4.- GRAFOS

4. Definiciones básicas:

Un grafo es la representación por medio de conjuntos de relaciones arbitrarias entre objetos. Existen dos tipos de grafos según la relación entre los objetos sea unívoca o biunívoca. Los primeros forman los grafos dirigidos o dígrafos y los segundos los grafos no dirigidos o simplemente grafos. En la mayor parte de los algoritmos que serán nuestro objeto de estudio se hace referencia a la termología básica que se propone a continuación. Dicha terminología; por desgracia, no es estándar y puede llegar a variar en los distintos textos que existen en la materia. Cuando exista ambigüedad se harán las aclaraciones según sea necesario.

Un grafo dirigido o dígrafo consiste de un conjunto de vértices V y un conjunto de arcos A. Los vértices se denominan nodos o puntos; los arcos también se conocen como aristas o líneas dirigidas que representan que entre un par de vértices existe una relación unívoca aRb pero no bRa. De modo que los arcos se representan comúnmente por medio de pares ordenados (a,b), donde se dice que a es la cabeza y b la cola del arco y a menudo se representa también por medio de una flecha, tal como se muestra en la figura 1.

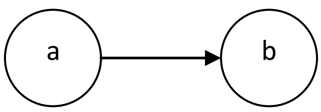


Figura 1 Grafo dirigido

$G = \{V, A\}$ donde $V = \{v_1, v_2, \cdots, v_n\}$, $A = \{a_1, a_2, \cdots, a_n\}$ y $a_i = (v_j, v_k)$ tal que $v_j, v_k \in V$. En dicho grafo se entiende que $(v_i, v_j) \neq (v_j, v_i)$ y en muchos casos solo existe uno de los pares de vértices.

Un vértice que solo tiene arcos saliendo de él se denomina fuente y un vértice que solo tiene arcos dirigidos hacia él se denomina sumidero. Dicha nomenclatura es importante cuando los dígrafos se usan para resolver problemas de flujos.

Un grafo no dirigido, o grafo, al igual que un dígrafo consiste de un conjunto de vértices V y un conjunto de arcos A. La diferencia consiste en que la existencia de aRb presupone que bRa también existe y además que son iguales. De este modo es indistinto hablar del arco (a,b) o (b,a), tampoco tiene sentido hablar de la cabeza o la cola del arco. Los grafos representan como lo indica la figura 2, donde los círculos representan los vértices y las líneas representan los arcos.

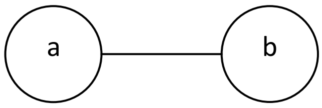


Figura 2 Grafo no dirigido

$G = \{V, A\}$ donde $V = \{v_1, v_2, \cdots, v_n\}$, $A = \{a_1, a_2, \cdots, a_n\}$ y $a_i = (v_j, v_k)$ tal que $v_j, v_k \in V$. En dicho grafo se entiende que $(v_i, v_j) \Leftrightarrow (v_j, v_i)$ y además $(v_i, v_j) = (v_j, v_i)$, donde ambos pares de vértices representan el mismo arco.

Existen además grafos en donde los arcos tienen asociado algún valor en cuyo caso hablamos de grafos ponderados y ahora se representan los arcos como tripletas. Sigue existiendo la información de los vértices unidos por dicho arco además de la información del peso de dicho arco. Así pues el arco se representa como $a = (v_i, v_j, w)$ donde v_i, v_j son el origen y destino y w es el peso respectivamente.

Un nodo **b** se dice que es **adyacente** al nodo **a** si existe el arco (a, b), tómese en cuenta que para un grafo no dirigido necesariamente **a** es también adyacente a **b**. Esto no ocurre en los grafos dirigidos donde la existencia de (a, b) no implica que (b, a) también existe. Este concepto es de particular importancia dado que los grafos suelen representarse en la computadora por medio de listas o matrices de adyacencias.

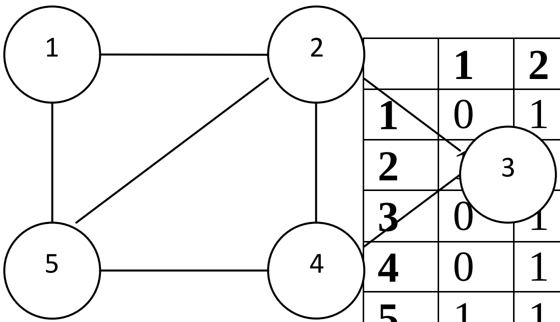
Un arco (a,b) **incide** en el nodo b, de igual modo en grafo no dirigido dicho arco también incide en el nodo a debido a que también existe (b, a). El número de arcos que inciden en un nodo le otorga el **grado** a dicho nodo. El nodo con mayor grado en el grafo le indica el grado de dicho grafo. También se acostumbra representar a un grafo por medio de listas o matrices de incidencias.

Existen otras definiciones que son útiles para explicar el funcionamiento de un algoritmo en particular, se definirán los conceptos en su momento.

4. Métodos de representación en computadora

Tal como se adelanto en el apartado anterior, existen varias formas de representar un grafo en la computadora y cada una tiene sus ventajas y desventajas. Mostraremos las más comunes y la forma de implementarlas.

La primera forma es por medio de una matriz de adyacencias, con este método se tiene una matriz de tamaño nxn, donde n es el numero de vértices o nodos en el grafo. Una forma simple de ver la información guardada en dicha matriz es que los renglones de las mismas representan el origen y las columnas el destino de cada arista o arco en el grafo. Si el grafo es no ponderado se acostumbra poner un cero en el (renglón i, columna j) de la matriz cuando no existe dicho arco y un uno cuando dicho arco existe en el grafo. En el caso de grafos ponderados, se acostumbra poner una bandera (normalmente el valor de infinito) en las posiciones donde no existe un arco y el peso correspondiente en las posiciones donde si existe.



	1	**2**	**3**	**4**	**5**
1	0	1	0	0	1
2	1	[illegible]	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

Figura 3 Grafo no ponderado y su matriz de adyacencia

Debe notarse que para un grafo no dirigido la matriz de adyacencia es simétrica y que la diagonal principal contiene ceros. Esto puede llegar a aprovecharse para ahorrar tiempo en algunos algoritmos. La representación por medio de matriz se prefiere para algoritmos donde el numero de arcos es grande en proporción al numero de vértices. Si sucediera lo contrario se prefiere la representación por medio de listas de adyacencia.

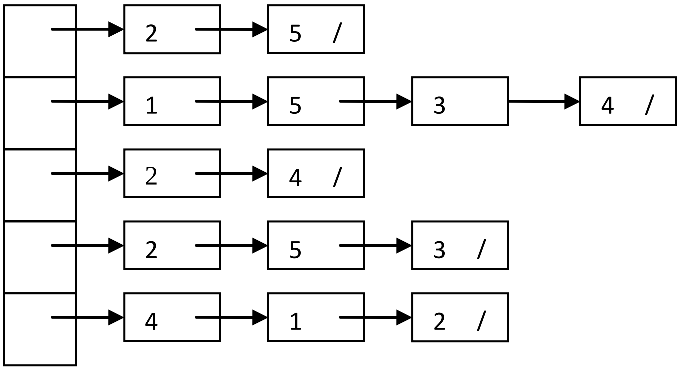


Figura 4 Lista de adyacencia para el grafo de la figura 3

Las estructuras de datos para las dos formas de representación anteriores pueden modelarse en C como sigue:

```
char grafo[MAX_VERT][MAX_VERT], visitado[MAX_VERT];

void inserta(char i, char j){
      grafo[i][j] = grafo[j][i] = 1;
}

void limpia_grafo(){
int i, j;
      for(i = 0; i < nvert; i++){
            visitado[i] = 0;
            for( j = i; j < nvert; j++)
                  grafo[i][j] = grafo[j][i] = 0;
      }
}
```

Listado 1 Representación por matriz de adyacencia

Para encontrar los adyacentes al vértice i se tendría que construir un ciclo que evaluara en el renglón i aquellas columnas que tienen un uno. Como en el siguiente fragmento de código, donde se quieren meter los adyacentes no visitados a una pila.

```
for(i = 0; i < nvert; i++){
      if(!visitado[i] && grafo[j][i]){
            pila.push(i);
            visitado[i] = 1;
      }
}
```

Listado 2 Encontrar adyacentes al vértice j

En las implementaciones de algoritmos se darán más detalles acerca del manejo de las estructuras de datos. Por ahora revisemos la versión por medio de listas de adyacencia.

```
#include <vector>
#include <list>

vector< list<int> > grafo(MAX_VERT);
char visitado[MAX_VERT];

void inserta_arista(int i, int j){
      grafo[i].push_back(j);
      grafo[j].push_back(i);
}
```

```
void limpia_grafo(){
int i;
      for(i = 0; i < nvert; i++){
            grafo[i].clear();
            visitado[i] = 0;
      }
}



list<int>::iterator aux, fin;

aux = grafo[j].begin();
fin = grafo[j].end();

while(aux != fin){
      if(!visitado[*aux]){
            pila.push(*aux);
            visitado[*aux] = 1;
      }
      aux++;
}
```

Listado 3 Versión por listas de adyacencias

En ambos casos se ha supuesto un grafo no dirigido y no ponderado. En el caso de un grafo dirigido basta con eliminar la doble inserción y no considerar la existencia de (j, i) para cada (i, j). La implementación para grafos ponderados por medio de matrices se presenta a continuación:

```
#define INFINITO MAXINT
```

```
char grafo[MAX_VERT][MAX_VERT], visitado[MAX_VERT];

void inserta_arista_ponderada(int i, int j, int w){
      grafo[i][j] = w;
}

void limpia_grafo(){
int i, j;
      for(i = 0; i < nvert; i++){
            visitado[i] = 0;
            grafo[i][i] = 0;
            for( j = i+1; j < nvert; j++)

                  grafo[i][j]   =   grafo[j][i]   =
INFINITO;
      }
}



int suma_pesos(int x, int y){
      if( x == INFINITO || y == INFINITO) return
INFINITO;
      else return x + y;
}
```

Listado 4 Grafos ponderados por medio de matrices

Adicionalmente se muestra una función para sumar pesos que permite solucionar el problema de sumar aristas con valor de infinito. Lo cual es muy común en algoritmos con grafos ponderados.

Ahora podemos revisar la versión con listas de adyacencias. Podemos notar que es necesario utilizar un par que guarde el nodo destino además del peso. Aquí se define el primer miembro como el destino y el segundo como el peso.

```
#include <vector>

#include <list>


vector< list< pair<int , int> > > grafo(MAX_VERT);

char visitado[MAX_VERT];


void inserta_arista_ponderada(int i, int j){

pair ady;

      ady.first = j;

      ady.second = w

      grafo[i].push_back(ady);

}
```

Listado 5 Grafos ponderados con listas de adyacencia

En muchos casos es necesario ordenar las aristas de un grafo ponderado de acuerdo a su peso. Ante tal situación es apropiado definir una estructura que contenga la información de las aristas y luego insertarlas en una cola de prioridad. En otras ocasiones, se desea formar un subconjunto de aristas que cumplen con una cierta propiedad como cuando se obtienen los árboles de expansión de los recorridos de un grafo o se encuentran los árboles de expansión mínima.

```
typedef pair< int , int > ARISTA

priority_queue< int, ARISTA> cola;
```

4.1 Grafos dirigidos

Definición de tipos para grafos ponderados

Intuitivamente un grafo es un conjunto de vértices unidos por un conjunto de líneas o flechas dependiendo de si el grafo es dirigido o no dirigido.

Gráficamente los vértices se representan por círculos, las líneas (o aristas) pertenecen a los grafos no dirigidos y las flechas (o arcos) a los grafos dirigidos.

Formalmente, un **grafo no dirigido** (o simplemente grafo) consta de:

1. un conjunto finito de vértices V
2. un conjunto de aristas E en el que cada arista es un conjunto de exacta-mente dos vértices.

Un **grafo dirigido** (o dígrafo) consta de:

1. un conjunto finito de vértices V
2. un conjunto de arcos $E \subset V \times V$ (obsérvese que cada arco es un par ordenado vértices)

Tanto en los grafos dirigidos como en los no dirigidos las secuencias de vértices pueden formar caminos y ciclos. Definimos un **camino de longitud** ` como una secuencia de vértices $u_0, u_1, . . . , u_{`}$ tales que, para todo i tal que $1 \leq i \leq `$, $(u_{i-1}, u_i) \in E$ (si se trata de un dígrafo) o $\{u_{i-1}, u_i\} \in E$ (si se trata de un grafo). Un camino es **simple** si todos los vértices del camino, excepto quizás el primero y el ´ultimo, son diferentes. Un **ciclo** es un camino simple que comienza y acaba en el mismo vértice.

Decimos que un grafo $G = (V, E)$ es **conexo** si para todo par de vértices $u, v \in V$ existe un camino en el grafo G que comienza en u y acaba en v.

Un tipo especial de grafo conexo es el **árbol** (árbol libre) que es un grafo no dirigido, conexo y acíclico. Un árbol también puede definirse como un grafo no dirigido en el que hay exactamente un camino entre todo par de vértices. Los ´arboles tienen algunas sencillas propiedades que pueden resultar muy útiles, como por ejemplo:

1. un árbol con n vértices contiene exactamente n − 1 aristas
2. si se añade una ´única arista a un ´árbol, el grafo resultante contiene un único´ ciclo
3. si se elimina una única arista de un árbol, entonces el grafo resultante deja de ser conexo

Decimos que un digrafo es **fuertemente conexo** si para cualquier par de v´ertices existe un camino que los une y decimos que es **débilmente conexo** si el grafo resultante de convertir los arcos en aristas es conexo.

En un grafo (o digrafo), decimos que un vértice $v \in V$ es **adyacente** a un vértice $u \in V$ si y solo si $\{u, v\} \in E$ en el caso de los grafos o $(u, v) \in E$ en el caso de los digrafos. En un digrafo, si un vértice v es adyacente a un vértice u, decimos que el vértice u es **incidente** al vértice v.

En un grafo, el **grado de un vértice** es ´el número de vértices adyacentes a ´el y el **grado del grafo** el máximo de los grados de sus vértices.

P

Teorema 1.1 En un grafo no dirigido $G = (V, E)$ se tiene que $_{v \in V}$ grado(u) =

2|E|.

La demostración hecha en clase es por inducción sobre el número de vértices del grafo.

En un digrafo el **grado de entrada de un vértice** es el número de sus vértices incidentes y el **grado de salida de un vértice** es el número de sus vértices adyacentes.

Decimos que un grafo (digrafo) es **completo** si contiene el máximo número de aristas (arcos) posible. ¿Cuántas aristas (arcos) son?

Ejemplos de todo lo anterior: Dados en clase.

4.1.1.2. Representación

Existen varias estructuras de datos que pueden utilizarse para representar grafos y digrafos. La elección de la estructura de datos adecuada depende del tipo de operaciones que se quieran aplicar al conjunto de vértices y aristas (arcos) del grafo (digrafo) en cuestión. Las representaciones más comunes son las matrices de adyacencia y las listas de adyacencia.

4.1.2.1. Matrices de Adyacencia

Dado un grafo (digrafo) G = (V, E) con V = {1, 2, . . . , n}, la matriz de adyacencias de G es una matriz A de boléanos de tamaño n x n en la que A[i][j] es cierto si y sólo si la arista (arco) que une al vértice i con el vértice j está en E ({i, j} en el caso de grafos y (i, j) en el caso de digrafos).

Es fácil observar que la matriz de adyacencias de un grafo no dirigido es una matriz simétrica y que podemos ahorrar espacio (la mitad) guardando solo su parte inferior.

En un grafo representado por matrices de adyacencia el tiempo que se re-quiere para accesar un elemento es independiente de las tallas de V y de E, por tanto, ´esta puede ser una representación adecuada en las aplicaciones en las que es necesario saber con mucha frecuencia si una determinada arista (arco) está presente en el grafo (digrafo).

La desventaja principal de utilizar una matriz de adyacencias para representar un grafo (digrafo) es que la matriz requiere un espacio $\Omega(n^2)$ incluso si el grafo (digrafo) es **esparso**, es decir, si tiene bastante menos de n^2 aristas (arcos). Solo leer o examinar la matriz requerirá tiempo $O(n^2)$, en perjuicio de posibles algoritmos de tiempo $O(n)$ para manipular grafos (digrafos) con $O(n)$ aristas (arcos). Una alternativa para evitar esta desventaja es utilizar listas para representar un grafo.

4.1.3.2. Listas de Adyacencia

Dado un grafo $G = (V, E)$ la lista de adyacencias de un vértice i de G, es una lista, en un orden cualquiera, de todos los vértices adyacentes a i. Se puede representar G como un vector L en el que L[i] es un puntero a la lista de adyacencias del vértice i.

La cantidad de memoria que requiere esta representación es proporcional a la suma del número de vértices más el número de punteros (que corresponde al número de aristas o arcos según sea el caso). Es decir, el coste en memoria es $\Theta(n + m)$ con $n = |V|$ y $m = |E|$. Si el grafo es esparso este coste es mucho menor que el requerido por la representación matricial, en cambio, si el grafo es **denso** (lo que sucede cuando el número de aristas o arcos es $\Theta(n^2)$) la diferencia en requerimientos de memoria entre una representación u otra no es tan significativa.

La desventaja de esta representación es que el determinar si una arista (arco) está o no en el grafo puede tomar tiempo $O(n)$

ya que el número máximo de vértices que pueden haber en la lista de adyacencias de un vértice dado es n.

Ejemplos: Dados en clase.

4.1.3. TAD Grafo

Es posible definir formalmente los TADs correspondientes a los grafos y a los grafos dirigidos y estudiar las implementaciones de sus operaciones. No entraremos en los detalles porque la mayoría de ellos se han estudiado antes en la asignatura. Las operaciones más comunes en grafos y digrafos incluyen las operaciones de leer la etiqueta de un vértice o de una arista (arco), insertar o borrar vértices y aristas (o arcos), navegar por el grafo (digrafo) siguiendo sus aristas (arcos). Esta ´ultima operación requiere la definición de un tipo ´índice que nos permita recorrer todos los vértices adyacentes a uno dado. Para ello, definimos las macros:

1. forall_ver(u,G) que recorre todos los vértices de G

2. forall_adj(uv,G[u]) que hace que el iterador uv recorra toda la lista de adyacencias G[u]. En este caso es posible imaginar que u es el vértice de salida, que uv es la arista (o arco) y que *uv es el vértice de destino.

La implementaci´on correspondiente se encuentra en el fichero graph.hh incluido en el c´odigo de la asignatura (http://www.lsi.upc.edu/~ada).

Sea G un grafo y $\bar{G}$ un grafo dirigido (dígrafo).

Vocabulario:

fuertemente conexo	$\bar{G}$ es fuertemente conexo, si existe un recorrido entre cada pareja de vértices de G
orientable	G es orientable, si existe un grafo $\bar{G} \simeq G$ que es fuertemente conexo
componente fuertemente conexa	conjunto máximo de vértices de $\bar{G}$ cuyo subgrafo inducido es fuertemente conexo

Notaciones:

$d_i(v)$ grado entrante

$d_o(v)$ grado saliente

Teorema (Robbins):

G orientable $\Longrightarrow$ G es conexo y no contiene puentes

¿Algoritmo que calcule una orientación?

(¿Qué se entiende bajo una orientación óptima? Depende: minimizar el promedio de las distancias, minimizar el máximo de las distancias, minimizar el máximo de las diferencias entre las distancias en G y en $\bar{G}$ correspondientes.)

Se puede explorar también un digrafo en anchura (BFS) o en profundidad (DFS). A parte de las aristas del árbol y las aristas de retroceso, DFS produce aristas de progreso y aristas de cruce.

DFS se usa para producir una ordenación topológica de un digrafo acíclico, es decir, en el orden aparece un vértice v antes de un vértice w, si existe un camino desde v a w.

DFS se usa para determinar los componentes fuertemente conexos.

¿Algoritmo que calcule los componentes fuertemente conexos?

se puede seguir también recorridos eulerianos:

Teorema:

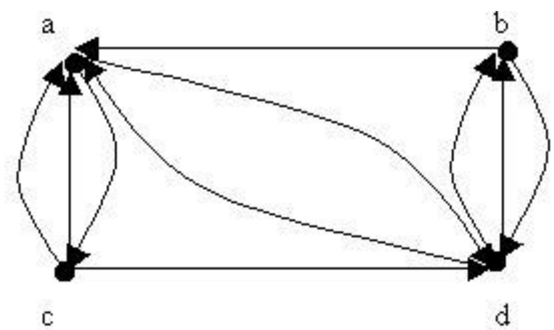
$\bar{G}$ es euleriano $\iff$ $\bar{G}$ es conexo y $\forall v \in V : d_i(v) = d_o(v)$

¿Algoritmo que calcule un camino euleriano en un digrafo?

4.2. Multigrafos y grafos pesados

Definición:

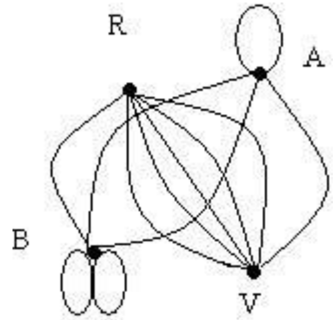
Sea G = (V, E) un grafo dirigido, donde V es un conjunto y E es un multiconjunto de pares ordenados de V $\times$V. G es llamado un multigrafo dirigido y geométricamente puede representarse como un conjunto de vértices V y un conjunto de flechas E entre los vértices, donde no existe restricción en el numero de flechas de un vértice a otro.



Multigrafo Dirigido

Ahora consideremos una representación gráfica de un mapa de carreteras en el cual una arista entre dos ciudades corresponde a un carril en una autopista entre las dos ciudades. Como a menudo hay autopistas de varios carriles entre pares de ciudades, esta representación origina un multigrafo.

La noción de multigrafo no dirigido puede definirse de manera similar a la de un multigrafo dirigido.

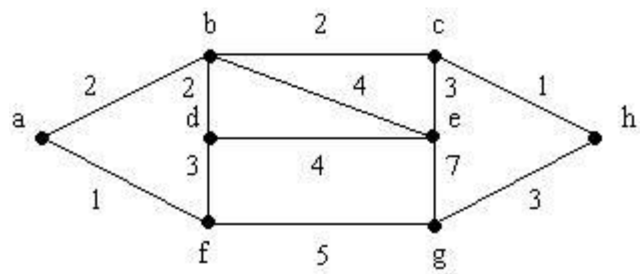


Multigrafo No Dirigido

Definición:
Un grafo ponderado (o grafo con peso) es un grafo en el cual hay datos asociados a sus lados, el valor *w*(*i*, *j*) esta asociado con el lado (*i*, *j*) y se llama ponderación o peso del lado (*i*, *j*).

Definición:
Eel peso o ponderación de un grafo es la suma de los pesos de sus lados. Frecuentemente el peso de un camino se le conoce como longitud del camino.

Ejemplo:
Si se interpretan las ciudades como vértices y los caminos entre ellas como sus lados, al asignarles un valor a sus caminos resulta un grafo ponderado o con peso.

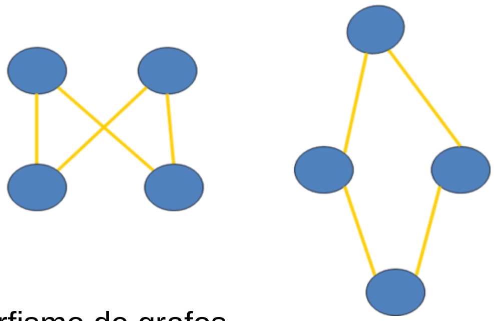


Grafo Ponderado

4.3. ISOMORFISMOS

Dos grafos son isomorfos si existe una correspondencia 1:1 entre nodos y orillas de forma que se mantengan las incidencias

Isomorfismo de subgrafos: un grafo es isomorfo a un subgrafo (subconjunto de nodos y orillas) de otro grafo



- Isomorfismo de grafos
  - correspondencia 1:1 entre dos grafos G1 - G2
- Isomorfismo de subgrafos
  - correspondencia entre un grafo G1 y los subgrafos de G2
- Doble isomorfismo de subgrafos
  - correspondencia entre los subgrafos de G1 y los subgrafos de G2

1. TÉCNICAS PARA ISOMORFISMO

Búsqueda con *backtracking*

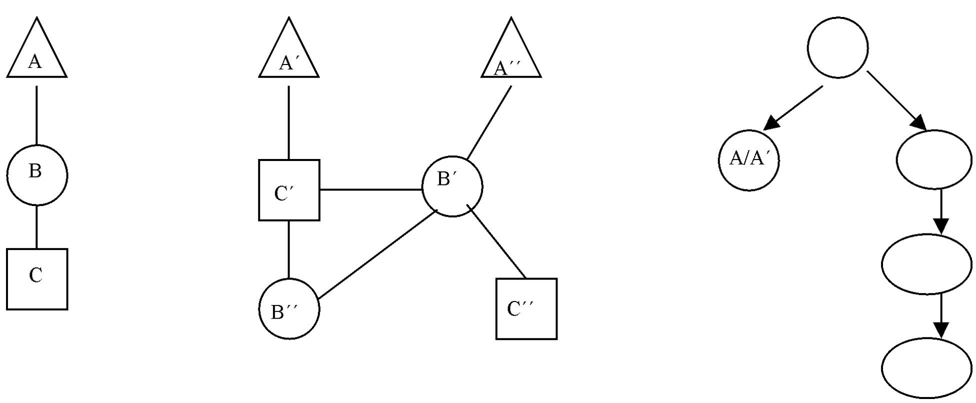
Se construye un árbol en el que las trayectorias corresponden a isomorfismos:

se toma un nodo de G1 y todas sus posibles correspondencias en G2 (primer nivel)

se buscan los nodos conectados a los nodos correspondientes del primer nivel (segundo nivel)

se continua hasta que no existan correspondencias

las trayectorias en el árbol corresponden a isomorfismos de subgrafos entre G1 y G2

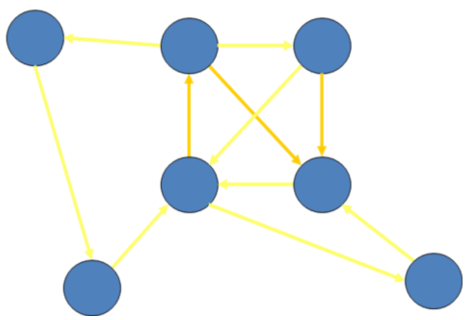


- Búsqueda de *cliques*

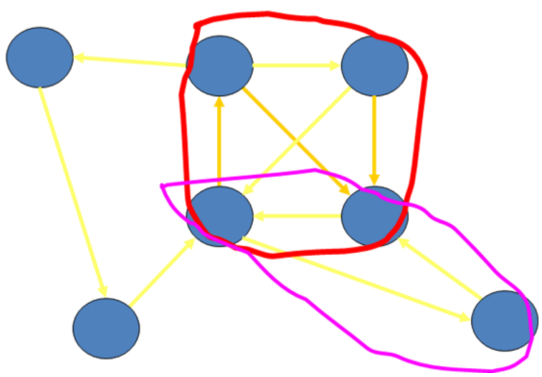
Grafo completo: cada par de nodos distintos son adyacentes

Conjunto completo: subconjunto W de G que induce un subgrafo completo de G

Clique: subconjunto de nodos que es conjunto completo y máximo (no hay un conjunto completo que lo contenga)

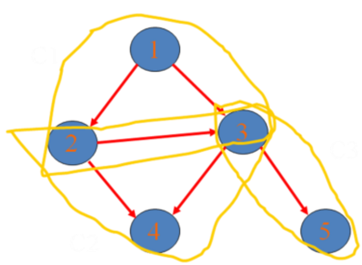


Cliques



Un ordenamiento de *cliques* [C_1, C_2, ... C_p] tiene la propiedad de intersección secuencial si todos los nodos comunes con *cliques* previos están contenidos en el mismo *clique* (padre)

Esto se cumple si los nodos tienen un ordenamiento perfecto y los *cliques* se ordenan de acuerdo al nodo con número mayor



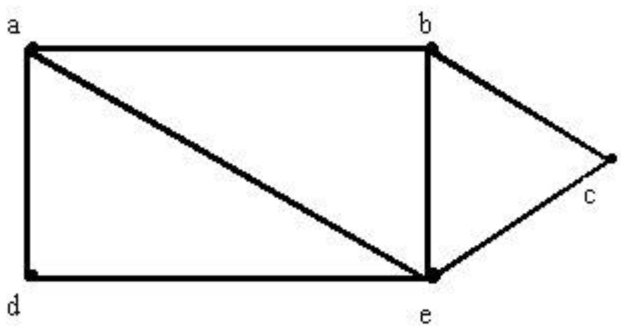
Un grafo dirigido es triangulado si cada circuito simple de longitud > 3 tiene una cuerda

Para tener un ordenamiento de *cliques* con la propiedad de intersección secuencial es necesario que el grafo sea triangulado

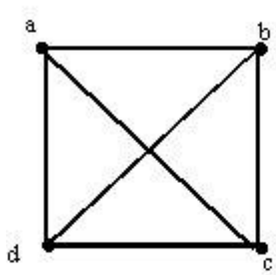
4.4. **GRAFOS APLANABLES**

Este tipo de grafos, además de aparecer con mucha frecuencia también cuentan con muchas propiedades interesante. Se analizarán algunas de las más importantes.

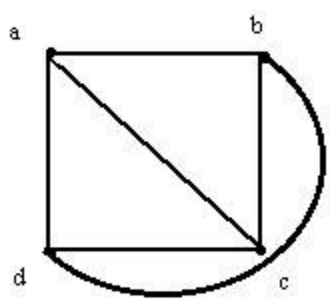
Definición:
Diremos que un grafo es aplanable si puede ser dibujado sobre un plano de tal manera tal que ninguna arista se cruce con otra excepto, desde luego, en los vértices comunes. El siguiente es un grafo aplanable:



el grafo *i*) también es aplanable ya que puede dibujarse como se muestra en el grafo *ii*)

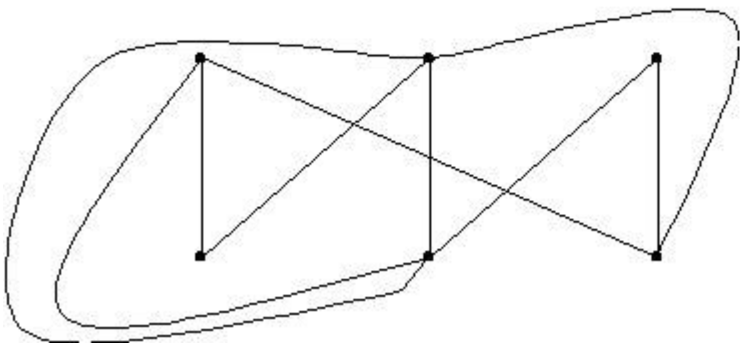


i)



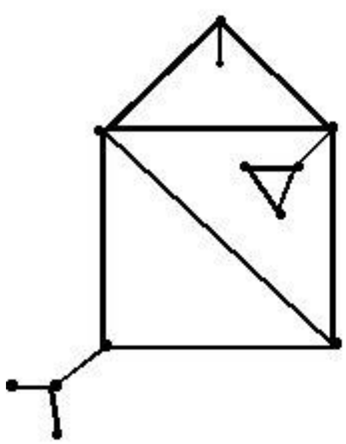
ii)

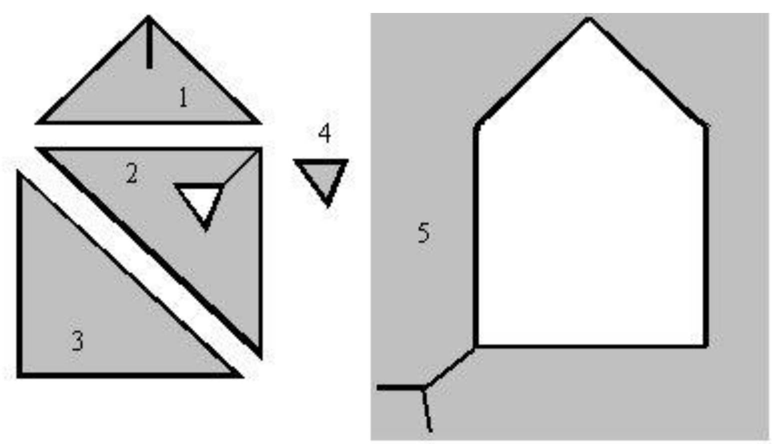
Ejemplo:
La siguiente figura es un grafo no aplanable que a decir verdad corresponde al problema de determinar si es posible conectar las casas 1, 2, 3 a los servicios de Luz, Agua y Drenaje, de tal manera que no haya 2 líneas de conexión que se crucen una con la otra.



Definición:
Una región (o cara) de un grafo aplanable se define como una área del plano que está acotada por aristas y no pude continuar dividiéndose subáreas.

Ejemplo:
El siguiente grafo tiene 5 regiones que son:





Definición:
Diremos que una región es infinita si su área es infinita y se dice que es finita, si su área es finita. En un grafo aplanable se tienen exactamente una región infinita.

Tenemos el siguiente resultado:

$v - e + r = 2$

donde *v*, *e* y *r* son el numero de vértices, aristas y regiones respectivamente. Esta ecuación se conoce como la Formula de Euler para grafos aplanables. Sin excepción alguna todos los grafos aplanables conexos deben satisfacer la formula de Euler.

En cualquier grafo aplanable lineal conexo que no tenga lazos y que tenga 2 o mas aristas se cumple la siguiente desigualdad:

$e \leq 3v - 6$

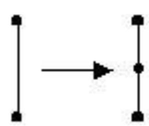
Debido a que el grafo es lineal cada región es acotada por 3 o más aristas por lo tanto el número es mayor o igual que $3r$. en la frontera a lo largo de 2 regiones, el numero total es igual o menor a $2e$ así tenemos:

$$2e \geq 3r \quad \text{ó} \quad \frac{2}{3}e \geq r$$

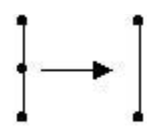
De acuerdo a la fórmula de Euler, tenemos que:

$$v - e + \frac{2e}{3} \geq 2 \quad \text{ó} \quad 3v - 6 \geq e$$

Es evidente que la planaridad de un grafo no se ve afectada si una arista es dividida en dos arista por la inserción de un vértice de grado 2 como *i*) o si 2 aristas se combinan en una sola arista al eliminar este vértice como en *ii*)



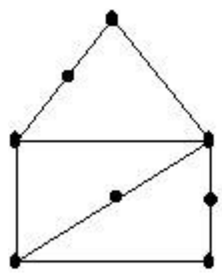
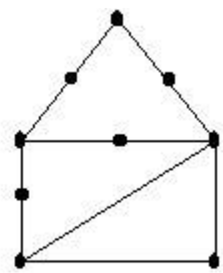
i)



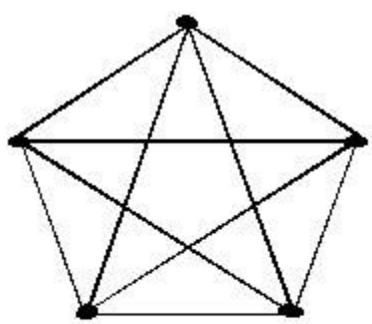
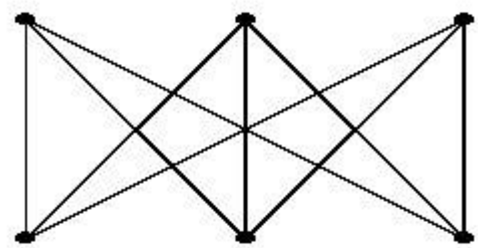
ii)

Definición:
Dos grafos $\mathbf{G_1}$ y $\mathbf{G_2}$ son isomorfos bajo vértices de grado 2, si son isomorfos ó si pueden transformarse en grafos isomorfos mediante repeticiones de inserciones y/o eliminaciones de vértices de grado 2 como en *i*) y *i i*).

Ejemplo:
Los siguientes grafos son isomorfos bajo vértices de grado 2.

Teorema de Kuratoswki
Un grafo es aplanable si y solo si no contiene cualquier subgrafo que sea isomorfo bajo vértices de grado 2 a cualquier de los siguientes grafos, que son llamados de Kuratowski

GRAFOS APLANABLES

En un mapa de carreteras , las líneas que indican las carreteras y autopistas se intersecan por lo general solamente en puntos de confluencia o en poblaciones . Pero hay ocasiones en que las carreteras parecen intersecarse cuando una se localiza sobre otra , como en el caso de un paso elevado .

En este caso , las dos carreteras están en diferentes niveles o planos.

En general, se puede probar que cualquier grafo plano G que cumpla las dos siguientes condiciones:

Propiedad del rombo. Toda arista de G es la diagonal de un rombo (diamond) una de cuyas dos mitades no contiene puntos de S.

Propiedad del buen polígono. No pueden existir caras largas con diagonales cortas.

DEFINICION 1

Un grafo o multigrafo G es plano si podemos dibujar G en el plano de modo que sus aristas se intersequen sólo en los vértices de G . Este dibujo de G se conoce como una inmersión de G en el plano.

EJEMPLO 1

En los grafos de la figura 1 son planos . El primero es un grafo 3-regular , ya que cada vértice tiene grado 3 ; es plano pues ningún par de aristas se intersecan , excepto en los vértices .

El grafo (b) parece un grafo no plano ; las aristas {x,z} y {w,y} se cruzan en un punto distinto del vértice . Sin embargo podemos trazar nuevamente este grafo como se muestra en la parte (c) de la figura . En consecuencia , K_4 es plano.

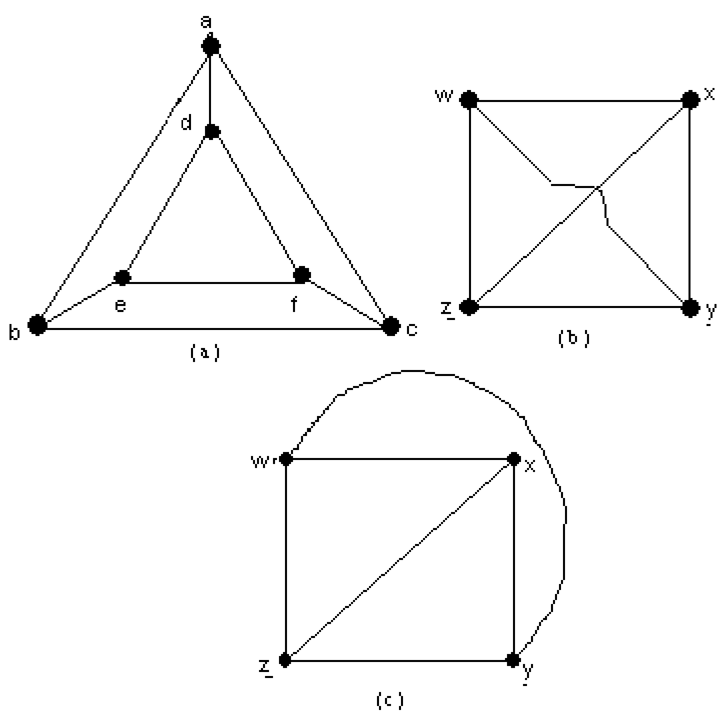


Figura 1 (a) (b) (c)

EJEMPLO 2

Al igual que k_4 , también k_1, k_2 y k_3 son planos. En la figura 2 se muestra un intento de representar k_5 en el plano . Si k_5 fuera plano, entonces cualquier inmersión tendría que contener el pentágono de la parte (a) de la figura .

Como un grafo completo tiene una arista por cada par de vértices distintos , añadimos la ariasta {a,c} como se muestra en la parte (b) .

Esta arista está contenida completamente dentro del pentágono de la parte (a) . (Podríamos haber dibujado la arista de la región exterior determinada por el pentagóno) . En la parte (c) , se añaden las aristas {a,d} , {c.e} y {b,e} .

Ahora examinaremos los vértices b y d . Se necesita la arista {b,d} para obtener k_5 . El vértice de está dentro de la región formada por el ciclo de aristas {a,c} , {c,e} y {e,a} , mientras que b está fuera de la región .

Así , al dibujar la arista {b,d} , hay que intersecar una de las aristas existentes al menos una vez , como lo muestran las aristas punteadas en la parte (d) . En consecuencia , k_5 no es plano .

DEFINICION 2

Un grafo G=(V,E) es bipartito si V= $V_1 \cup V_2, V_1 \cap V_2 = \phi$ y cada arista de G es la forma {a,b} con a Î $a \in V_1$ y b Î $b \in V_2$. Si cada vértice de V_1 está unido con los vértices de V_2, se tiene un grafo bipartito completo . En este caso , si ½ V_1½= m , ½ V_2½= n , el grafo se denota con $k_{m,n}$.

La figura 3 muestra dos grafos bipartitos . El grafo de la parte (a) satisface la definición para V_1= {a,b} y V_2= {c,d,e} . Si se añaden las aristas {b,d} y {b,c} , el resultado es el grafo bipartito completo $k_{2,3}$, que es plano . El grafo (b) de la figura es $k_{3,3}$. Sea V_1 = { h_1, h_2, h_3 } y V_2 = {u_1, u_2, u_3} ; interpretamos V_1 como un conjunto de casas y V_2 como un conjunto de servicios .

Entonces $k_{3,3}$ es el grafo de servicios . En la figura 3(b) parece que esto no es posible y que $k_{3,3}$ no es plano.

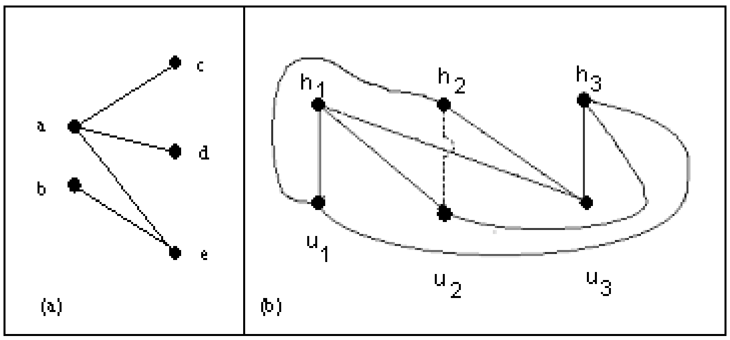


Figura 3

Sin embargo observamos que cuando estamos trabajando con grafos no planos , k_5 o $k_{3,3}$ serán el origen del problema .

DEFINICION 3

Sea G=(V,E) un grafo no dirigido sin lazos , tal que E ¹ Æ . Una subdivisión elemental de G resulta cuando eliminamos una arista e={v,w} de G y entonces las aristas {u,v}{v,w} se añaden a G , donde v Ï V.

Los grafos no dirigidos sin lazos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$ son homeomorfos si son isomorfos o si ambos pueden obtenerse al mismo grafo no dirigido sin lazos H por una sucesión de subdivisiones elementales.

EJEMPLO 3

a) Sea G=(V,E) un grafo no dirigido sin lazos con ½E½ $\geq$ 1 . Si G' se obtiene de G por una subdivisión elemental , entonces el grafo G' =(V',E') satisface ½V'½=½V½+ y ½ E'½=½E½+ .

b) Consideremos los grafos G, G_1, G_2 y G_3 de la figura 4 . En este caso , G_1 se obtiene de G por medio de una subdivisión elemental : se elimina la arista {a,b} de G y se añaden las aristas {a,w} y {w,b} .

El grafo G_2 se obtiene de G mediante dos subdivisiones elementales . Por lo tanto , G_1 y G_2 son homeomorfos .

Así mismo , G_3 puede obtenerse de Gm con cuatro subdivisiones elementales , por lo que G_3 es homeomorfo a G_1 y G_2.

Sin embargo , G_1 no puede obtenerse de G_2 (o G_2 de G_1) por una sucesión de subdivisiones elementales . Además el grafo G_3 puede obtenerse de G_1 o G_2 por una sucesión de subdivisiones elementales : seis (de tales sucesiones de tres subdivisiones elementales) para G_1 y dos para G_2. Pero ni G_1 ni G_2 pueden obtenerse de G_3 por una sucesión de subdivisiones elementales .

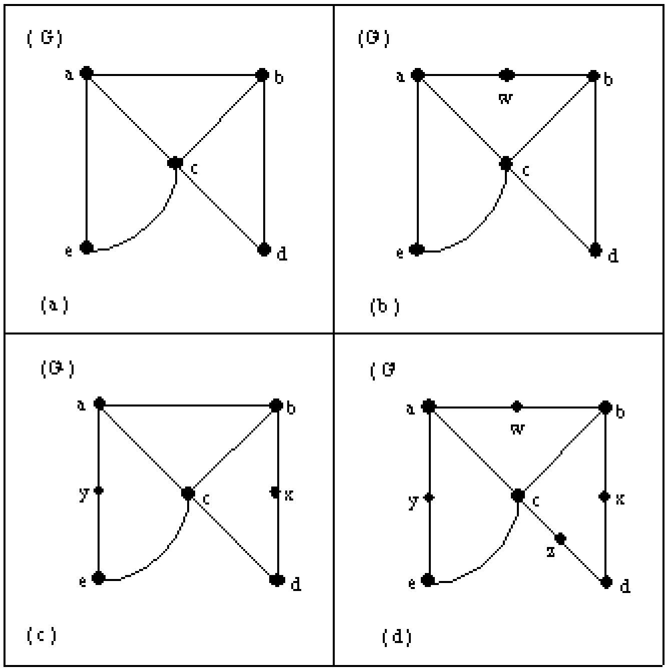


Figura 4

Podríamos pensar que los grafos homeomorfos son isomorfos excepto , posiblemente , por los vértices de grado 2 . En partircular si dos grafos son homeomorfos , son simúltaneamente planos (o no planos) .

TEOREMA 1

Sea G=(V,E) un grafo o multigrafo plano conexo con ½V½= v y ½E½= e . Sea r el número de regiones en el plano determinadas por una inmersión (o representación) plana de G ; una de estas regiones tiene un área infinita y se conoce como región infinita . Entonces v – e + r = 2.

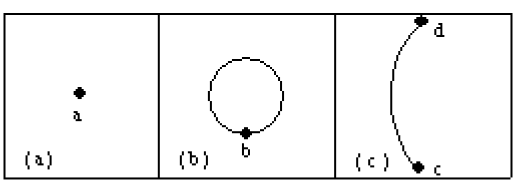


Figura 5

Demostración .- La demostración se hace por inducción sobre e. Si e=0 o 1 , entonces G es isomorfo a uno de los grafos de la figura 5 . El grafo de la parte (a) tiene v=1 , e=0 y r=1 ; entonces v – e + r = 1 – 0 + 1 = 2 .

Para el grafo de la parte (b) , v=1 , e=1 y r=2. El grafo de la parte (c) tiene v=2 , e=1 y r=1 . En ambos casos , v – e + r = 2 .

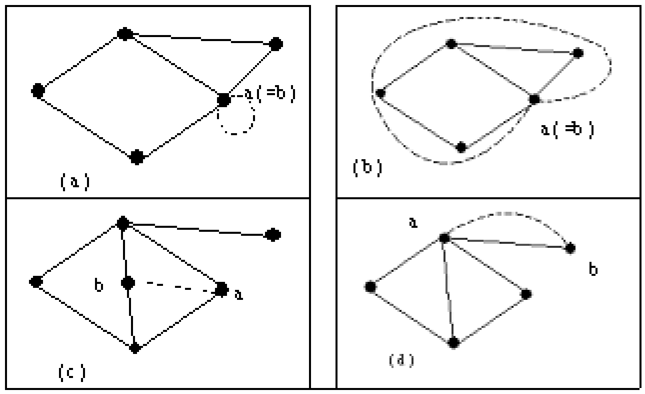
Ahora sea k Î **N** y supongamos que el resultado es verdadero para cualquier grafo o multigrafo plano conexo con e aristas ,donde 0 $\leq$ e $\leq$ k . Si G=(V,E) es un grafo o multigrafo plano conexo con v vértices , r regiones y e = k+1 aristas , sean a,b Î V con {a,b} Î E . Considere El subgrafo H de G obtenido al eliminar la arista {a,b} de G . (Si G es un multigrafo y {a,b} es una de un conjunto de aristas entre a y b, entonces la eliminamos de una sola vez.) En consecuencia , se puede escribir H=G – { a,b } o G=H + {a,b} . Consideremos los dos casos siguientes , que dependen de si H es conexo o disconexo.

Caso 1 : Los resultados de las partes (a),(b),(c) y (d) de la figura 6 muestran como un rafo G puede obtenerse de un grafo conexo H cuando se dibuja el lazo (nuevo) {a,a} como en las partes (a) y (b) o cuando la arista (nueva) {a,b} une dos vértices distintos en H como en las partes (c) y (d) . En todas estas situaciones , H tiene v vértices , k aristas y r – 1 regiones , ya que una de las regiones de H se divide en dos regiones para G . La hipótesis de inducción aplicada al grafo H indica que v – k + (r – 1) = 2 y de esto se sigue que 2 = v – (k + 1) + r = v – e + r .

Caso 2 : Ahora consideremos el caso en que G – {a,b} = H es un grafo disconexo [como se muestra en la figura 6(e) y (f)] . En este caso , H tiene v vértices , k aristas y r regiones .

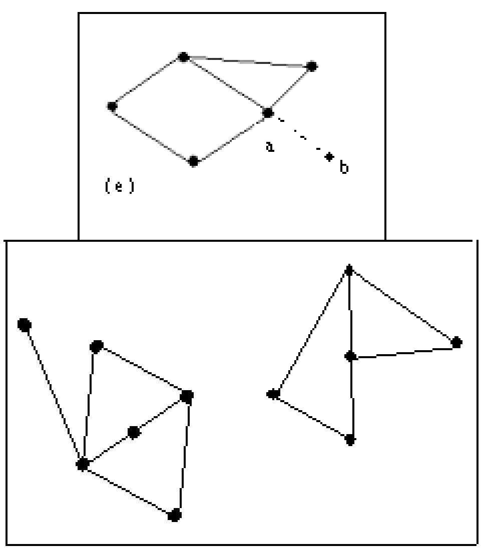
Así mismo , H tiene dos componentes H_1 y H_2 , donde Hi tiene v_i vértices , e_i aristas y r_i regiones , para i = 1,2 [la parte (e) de la figura 6 indica que una componente podría se un solo vértice aislado] . Además , $v_1 + v_2 = v$,

$e_1 + e_2 = k(= e - 1)$ y $r_1 + r_2 = r + 1$ ya que H_1 y H_2 determinan , cada uno , una región infinita . Cuando se aplica la hipótesis de inducción a H_1 y H_2 vemos que :



$v_1 - e_1 + r_1 = 2$ y $v_2 - e_2 + r_2 = 2$

La ultima idea que analizaremos para grafos planos es el concepto de grafo dual . Este concepto también en válido para grafos planos con lazos y para multigrafos planos.



Para construir un grafo dual (respecto a una inmersión particular) de un grafo o multigrafo plano G con V= {a,b,c,d,e,f} , colocamos un punto (vértice) dentro de cada región , incluyendo la región infinita , determinada por el grafo , como se muestra en la figura 7 (a) . Para cada arista compartida por las dos regiones , dibujamos una arista que conecte los vértices dentro de estas regiones .

Para una arista que se recorre dos veces en el camino cerrado alrededor de las aristas de una región , dibujamos un lazo en el vértice de esta región . En la figura6(b) , G^d es un dual del grafo G=(V,E) . Apartir de aquí haremos las siguientes observaciones :

1. Una arista en G corresponde a una arista en G^d , y viceversa.
2. Un vértice de grado 2 en G origina un par de aristas en G^d que conectan los mismos dos vértices . Por lo tanto , G^d podría ser un multigrafo . En este caso , el vértice e proporciona las aristas {a,e},{e,f} de G que originan las dos aristas que conectan v y z en G^d .
3. Dado un lazo en G, si el interior de la región (área finita) determinada por el lazo no contiene ningún otro vértice o arista de G , entonces el lazo origina un vértice colgante en G^d .
4. El grado de un vértice en G^d es el número de aristas en la frontera del camino cerrado en torn de la región en G que contiene ese vértice .

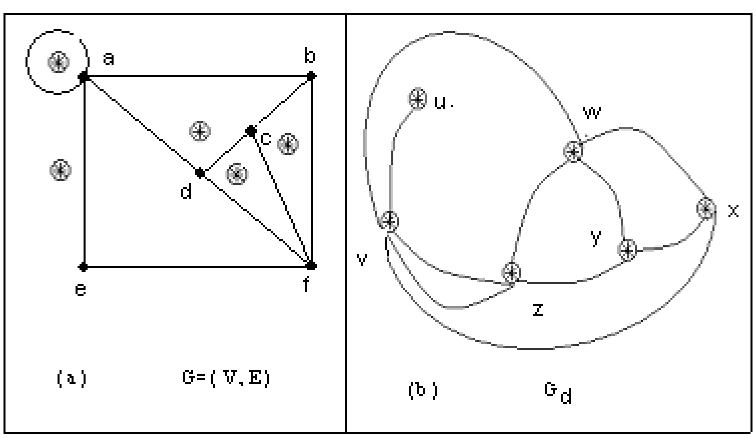


figura 7

DEFINICION 4

Sea G=(V,E) un grafo o multigrafo no dirigido . Un subconjunto E' de E es un conjunto de corte de G si al eliminaar las aristas pero no los vértices en E' de G , tenemos $k(G) < k(G')$, donde G' = (V, E – E'); pero cuando eliminamos (de E) cualquier subconjunto propio de E'' de E' , se tiene $k(G) = k(G'')$, para G'=(V , E – E'').

BIBLIOGRAFIA

- *Matemáticas Discretas y Combinatoria ;Ralph P. Grimaldi 3° edición Pretince Hall.*
- *Matematica Discretas Sexta edición Richard Johnsonbaugh; Pretince Hall.*
- *Matematicas Discretas eduard R. Sheninerman; thomson Leraning.*

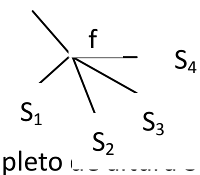
Actividades Complementarias

1.-Cuándo decimos que un grafo es plano ?

2.- Si una cara es conexa y plana y esta se traza en el plano , el plano se divide en regiones contiguas¿ Cómo se llaman esas regiones?

3.- a.- Trace los grafos de todos los árboles no isomorfos con seis vértices b.- ¿Cuántos isómeros tiene el hexano C_6H_{14}?

4.-para $m \geq 3$, podemos transformar un árbol m-ario completo en un árbol binario completo mediante la idea que se muestra en la figura



Si T es un árbol cuaternario completo [illegible] ¿Cuál es la altura máxima posible para T después de transformarlo en un árbol binario completo?¿Cual es la mínima altura?