MA3705. Algoritmos Combinatoriales 2020.
Profesor: José Soto
Escriba(s): Manuel Torres V., Vicente Cabezas M..
Fecha: 7 de Septiembre de 2020
(https://youtu.be/gqHrjFHlWn8)

Cátedra 3

1. Generación de aristas

Recordemos que:

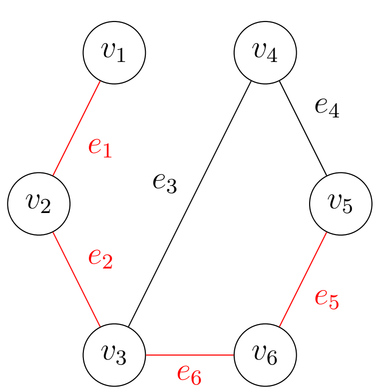
Definición 1 (Grafo generado)**.** Sea $G = (V, E)$ un grafo, y sea $F \subseteq E$.

1. Se define el *grafo generado* como
$$\text{span}(F) = \{e = uv \in E | \exists u - v \text{ camino en } F\}$$
2. Se dice que H (o bien $E(H)$) genera a G si
$$E(G) \subseteq \text{span}(E(H))$$

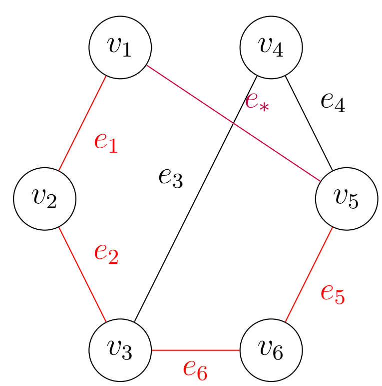
En el siguiente ejemplo se ilustra el significado de que $e \in \text{span}(F)$.

Ejemplo 1. Un arista e es generada por $F \subseteq E$ cuando existe un camino en F que une los dos extremos de la arista e, tratemos de ver esto visualmente, podemos ver que si se tiene el grafo $G = (V, E)$, donde:

1. $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$.
2. $E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$.
3. $F = \{e_1, e_2, e_5, e_6\}$, que es subconjunto de E.



Luego, para ver que $e_* = v_1v_5 \in \text{span(F)}$ pues es posible hallar algún camino contenido en F que conecte v_1 con v_6.
En la siguiente figura es posible visualizar que existe un camino $e_1e_2e_6e_5$ contenido en F que conecta los vértices v_1 y v_5, por lo tanto la arista e_* es generada por F.



También es posible tomar otro camino alternativo que conecte v_1 con v_5, pues basta la existencia de algún camino. Una alternativa sería considerar $e_1e_2e_3e_4$.

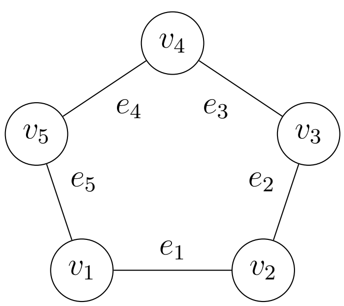
Lema 1.
1. *Si H genera a G, entonces cada componente de G está contenida en una componente de H.*
2. *Si $e = uv \in E(G)$ pertenece a algún ciclo de G, entonces $G - e$ genera a G.*

Ilustrativamente, el punto 2 del lema anterior lo podemos comprender con el siguiente ejemplo:

Ejemplo 2. Sea el grafo $G = (V, E)$, con:

1. $V = \{v_1, v_2, v_3, v_4, v_5\}$.
2. $E = \{e_1, e_2, e_3, e_4, e_5\}$.

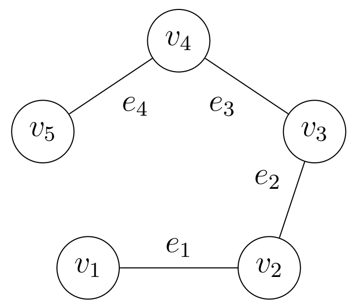
Luego es fácil notar que G posee un ciclo, el grafo anterior se puede representar de forma pictórica como:



Luego al quitarle una arista, o sea $G-e_{=}(V, E-e_5)$, donde:

$$E - e_5 = \{e_1, e_2, e_3, e_4, e_5\} \setminus \{e_5\}$$

Lo que visualmente queda como:

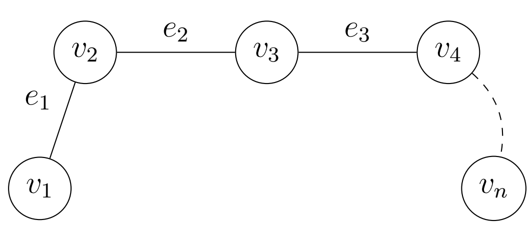


Luego es directo que el grafo anterior genera a G, pues $e_5 \in \operatorname{span}(E(G))$

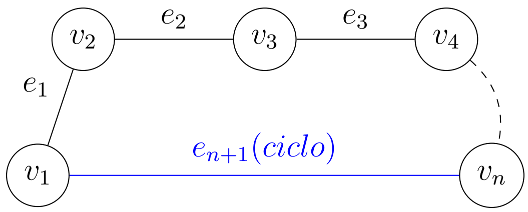
Notación. Sea $F \subseteq \binom{V}{2}$. Diremos que F es árbol, bosque, camino, ciclo, etc. Cuando (V, F) es árbol, bosque, camino, ciclo, etc.

Lema 2. *Sea $G = (V, E)$ un grafo, y sean $F \subseteq E$ y $e \in E$.*
1. *Si (V, F) tiene alguna arista, y todos los grados son pares, entonces tiene un ciclo.*
2. *Si F es bosque, entonces para cada $u, v \in V$ existe a lo más un $u - v$ camino.*
3. *Si F es bosque, entonces $F + e$ tiene a lo más un ciclo $C(F, e)$.*
4. *Si F es bosque generador, entonces (V, F) tiene las mismas componentes que G y cada una es un árbol.*

Ejemplo 3 (*Punto 1, del Lema 2:*)**.** Sea el grafo $G = (V, E)$, con al menos un arista.
Sabemos por enunciado que existe al menos un arista en G, lo cual implica que necesariamente hay al menos un camino en este grafo.
Luego, llamamos P al camino más largo en G, el cual será representado de la siguiente forma:

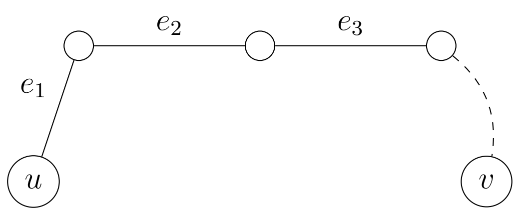


Como el grafo por enunciado tiene grado par, entonces tiene al menos grado 2. Esto quiere decir que de v_1 sale otra arista, la cual tiene que llegar a alguno de los vértices que ya existían, porque si al vértice que llegase fuese nuevo, entonces se extendería el camino, lo cual contradiría el hecho que P era el camino más largo.
Por ende nos queda algo de la forma:



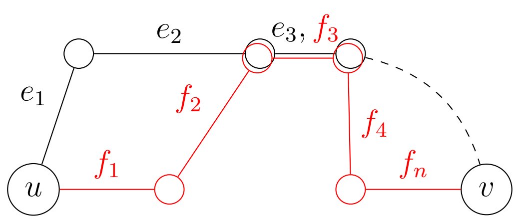
Con esto queda claro que se permitió crear un ciclo a partir del camino más largo del grafo (no necesariamente el arista tiene que llegar al último vértice del camino, pero sí debe hacer que todos los grados sean pares).

Ejemplo 4 (*Punto 2 del Lema 2:*)**.** Tomemos un conjuntos de aristas de F que son un bosque (es decir son acíclicos) y luego tomemos 2 vértices u, v, representados de la siguiente forma:

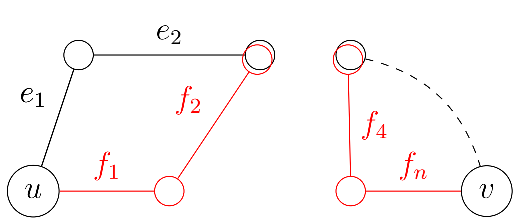


Veamos que existe a lo más un camino:

Si no hay camino, entonces claramente no hay problema y queda solucionado, pero ahora como queremos probar que hay un único camino, tomemos hacia contradicción que hay 2 caminos, los de e y f, los cuales llamaremos P Y Q respectivamente:

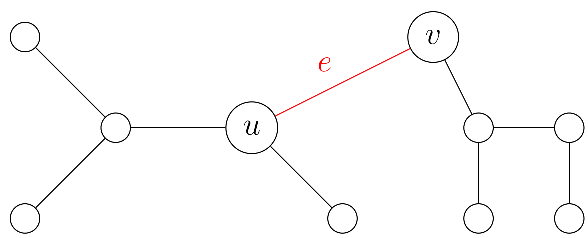


Luego veamos que P y Q se pueden intersectar, por ende conviene tomar la diferencia simétrica entre ambos, es decir $R = P \Delta Q$ ($R \subseteq F$)., lo cual nos modificaría el grafo de la siguiente forma:

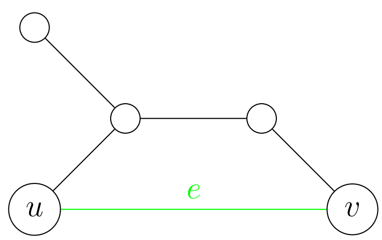


Analizando este grafo, se puede notar que R siempre va a ser distinta de vacío, pues si fuera igual, entonces P sería igual a Q, lo cual contradice que eran caminos distintos. Luego veamos los grados de cualquier vértice del grafo, los cuales llamaremos x. Si $x \neq u, v$, entonces en el camino P $\deg_P(x) = 0$ o 2 y en Q ocurriría lo mismo $\deg_Q(x) = 0$ o 2. Pero como vimos, en R hay aristas que se borran o se cancelan entre ambos caminos, sin embargo, esto da igual debido a que la paridad se mantiene en la diferencia simétrica, por ende, en cualquier caso posible, el grado de x siempre será par. El único lugar en donde se podría caer este argumento es en $x = u, v$, porque aquí en P $\deg_P(x) = 1$ y en Q $\deg_Q(x) = 1$, sin embargo el vértice sumando ambos caminos tendría en total grado 2 (o 0 si es que se cancelan por la dif. simétrica), por lo tanto para todo x el grado es par, luego por el Punto 1 del Lema 2, existe un ciclo, lo cual contradice el hecho de que habíamos tomado un bosque al comienzo.
Por lo tanto, existe a lo más un camino $u - v$.

Ejemplo 5 (*Punto 3 del Lema 2:*)**.** Sea F un bosque, al cual se le agrega un arista e. Si este arista se agrega entre 2 componentes conexas diferentes, entonces $F + e$ no tiene ciclos, ya que si creara ciclos y luego le sacáramos la arista e nos debería quedar sí o sí un camino entre $u - v$, el cual no puede existir debido a que inicialmente eran diferentes componentes conexas, como en el siguiente grafo:



Ahora, si la arista e está en una misma componente conexa, claramente va a crear un ciclo, pero el problema radica en que se necesita demostrar que este ciclo es único. Tomemos hacia contradicción que genera 2 ciclos distintos, los cuales ambos contienen a la arista e, esto significa que necesariamente tenemos 2 caminos distintos para $u - v$, P y Q, en donde $P + e$ y $Q + e$ son ciclos, pero por el Punto 2 del Lema 2, sabemos que existe a lo más un camino $u - v$ en un bosque, lo cual contradice nuestro supuesto.
Por ende existe a lo más un ciclo, el cual llamaremos $C(F, e)$.



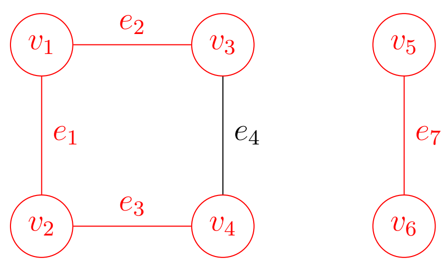
Punto 4 del Lema 2:

Si F es un bosque generador, entonces F es acíclico y además es generador de otro grafo, el cual a su vez es subgrafo de G. Luego, por el *Lema 1*, las componentes conexas de F, son las mismas que las de G. Por otro lado, los bosques tienen la propiedad conocida en la cual sus componentes conexas son un árbol, lo cual comprueba lo pedido.

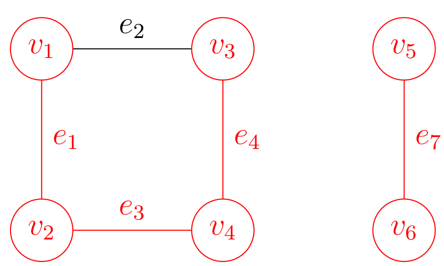
Consecuencia 1 (Intercambio)**.** Sea F bosque generador de G.
Para todo $e \in E(G)$, existe $f \in F$ tal que $F - f + e$ es bosque generador de G.
De hecho, si $e \in F$, basta tomar $f = e$.
Si $e \notin F$, basta tomar $f \in C(F, e) - e$

Observación. No se pueden agregar aristas e que junten dos componentes conexas diferentes

Ejemplo 6. Sea $G = (V, E)$ un grafo con dos componentes conexas y F un bosque generador (en rojo) como se muestra en la siguiente representación pictórica:



Note que es posible agregar un arista al bosque generador al mismo tiempo que se elimina otra arista, de tal forma que el resultado siga siendo un bosque generador, por ejemplo: Eliminando e_2 y añadiendo e_4.



También serviría para el mismo propósito eliminar cualquiera de las aristas e_1, e_2, e_3 y agregar e_4, esto debido a que se siguen conectando los vértices v_1, v_2, v_3, v_4 y además se mantendría la no existencia de ciclos en la componente conexa.

A partir de lo visto hasta ahora se tienen las herramientas necesarias para presentar el siguiente problema:

Motivación: Naturalmente nos preguntamos: *¿Es posible encontrar algún árbol generador en un grafo conexo? (o bien, un bosque en un grafo cualquiera).* La primera estrategia sería comenzar desde un vértice r y comenzar a agregar aristas sin crear ciclos, hasta que por fin todos los vértices sean alcanzados. El procedimiento anteriormente descrito corresponde a la realización de un *algoritmo de búsqueda.*

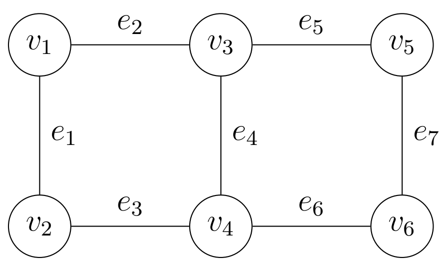
2. Algoritmos de búsqueda

Buscar árbol que cubra la componente conexa de un vértice r dado:

Algorithm 1: Algoritmo de búsqueja genérico

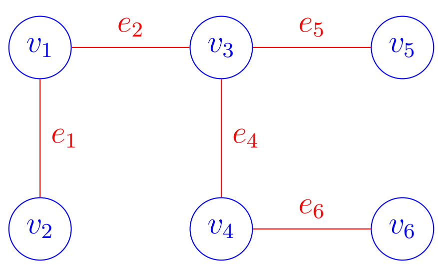
Sea $G = (V, E)$, $r \in V$;
$U \leftarrow \{r\}$// Nodos visitados $F \leftarrow \emptyset$// Aristas de solución
while $\delta(U) \neq \emptyset$ **do**
  Elegir $e = uv \in \delta(U), u \in U, v \notin U$;
  $U \leftarrow U + v$;
  $F \leftarrow F + e$;
end
return *(U.F)*;

Ejemplo 7 (Descripción del algoritmo)**.** Sea $G = (V, E)$ un grafo como se muestra a continuación:



1. *Entrada:* Escogiendo $r = v_1$,
2. Como $\delta(\{v_1\}) \neq \emptyset$:
   a) Se elige algún arista incidente en v_1: Se elige la arista $e_1 = v_1v_2 \in \delta(\{v_1\})$ tal que $v_2 \notin U$.
   b) Luego $U = \{v_1, v_2\}$.
   c) Luego $F = \{e_1\}$.
3. Como $\delta(U) \neq \emptyset$:
   a) Se elige algún arista incidente en v_1: Se elige la arista $e_2 = v_1v_3 \in \delta(\{v_1\})$ tal que $v_3 \notin U$.
   b) Luego $U = \{v_1, v_2, v_3\}$.
   c) Luego $F = \{e_1, e_2\}$.
4. Como $\delta(U) \neq \emptyset$, no quedan aristas incidentes en v_1, por lo que se pasa a v_3.
   a) Se elige algún arista incidente en v_3: Se elige la arista $e_5 = v_3v_5 \in \delta(\{U\})$ tal que $v_5 \notin U$.
   b) Luego $U = \{v_1, v_2, v_3, v_5\}$.
   c) Luego $F = \{e_1, e_2, e_5\}$.

5. Como $\delta(U) \neq \emptyset$:
   a) Se elige algún arista incidente en v_3: Se elige la arista $e_4 = v_3v_4 \in \delta(\{U\})$ tal que $v_4 \notin U$.
   b) Luego $U = \{v_1, v_2, v_3, v_5, v_4\}$.
   c) Luego $F = \{e_1, e_2, e_5, e_4\}$.
6. Como $\delta(U) \neq \emptyset$, no quedan aristas incidentes en v_3, por lo que se pasa a v_4.
   a) Se elige algún arista incidente en v_4: Se elige la arista $e_6 = v_4v_6 \in \delta(\{U\})$ tal que $v_6 \notin U$.
   b) Luego $U = \{v_1, v_2, v_3, v_5, v_6\}$.
   c) Luego $F = \{e_1, e_2, e_5, e_6\}$.
7. Luego $\delta(U) = \emptyset$, se retorna finalmente el grafo (U, F), el cual puede verificar que es un árbol en la siguiente figura:



Casos especiales de algoritmos de búsqueda

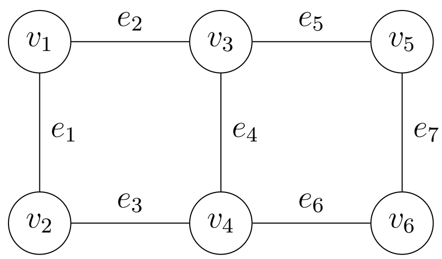
Algorithm 2: Algoritmo de búsqueja en amplitud (BFS)

Sea $G = (V, E)$, $r \in V$;
$U \leftarrow \{r\}$;
`COLA` $\leftarrow \emptyset$;
Insertar aristas de $\delta(r)$ en `COLA`;
while *COLA* $\neq \emptyset$ **do**
    Extraer primer e de `COLA`;
    if $e \in \delta(U), u \in U, v \notin U$ **then**
        $U \leftarrow U + v$;
        $F \leftarrow F + e$;
        Insertar aristas de $\delta(v)$ en `COLA`;
    end
end
return *(U.F)*;

Algorithm 3: Algoritmo de búsqueja en profundidad (DFS)

Sea $G = (V, E)$, $r \in V$;
$U \leftarrow \{r\}$;
`PILA` $\leftarrow \emptyset$;
Insertar aristas de $\delta(r)$ en `COLA`;
while *PILA* $\neq \emptyset$ **do**
    Extraer ultima e de `COLA`;
    if $e \in \delta(U), u \in U, v \notin U$ **then**
        $U \leftarrow U + v$;
        $F \leftarrow F + e$;
        Insertar aristas de $\delta(v)$ en `PILA`;
    end
end
return *(U.F)*;

Ejemplo 8 (Descripción del algoritmo)**.** Sea $G = (V, E)$ un grafo como se muestra a continuación:



1. *Entrada:* Escogiendo $r = v_1$,
2. Como $Cola = \emptyset$:
   a) Se agregan las aristas incidentes a v_1 a Cola, es decir, las aristas $e_1, e_2 \in \delta(\{v_1\})$.
   b) Luego $Cola = \{e_1, e_2\}$
   c) Luego $U = \{v_1\}$
   d) Luego $F = \emptyset$
3. Como $Cola \neq \emptyset$:
   a) Se extrae el primer arista de $Cola$. Se revisa si $e_1 \in \delta(U)$, $v_1 \in U$, $v_2 \notin U$.
   b) Luego $U = \{v_1, v_2\}$.
   c) Luego $F = \{e_1\}$.
   d) Luego $Cola = \{e_2, e_1, e_3\}$
4. Como $Cola \neq \emptyset$.
   a) Se extrae el primer arista de $Cola$. Se revisa si $e_2 \in \delta(U)$, $v_1 \in U$, $v_3 \notin U$.
   b) Luego $U = \{v_1, v_2, v_3\}$.
   c) Luego $F = \{e_1, e_2\}$.
   d) Luego $Cola = \{e_1, e_3, e_2, e_4, e_5\}$
5. Como $Cola \neq \emptyset$:
   a) Se extrae el primer arista de $Cola$. Se revisa si $e_1 \in \delta(U)$, $v_1 \in U$, $v_2 \notin U$.
   b) Luego $Cola = \{e_3, e_2, e_4, e_5\}$.
6. Como $Cola \neq \emptyset$.
   a) Se extrae el primer arista de $Cola$. Se revisa si $e_3 \in \delta(U)$, $v_2 \in U$, $v_4 \notin U$.
   b) Luego $U = \{v_1, v_2, v_3, v_4\}$.
   c) Luego $F = \{e_1, e_2, e_3\}$.
   d) Luego $Cola = \{e_2, e_4, e_5, e_3, e_4, e_6\}$
7. Como $Cola \neq \emptyset$.

   a) Se extrae el primer arista de $Cola$. Se revisa si $e_2 \in \delta(U)$, $v_1 \in U$, $v_3 \notin U$.

   b) Luego $Cola = \{e_4, e_5, e_3, e_4, e_6\}$

8. Como $Cola \neq \emptyset$.

   a) Se extrae el primer arista de $Cola$. Se revisa si $e_4 \in \delta(U)$, $v_3 \in U$, $v_4 \notin U$.

   b) Luego $Cola = \{e_5, e_3, e_4, e_6\}$

9. Como $Cola \neq \emptyset$.

   a) Se extrae el primer arista de $Cola$. Se revisa si $e_5 \in \delta(U)$, $v_3 \in U$, $v_5 \notin U$.

   b) Luego $U = \{v_1, v_2, v_3, v_4, v_5\}$.

   c) Luego $F = \{e_1, e_2, e_3, e_5\}$.

   d) Luego $Cola = \{e_3, e_4, e_6, e_5, e_7\}$

10. Como $Cola \neq \emptyset$.

    a) Se extrae el primer arista de $Cola$. Se revisa si $e_3 \in \delta(U)$, $v_2 \in U$, $v_4 \notin U$.

    b) Luego $Cola = \{e_4, e_6, e_5, e_7\}$

11. Como $Cola \neq \emptyset$.

    a) Se extrae el primer arista de $Cola$. Se revisa si $e_4 \in \delta(U)$, $v_3 \in U$, $v_4 \notin U$.

    b) Luego $Cola = \{e_6, e_5, e_7\}$

12. Como $Cola \neq \emptyset$.

    a) Se extrae el primer arista de $Cola$. Se revisa si $e_6 \in \delta(U)$, $v_4 \in U$, $v_6 \notin U$.

    b) Luego $U = \{v_1, v_2, v_3, v_4, v_5, v_6\}$.

    c) Luego $F = \{e_1, e_2, e_3, e_5, e_6\}$.

    d) Luego $Cola = \{e_5, e_7, e_6, e_7\}$

13. Como $Cola \neq \emptyset$.

    a) Se extrae el primer arista de $Cola$. Se revisa si $e_5 \in \delta(U)$, $v_3 \in U$, $v_5 \notin U$.

    b) Luego $Cola = \{e_7, e_6, e_7\}$

14. Como $Cola \neq \emptyset$.

    a) Se extrae el primer arista de $Cola$. Se revisa si $e_7 \in \delta(U)$, $v_5 \in U$, $v_6 \notin U$.

    b) Luego $Cola = \{e_6, e_7\}$

15. Como $Cola \neq \emptyset$.

    a) Se extrae el primer arista de $Cola$. Se revisa si $e_6 \in \delta(U)$, $v_4 \in U$, $v_6 \notin U$.

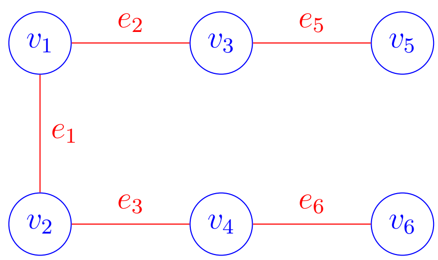
   b) Luego $Cola = \{e_7\}$

16. Como $Cola \neq \emptyset$.

   a) Se extrae el primer arista de $Cola$. Se revisa si $e_7 \in \delta(U)$, $v_5 \in U$, $v_6 \notin U$.

   b) Luego $Cola = \{\}$

17. Luego $Cola = \emptyset$, se retorna finalmente el grafo (U, F), el cual se representa en la siguiente figura:



El algoritmo DFS, se hace de forma análoga, solo que en vez de tomar el primer elemento de $Cola$, se toma el último.

Además de este algoritmo, se desprenden dos proposiciones, en el cual uno de ellos dice que cada arista aparece dos veces en $Cola$, y el otro dice que la distancia para llegar a cualquier vértice, en el grafo inicial, es la misma para llegar al vértice en el árbol final.