Universidad Nacional Autónoma de México

Facultad de Ciencias

Tarea 10

Presentan

- Nelson Osmar Garcia Villa - 322190357
- Valeria Camacho Hernández - 322007273
- Mauricio Casillas Álvarez - 322196342

Asignatura

Gráficas y Juegos 2025-2

Profesor

César Hernández Cruz

Ayudante

Iñaki Cornejo de la Mora

Fecha

Viernes 06 de junio del 2025

1. Una *oruga* es un árbol tal que al borrar todas las hojas se obtiene una trayectoria. Demuestre que un árbol es una oruga si y sólo si no contiene a la garra subdividida (Figura 1) como subgráfica inducida.

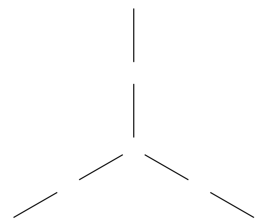

Figure 1: Garra subdividida.

Respuesta: Procedemos a probar ambas implicaciones.

$\Longrightarrow$) Supongamos que T es una oruga. Entonces existe una trayectoria P tal que todo vértice de T que no pertenece a P es una hoja adyacente a algún vértice de P (por definición de oruga).

Supongamos por contradicción que T contiene a la garra subdividida como subgráfica inducida.

Como la garra subdividida es un vértice central del cual salen tres caminos internos disjuntos (solo comparten al vértice central), cada uno con longitud de al menos 1 y todos terminan en hoja, y como en una orgua todos los vértices que no son hoja pertenecen a la trayectoria P y todas las hojas están directamente conectadas a vértices de P, entonces no puede existir en T un vértice desde el cual partan tres caminos disjuntos de longitud de al menos 1, pues desde cualquier vértitce de P, a lo mucho puede salor un camino de longitud hacia una hoja. Esto, contradice la existencia de una garra subdividida como subgráfica inducida.

$\therefore$ El árbol T es una oruga por lo que no puede contener a la garra subdividida como subgráfica inducida.

$\Longleftarrow$) Sea T un árbol que no contiene a la garra subdividida como gráfica inducida. Procedemos por inducción sobre el $n \geq 1$ número de vértices del árbol y eliminando vértices.

Casos Base:

- Sea $n = 1$. T es un vértice aislado. Entonces trivialmente es una oruga.
- Sea $n = 2$. T una arista. Al borrar sus dos hojas, queda vacío (trayectoria trivial).
- Sea $n = 3$. T es una trayectoria. Al borrar las hojas queda un vértice, entonces es una oruga.

Hipótesis: Supongamos que para todo árbol con k vértices que no contiene la garra subdividida como subgráfica inducida, se cumple que es una oruga.

Sea x una hoja de T y sea z su único vecino. Procedemos a quitar a x para obtener al árbol $T' = T - \{x\}$ con k vértices.

Como quitar una hoja no crea nuevas subgráficas inducidas, si T no contiene a la garra subdividida, entonces T' tampoco la contiene. Por hipótesis inductiva, T' es una oruga.

Ahora, sea P la trayectoria central de T'. Si añadimos de nuevo la hoja x al vértice z, como x es una hoja, entonces solo se conecta a z. Ahora tenemos dos casos por revisar:

- Caso 1: Si $z \in P$, entonces al agregar a x lo que sucede que es añadimos una hoja a la trayectoria central. Es decir, es una hoja conectada a la trayectoria. Entonces T claramente es una oruga.
- Caso 2: Si $z \notin P$. Tenemos los siguientes dos casos:
  - Caso 2.1: z es una hoja en T', osea es un extremo en P. Entonces al agregar x, sucede que z ahora tiene grado 2 (está conectado a x, es su vecino, y es vecino a otro vértice), y puede incorporarse o quedar como extremo de la trayectoria central (como hoja), es decir, podemos extender la trayectoria central P de T' incluyendo a z, osea alargar un extremo de P hacia z. Luego, x queda como una hoja unida a z, que ahora está en la nueva trayectoria central. Entonces, sigue siendo una oruga.
  - Caso 2.2: z no es hoja pero está conectada, osea no era extremo en P. Entonces z se añade como una rama, pero como z solo tenía grado 2, puede ser que esa rama fuera rama de otra rama. Esto contradice que T es una oruga pues z no sería hoja. Entonces solo consideramos posible al caso 2.1 y no este.

En ambos casos, se cumple la estructura de oruga.

$\therefore$ El árbol T no contiene a la garra subdividida como subgráfica inducida, por lo que es una oruga.

Por lo tanto, como ambas implicaciones se cumplen, podemos afirmar que un árbol es una oruga si y sólo si no contiene a la garra subdividida como subgráfica inducida. ★

2. Sea G un árbol con bipartición (X, Y). Demuestre que si $|X| \leq |Y|$, entonces G tiene una hoja en Y, y si $|X| = |Y|$, entonces hay una hoja en X y una en Y.

   Respuesta: Procedemos por doble implicación.

   $\Longrightarrow$) Si $|X| \leq |Y|$, entonces G tiene una hoja en Y.

   Supongamos por contradicción que G no tiene hojas en Y. Entonces, todo vértice en Y tiene grado al menos 2. Como G es un árbol, el número de aristas es:

   $$|E(G)| = |X| + |Y| - 1.$$

   La suma de los grados de los vértices en X es igual a $|E(G)|$:

   $$\sum_{v \in X} \deg(v) = |X| + |Y| - 1.$$

   La suma total de grados en G es $2|E(G)| = 2(|X| + |Y| - 1)$. Por hipótesis, los vértices en Y tienen grado al menos 2:

   $$\sum_{v \in Y} \deg(v) \geq 2|Y|.$$

   Combinando estas expresiones:

   $$(|X| + |Y| - 1) + 2|Y| \leq 2(|X| + |Y| - 1),$$

   lo que implica:

   $$|X| + 3|Y| - 1 \leq 2|X| + 2|Y| - 2 \implies |Y| - 1 \leq |X|.$$

   Esto contradice $|X| \leq |Y|$ a menos que $|X| = |Y|$ o $|X| = |Y| - 1$. Si $|X| < |Y| - 1$, la desigualdad falla, por lo que debe existir al menos una hoja en Y.

   $\Longleftarrow$) Si $|X| = |Y|$, entonces hay una hoja en X y una en Y.

   Aplicamos el primer resultado dos veces:

   - Como $|X| \leq |Y|$, existe al menos una hoja en Y.
   - Como $|Y| \leq |X|$, existe al menos una hoja en X.

   Por lo tanto, hay al menos una hoja en cada conjunto.

   Ambas direcciones se cumplen. ■

3. Dada una digráfica D, un *orden topológico* de D es un orden lineal $v_1 < \cdots < v_n$ de su conjunto de vértices de tal forma que si $(v_i, v_j) \in A_D$, entonces $i < j$.

   (a) Demuestre que una digráfica tiene un orden topológico si y sólo si no contiene ciclos dirigidos.

   Respuesta: Procedemos a demostrar ambas implicaciones.

   $\Longrightarrow$) Sea D una digráfica. Supongamos D tiene un orden topológico en sus vértices. Esto significa que para toda flecha (v_i, v_j), se cumple que $i < j$, es decir, las flechas solo están dirigidas en orden.

   Por demostrar: D es acíclica.

   Procedemos por contradicción. Supongamos que D tiene un ciclo $(v_1, v_2, \ldots, v_k, v_1)$. Como es un ciclo, los vértices se repiten, y en particular v_1 aparece al inicio y al final. Entonces existe una flecha (v_k, v_1) con la que deberíamos tener $k < 1$, sin embargo, no cumple el orden topológico (de hipótesis), ya que no es posible que una flecha este orientada de regreso, pues esta debe de orientarse a un siguiente vértice. Así, no puede haber ciclos en D.

   $\therefore$ Es acíclica.

   $\Longleftarrow$) Sea D una digráfica acíclica.

   Por demostrar: D tiene un orden topológico.

   Procedemos por inducción sobre el número de vértices de D.

   Casos base:

   - Caso 1: Sea $n = 1$, D tiene un sólo vértice y ningún arco, entonces el orden se cumple inmediato.
   - Caso 2: Sea n = 2, entonces tenemos dos casos:
     - Si no hay arcos, entonces el orden (o cualquier otro) se cumple inmediato.
     - Si hay un arco (v_1, v_2), entonces el orden es v_1, v_2 , el cual cumple $i < j$ pues v_1 debe aparecer antes de v_2 .

   Hipótesis de Inducción: Supongamos que toda digráfica acíclica con $n = m$ vértices admite un orden topológico.

   Paso Inductivo: Sea D una digráfica sin ciclos con $m + 1$ vértices.

   Por demostrar: Toda digráfica sin ciclos dirigidos con $n = m + 1$ vértices también tiene un orden topológico.

   Como D es aciclica (por hipótesis), entonces debe existir al menos un vértice v con $ingrado = 0$, es decir, es una fuente (adem. De lo contrario, si todos los vértices tuvieran al menos un arco entrante, podríamos construir un ciclo dirigido siguiendo los arcos entrantes, lo que contradice la hipótesis.

   Ahora construyamos una nueva digráfica D'. Eliminamos el vértice v y todos sus arcos salientes, con lo que obtenemos a D' , la cual tiene m vértices y es acíclica, pues eliminar vértices no crea ciclos.

Por H.I, D' tiene un orden topológico $(v_1, v_2, ..., v_m)$ en los vértices de D' tal que todo arco (v_i, v_j) cumple $i < j$.

Ahora, insertemos el vértice v al inicio del orden. Entonces tenemos $(v, v_1, v_2, ..., v_m)$. Como v es una fuente, no hay arcos entrantes a el en D. En particular, los únicos posibles arcos que lo involucran son de la forma (v, v_i) y estos se orientan hacia un siguiente vértice en el orden, pues v aparece antes que cualquier otro vértice.

$\therefore D$ tiene un orden topológico.

Por lo tanto, ambas implicaciones se cumplen. ★

(b) Exhiba un algoritmo de tiempo $O(|V| + |E|)$ para determinar si una digráfica tiene un orden topológico. En caso de existir, el algoritmo debe devolver el orden topológico. En caso de no existir, el algoritmo debe devolver un ciclo dirigido.

Respuesta: Proponemos el siguiente algoritmo, en el que iteramos sobre los vértices: eliminamos aquellos con ingrado 0 (las fuentes), uno por uno, actualizamos los grados de los demás vértices, y los agregamos al orden.

Para ello, para una parte implementamos una modificación a DFS para obtener el orden topológico, y para otra implementamos el algoritmo de Kahn para detectar si hay un ciclo, lo cual nos indica si el orden topológico existe.

Durante el algoritmo de Kahn, vamos sacando vértices con ingrado 0, es decir las fuentes, las cuales podemos poner al principio del orden, los eliminamos y actualizamos los ingrados de sus vecinos pues ya no será una entrada para ellos y así vamos generando nuevas fuentes, luego, se construye una lista orden con ellos, la cual, si al final tiene menos vértices que la cantidad total, significa que quedaron vértices atrapados en un ciclo (porque nunca bajaron a ingrado 0). En ese caso, no hay orden topológico posible.

Como vemos, el algoritmo de Kahn es ideal para intentar construir el orden y detectar si no existe, pero no nos dice cuál es el ciclo, y como queremos saber el ciclo explícitamente, entonces implementamos DFS a los vértices, con el cual tenemos tres posibilidades: el vértice no ha sido visitado entonces seguimos con DFS recursivo sobre él; el vértice ya fue visitado y ya salió de la recursión entonces no hacemos nada; o el vértice ya fue visitado y aún está en la pila de la recursión, es decir, estamos volviendo a visitar un padre.

El algoritmo es el siguiente:

Input: Una digráfica acíclica G, sus V vértices y A arcos.

Output: Una lista con los vértices en orden topológico.

1. orden $\longleftarrow$ [ ] ; Inicializamos una lista vacía para guardar el orden topológico.
2. ingrado $\longleftarrow$ [ ] ; Creamos una lista ingrado para guardar cuántas flechas entran a cada vértice, y los inicializamos en 0.
3. fuente $\longleftarrow$ [ ] ; Creamos una lista fuente para guardar los ingrados.
4. for each vértice v en $V(G)$ do

4. if ingrado[v] $\longleftarrow$ 0 then
5. añadir v a fuente
6. for each arco (u, v) en G do ; Recorremos todos los arcos (u, v) de G y actualizamos los ingrados.
6. ingrado[v] $\longleftarrow$ ingrado[v] +1
7. while fuente $\neq \emptyset$ do; Ahora trabajamos con los vértices que tienen $ingrado = 0$ (fuentes). Esto se debe a que en toda digráfica acíclica siempre existe al menos un vértice sin flechas entrantes (una fuente), pues si no existiera tal vértice, podríamos construir un ciclo.
8. añadir v a [orden] ; Añadimos al vértice a la lista [orden]
9. eliminar v de fuente ; Eliminamos al vértice de los que tienen $ingrado = 0$ para marcarlo como procesado.
10. for each vecino de w en $V(G)$ do ;
11. ingrado[w] $\longleftarrow$ ingrado[w] -1 ; Actualizamos los ingrados disminuyendo su ingrado en 1 para cada vértice al que apunta v.
12. if ingrado[w] = 0 then
13. añadir w a fuente ; Como ya tiene ingrado = 0, actualizamos la lista añadiendolo.
14. if longitud de [orden] = $|V(G)|$ then ; Verificamos si el orden es válido, es decir, si procesamos todos los vértices.
15. return [orden] ; El orden es válido, no hay ciclos. Entonces procedemos a devolver el orden topologico.
16. ciclo $\longleftarrow$ encontrarCiclo(G, ingrado) ; Para devolver el orden, le pasamos la gráfica y los ingrados.
17. return ciclo

Función encontrarCiclo:
1. for each vértice v en G do
2. visitado[v] $\longleftarrow$ false
3. enCamino[v] $\longleftarrow$ false
4. padre[v] $\longleftarrow \emptyset$
5. for each vértice v en G do
6. if ingrado[v]$>$ 0 y no visitado[v] then
7. if dFS(v) = $true$ then
8. return ciclo
9. return [ ] ; Entonces no hay ciclos

Función dFS
1. visitado[v] $\longleftarrow$ true
2. enCamino[v] $\longleftarrow$ true
3. for each vecino w en G do
4. if no visitado[w] then
5. padre[w] $\longleftarrow v$
6. if dfs(w) = $true$ then

```
            return true
    if enCamino(w) then
        ciclo ⟵ w
        x ⟵ v
        while x ≠ w do
            insertar x al inicio de ciclo
            x ⟵ padre(x)
        insertar w al inicio de ciclo
        return true
enCamino(v) ⟵ false
return false
```

En términos de complejidad, el lgoritmo principal es $O(|V|+|E|)$ (Kahn), mientras que la detección de ciclos es $O(|V|+|E|)$ (DFS), por lo que el total es $O(|V|+|E|)$.
★

4. Sea D una digráfica en la que $d^-(v) = d^+(v)$ para cada vértice v de D, excepto para un par de vértices x y y, en los que $d^+(x) - d^-(x) = k = d^-(y) - d^+(y)$. Demuestre que D contiene k xy-trayectorias dirigidas ajenas por flechas

   Respuesta: Sea D una digráfica donde $d^-(v) = d^+(v)$ para todo vértice $v \neq x, y$, y $d^+(x) - d^-(x) = k = d^-(y) - d^+(y)$. Demostramos que D contiene k xy-trayectorias dirigidas ajenas por flechas.

   Construimos una digráfica D' añadiendo:

   - Un vértice s con k aristas dirigidas de s a x.
   - Un vértice t con k aristas dirigidas de y a t.

   En D', todos los vértices excepto s y t tienen grados balanceados ($d^-(v) = d^+(v)$), mientras que:

$$d^+(s) - d^-(s) = k \quad \text{y} \quad d^-(t) - d^+(t) = k.$$

   Por el Teorema de Menger para digráficas, D' contiene k st-trayectorias dirigidas ajenas por aristas. Estas trayectorias deben pasar por x y y, pues s solo incide en x y t solo es incidido por y. Eliminando s y t, obtenemos k xy-trayectorias en D ajenas por flechas. ■

5. Sean G una gráfica conexa y x un vértice de G. Un árbol generador T de G es llamado un x-árbol de distancia si para todo vértice y de G se tiene que $d_T(x,y) = d_G(x,y)$.

   (a) Sin usar BFS, demuestre que para cualquier vértice x, la gráfica G tiene un x-árbol de distancia.

   Respuesta: Sea G una gráfica conexa y elegimos un vértice arbitrario $x \in V(G)$

   Comenzamos considerando únicamente el vértice x incluido en el árbol, es decir, el conjunto inicial de vértices del árbol es $V_0 = \{x\}$ y el conjunto de aristas es $E_0 = \emptyset$.

   Mientras $V_0 \neq V(G)$:

   - Elegimos una arista $\{u,v\} \in E(G)$ tal que: $u \in V_0$, $v \notin V_0$ y $d_G(x,v) = d_G(x,u) + 1$
   - Incorporamos el vértice v al conjunto V_0 y la arista $\{u,v\}$ al conjunto E_0.

   La gráfica $T = (V, E_0)$ obtenido es conexo, acíclico y contiene todos los vértices, por lo tanto es un árbol generador.

   Cada vértice v fue conectado mediante una arista a un vértice u, es decir, $d_G(x,v) = d_G(x,u) + 1$.■

   (b) Demuestre que toda gráfica conexa de diámetro d tiene un árbol generador de diámetro a lo más $2d$.

   Respuesta: Sea $u, v \in V(G)$ vértices cualesquiera.

   El camino en T entre u y v puede ser recorrido pasando por x, pues dado que que T es un x-árbol de distancia, no necesariamente conserva todas las distancias entre pares arbitrarios de vértices (u,v) pero sí conserva las distancias desde x: $d_T(u,v) \leq d_T(u,x) + d_T(x,v)$.

   Usando que T es un x-árbol de distancia, se tiene: $d_T(u,v) \leq d_G(u,x) + d_G(x,v)$.

   Como el diámetro de G es d, se cumple que: $d_G(u,x) \leq d$, $\quad d_G(x,v) \leq d$.

   Por lo tanto, $d_T(u,v) \leq d + d = 2d$.

   Como u y v eran arbitrarios, se concluye que el diámetro de T satisface $d_T \leq 2d$.■