada **matriz de adyacencia** de  $G$ , definida por

$$A = [a_{ij}] = \begin{cases} 1 & \text{si } x_i A x_j, (x_i, x_j) \in U \\ 0 & \text{si no} \end{cases}$$

Es importante observar que en un grafo no orientado la relación de adyacencia es simétrica y por lo tanto también lo será su matriz asociada, ya que en el momento del almacenamiento solamente es necesario tratar la mitad de la matriz.

Observaciones

- 1) La matriz de adyacencia es una matriz booleana, sus elementos son 0 o 1.
- 2)  $\sum_i x_{ij}$  = semigrado exterior de  $x_j$
- 3)  $\sum_j x_{ij}$  = semigrado interior de  $x_i$

En aquellas aplicaciones en las que intervienen costos o propiedades propias de la relación entre los elementos del sistema, estamos ante un Grafo Ponderado:  $G=(X,U,W)$

En estos casos la matriz de adyacencia se define como:

$$A = [a_{ij}] = \begin{cases} w(u) & \text{si } x_i A x_j, (x_i, x_j) \in U \\ 0 & \text{si no} \end{cases}$$

Esta matriz es booleana solamente en aquellos casos en que  $w(u) = 1$ , para toda arista  $u$ .

El concepto básico en grafos es el de relación binaria, ya que una relación binaria puede ser una relación de equivalencia, la que particiona al conjunto  $X$  de vértices en clases de equivalencias.

Una relación binaria es una **relación de equivalencia** cuando es:

- 1) Reflexiva.  $x \equiv x$
- 2) Simétrica.  $x \equiv y$  entonces  $y \equiv x$
- 3) Transitiva.  $x \equiv y, y \equiv z$  entonces  $x \equiv z$

Estas tres propiedades asociadas a un grafo  $G = (X,U)$ ,  $X=\{x_i ; 1 \leq i \leq n\}$ , significan que:

**$x_i \equiv x_j$  si  $x_i = x_j$  o si existe algún camino de  $x_i$  a  $x_j$  y de  $x_j$  a  $x_i$ .**

**Definición:**

*Un grafo o subgrafo es **fuertemente conexo**, si  $\forall (x_i, x_j), x_i \neq x_j, \exists$  camino de  $x_i$  a  $x_j$ .*

Toda relación de equivalencia particiona al conjunto  $X$  en clases de equivalencias  $X_i = \{x_{i1}, \dots, x_{ip}\}$  que llamaremos **componentes fuertemente conexas** del grafo  $G$ .

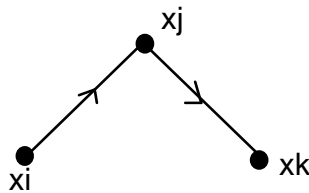
Por lo tanto, cada componente fuertemente conexa de un grafo  $G$  es un subgrafo fuertemente conexo,  $G_i=(X_i,U_i)$ , del grafo  $G$ .

Notar que la conexidad está muy relacionada a la existencia de caminos, existencia que se puede verificar mediante la propiedad transitiva.

**Definición** Una relación  $R$  es transitiva, en un conjunto  $X$ ,

si  $\forall i, j, k = 1, 2, \dots | X |$  se cumple que  $x_i R x_j$  y  $x_j R x_k$  entonces  $x_i R x_k$

Ahora bien la relación de adyacencia  $A$ , no siempre es transitiva, observese el siguiente ejemplo donde  $x_i A x_j, x_j A x_k$  no implica que  $x_i A x_k$ .



Introducimos entonces el concepto de **clausura transitiva** o **alcance de la relación  $A$**  en el conjunto  $X$ .

### 3.3.2 Clausura Transitiva, Matriz de Alcance

**Definición:** La clausura transitiva de una relación  $A$  en el conjunto  $X$  es la relación  $T$  definida por:

$$x_i T x_j \Leftrightarrow \begin{cases} x_i A x_j \\ \exists m \geq 2 \text{ y una secuencia de elementos } x_{k1}, x_{k2}, \dots, x_{km} \in X \\ \text{tales que } x_i = x_{k1}, x_{kt} A x_{k(t+1)} \text{ para } t = 1, 2, \dots, m-1, x_{km} = x_j \end{cases}$$

Observación: La clausura transitiva de una relación  $A$ , es la relación  $A$  en sí misma.

Si  $A$  es la relación de adyacencia en  $X$ , entonces la Clausura Transitiva en  $X$  es la Relación de Alcance definida por:

$$x_i T x_j \Leftrightarrow \exists C_{i,j} x_j, \quad C_{i,j} = \text{camino que va de } x_i \text{ a } x_j$$

$x_j$  es alcanzable desde  $x_i$ , o que  $x_j$  pertenece a la Clausura transitiva de  $x_i$ , lo notamos :  $x_j \in \Gamma^*(x_i)$

La relación de alcance es una relación transitiva y cuando además cumple con las propiedades de simetría y reflexividad, es una relación de equivalencia que particiona a  $X$  en clases de equivalencia

$$X_1, X_2, \dots, X_m \Leftrightarrow \forall x_i, x_j \in X_k \quad k=1,2,\dots,m \text{ se verifica: } x_i = x_j \text{ o } x_i T x_j$$

Cada clase corresponde a una componente conexa del grafo. Es decir que se pueden determinar la componentes fuertemente conexas del grafo calculando la clausura transitiva en el conjunto X de nodos.

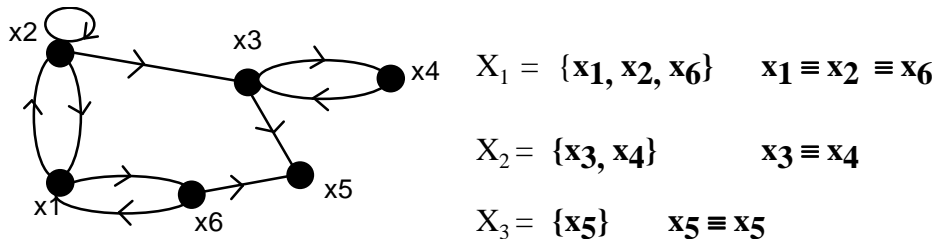


fig.CFC

La relación de alcance se representa matricialmente.

**Definición :** La matriz de alcance es una matriz  $T = [t_{ij}]_{n \times n}$

$$T = [t_{ij}] = \begin{cases} 1 & \text{si } x_i T x_j, (x_i, x_j) \in U \\ 0 & \text{si no} \end{cases}$$

la que indica la existencia de caminos entre pares de nodos y se obtiene calculando la potencia booleana de la matriz de adyacencia A.

**Definición:** Matriz de potencias booleana de A es la matriz  $A^{(p)}$  con  $p > 0$

$$A^{(p)} = [a_{ij}^{(p)}] = \begin{cases} 1 & \text{si } \exists C_{i,j}, l(C_{i,j}) = p \\ 0 & \text{si no} \end{cases}$$

$$A^{(0)} = I, \quad A^{(1)} = A \quad \text{y} \quad A^{(p)} = A^{(p-1)} \otimes A, \quad (\otimes \text{ producto booleano}).$$

Nota:  $\mathbf{a_{ij}}^{(2)} = \bigvee_{k=1,n} (\mathbf{a_{ik}} \wedge \mathbf{a_{kj}})$

**Teorema (Matriz de alcance):** Sea  $A_{n \times n}$  la matriz booleana de la relación de adyacencia en un grafo G de orden n entonces  $T = A(I \vee A)^{(s-1)} \quad \forall s \geq n-1$ , es la matriz de alcance de G.

La experiencia indica que alcanza con calcular  $T = (I \vee A)^{(n-1)}$

Si al calcular T,  $t_{ij} = 1 \quad \forall x_i$  en X, entonces el grafo G es fuertemente conexo, o sea que  $\forall x_i, x_j$  existe un camino de  $x_i$  a  $x_j$ . En caso contrario, si  $t_{ij} = 0$  para algún par  $x_i, x_j$ ,  $x_j$  no es alcanzable desde  $x_i$  pudiendose diferenciar las distintas componentes fuertemente conexas de G, clasificando grupos de nodos equivalente como aquellos que son alcanzables entre sí.

Si los nodos del grafo están ordenados por componentes, cada una de ellas se va a caracterizar por un bloque o submatriz cuadrada de 1s en la diagonal de la matriz T. En estos casos se dice que la matriz T es **triangular por bloques**, es decir que esta compuesta por bloques cuadrados dispuestos en la diagonal.

**Justificación del Teorema**

Sea  $A^{(p)}$  la Matriz de Potencias Booleana de A,  $p \geq 0$ ,

$$a_{ij}^{(p)} = \begin{cases} 1 & \text{ssi } \exists C_{i,j}, l(C_{i,j}) = p \\ 0 & \text{si no} \end{cases}$$

Entonces la matriz de alcance  $T = \bigvee_{p=1, \infty} A^{(p)}$  cuyos elementos se definen

$$t_{ij} = \begin{cases} 1 & \text{ssi } \exists C_{i,j}, l(C_{i,j}) > 0 \\ 0 & \text{si no} \end{cases}$$

La suma booleana de todas las potencias posibles de la matriz de Adyacencia A, implica un cálculo poco cómodo, entonces usamos el siguiente lema.

**Lema 1:** *Sea un grafo G de orden n, si  $\exists Cij$  simple y elemental entonces  $l(Cij)$  es a lo sumo  $n-1$ . (los otros camino no interesan).*

**Demostación:**

$$T = \bigvee_{p=1, s} A^{(p)} \quad \forall s \geq n-1 \quad (\text{para no perder generalidad})$$

$$T = A^{(1)} \bigvee A^{(2)} \bigvee A^{(3)} \dots \bigvee A^{(s)} = (\text{por prop.distr}) = A ( I \bigvee A \bigvee \dots A^{(s-1)} )$$

$$T = A ( I \bigvee A )^{(s-1)} \quad \forall s = n-1 \quad \text{LQQD}$$

**Lema 2**

*Se cumple :  $I \bigvee A \bigvee \dots A^{(s-1)} = (I \bigvee A)^{(s-1)}$ ,  $\forall s \geq 1$  (se demuestra por Induccion Completa en s).*

La experiencia indica que para calcular la matriz de alcance, alcanza con calcular

$$T = (I \bigvee A)^{(n-1)}$$

Intuitivamente se justifica dado que I incluye  $x_i = x_j$ , para cualquier  $i, j$ , (recordar definición de relación de alcance), además  $A (I \bigvee A)^{(n-2)} \approx (I \bigvee A)^{(n-1)}$

### Triangularización por bloques

Para una mejor manipulación de las componentes fuertemente conexas y para una mejor distinción de las mismas, se triangulariza la matriz de alcance  $T$ . Se define una nueva base mediante un reordenamiento de los nodos, permutando las filas y columnas de  $T$ . Siendo  $T$  la matriz del sistema de ecuaciones  $Tx = b$ , debo transformar este sistema a otro equivalente con otra base que llamaré  $Qx$ . Entonces construyo una matriz  $Q$ , tal que  $QTQ^tQx = Qb$ . La matriz de este nuevo sistema es  $QTQ^t$  que implica un nuevo orden del conjunto  $X$  de nodos, a partir de las componenetes fuertemente conexas del grafo.

**Ejemplo:** La siguiente matriz de adyacencia corresponde al grafo de la fig. CFC:

I V A

	x1	x2	x3	x4	x5	x6
x1	1	1	0	0	0	1
x2	1	1	1	0	0	0
x3	0	0	1	1	1	0
x4	0	0	1	1	0	0
x5	0	0	0	0	1	0
x6	1	0	0	0	1	1

	1	1	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	1
	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0
	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0
	0	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	0	0
	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
	1	0	0	0	1	0	1	1	0	0	1	1	1	1	0	0	1	0
1	1	0	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	
1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0	
0	0	1	1	0	0	0	0	1	1	1	0	0	0	1	1	1	0	
0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	
1	0	0	0	1	0	1	1	0	0	1	1	1	1	1	1	1	1	

$(I V A)^2$ 
 $(I V A)^4$ 
 $(I V A)^5 = T$

$X_1 = \{x_1, x_2, x_6\}, \quad X_2 = \{x_3, x_4\}, \quad X_3 = \{x_5\}$

1	0	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	1
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	1	0

= Q

Mantengo  $x_1, x_2$ :  $x_1 \implies x_1, x_2 \implies x_2$   
 Reordeno  $x_6, x_3, x_4, x_5$ :  
 $x_3 \implies x_4$   
 $x_4 \implies x_5$   
 $x_5 \implies x_6$   
 $x_6 \implies x_3$

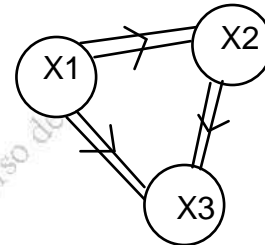


1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	0	0	1

$$= Q T Q^t$$

Las componentes fuertemente conexas de un grafo  $G$ , se pueden condensar en un punto, formando un nuevo grafo  $G_C$  sin ciclos, cuyos arcos son aquellos del grafo  $G$  que unen distintas componentes, al que se llama **grafo condensado**.

El siguiente es el  $G_C$  de la fig CFC. Los grafos condensados se utilizan en la planificación del tránsito y de transporte urbano.



**Procedimiento** para condensar un grafo:

- 1) Condensar  $G$ .
- 2) Encontrar algún camino hamiltoniano, si existe, en el grafo condensado  $G_C$ .
- 3) Encontrar un camino hamiltoniano en cada nodo del grafo condensado (en cada componente fuertemente conexa de  $G$ ).
- 4) Si es posible, unir el o los caminos encontrados en el paso 3) con el encontrado en el paso 2).

Supongamos que el grafo condensado del ejemplo anterior describe el posible recorrido de una línea de ómnibus, y se desea determinar aquel recorrido que pasa una sola vez por cada una de las paradas (nodos del grafo).

**Definición**

*Un camino hamiltoniano en un grafo  $G$ , es aquel camino que pasa una y solo una vez por cada vértice del grafo.*

Si  $G$  es de orden  $n$ , el largo del camino hamiltoniano es el del camino elemental de longitud máxima.

### 3.4 Multiplicación Latina - Camino Hamiltoniano

**Procedimiento:**

Se enumeran todos los caminos elementales posibles de longitud 1,2,... n-1.  
 Detenerse al encontrar todos los caminos de largo n-1.

Si no existe Camino Hamiltoniano, se conocerán todos los caminos elementales intermedios. La enumeración de los caminos elementales se realiza mediante el método de *Multiplicación Latina*.

Los elementos de la matriz Latina de largo 1,  $[M]^{(1)} = ((m_{ij}^{(1)}))$  se definen

$$m_{ij}^{(1)} = \begin{cases} x_i x_j & \text{si } (x_i, x_j) \in U, i \neq j \\ 0 & \text{si } (x_i, x_j) \notin U, i \neq j \text{ o } i = j \end{cases}$$

Retirando la primera letra de los elementos distintos a 0 obtenemos  $[M']^{(1)} = ((m'_{ij}{}^{(1)}))$  latina.

$$m'_{ij}{}^{(1)} = \begin{cases} x_j & \text{si } (x_i, x_j) \in U, i \neq j \\ 0 & \text{si } (x_i, x_j) \notin U, i \neq j \text{ o } i = j \end{cases}$$

Los caminos elementales de largo 2 se calculan de la siguiente manera

$$[M]^{(1)} \text{ L } [M']^{(1)} = [M]^{(2)}$$

siendo

$$m_{ij}^{(2)} = \bigcup_{k=1, n} (m_{ik}^{(1)} \otimes m'_{kj}{}^{(1)}), \text{ donde}$$

$$m_{ik}^{(1)} \otimes m'_{kj}{}^{(1)} = \begin{cases} m_{ik}^{(1)} \text{ conc. } m'_{kj}{}^{(1)} & \text{si } m'_{kj}{}^{(1)} \notin m_{ik}^{(1)} \text{ y } m_{ik}^{(1)} \wedge m'_{kj}{}^{(1)} = 1 \\ 0 & \text{si no.} \end{cases}$$

$$M^{(3)} = M^{(2)} \text{ L } M^{(1)}, \quad M^{(4)} = M^{(3)} \text{ L } M^{(1)}$$

Para encontrar los caminos hamiltonianos nos detenemos en el paso n-1 (4, en nuestro caso). Los elementos de la matriz  $M^{(n-1)}$  contiene la sucesión de nodos de cada uno de los caminos hamiltonianos, si existe alguno.

**Ejemplo:**

