Grafos de precedencia y ordenamiento topológico

Guillermo Palma

Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información

Plan

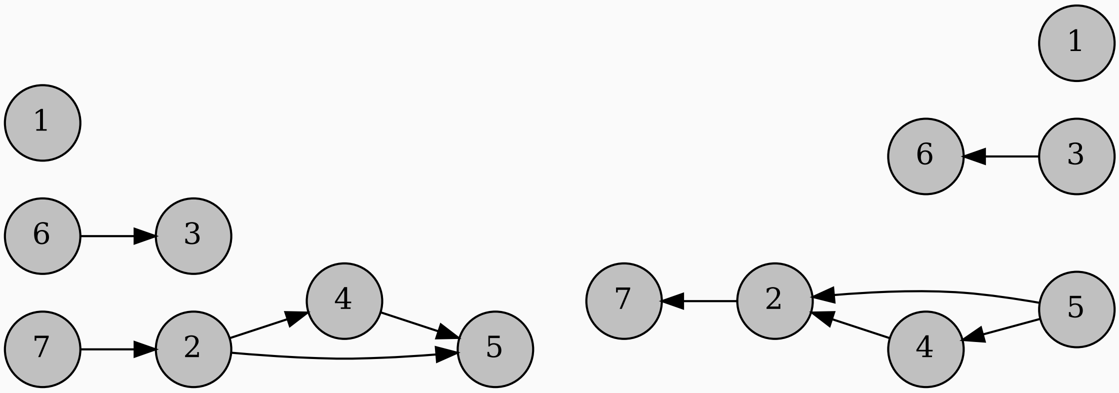
Preliminares

Digrafo inverso

Definición de digrafo inverso

Un digrafo inverso, que denotamos como G^{-1}, es un digrafo que se obtiene de cambiar la dirección de los arcos de un digrafo G.

Ejemplo de un digrafo inverso



(a) Grafo dirigido G (b) Digrafo G^{-1} de G.

Figura 1: Se muestra un digrafo G y su digrafo inverso G^{-1}.

Vértice fuente y sumidero de un digrafo

Definición de vértice fuente

Un vértice v de un digrafo G es un vértice fuente, si el grado interior de v en G es igual a cero.

Definición de vértice sumidero

Un vértice v de un digrafo G es un vértice sumidero, si el grado exterior de v en G es igual a cero.

Ejemplo de vértice sumidero y fuente en un digrafo

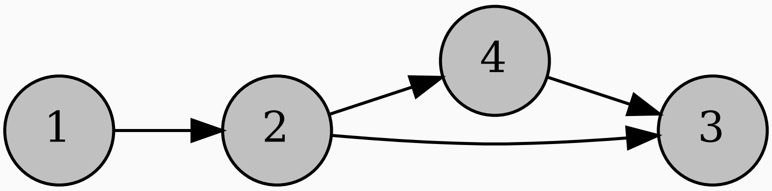


Figura 2: Digrafo con vértice fuente 1 y con vértice sumidero 3.

Nivel y altura de un vértice

Definición de nivel de un vértice

El nivel de un vértice v en un digrafo G, corresponde a la longitud del camino simple más largo en G que termina en v.

Definición de altura de un vértice

La altura de un vértice v en un digrafo G, corresponde a la longitud del camino simple más largo en G que comienza en v.

Ejemplo de nivel y altura

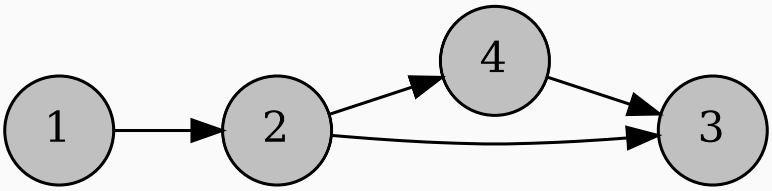


Figura 3: En este digrafo el vértice 4 tiene un nivel de 2 y una altura de 1

Grafos de precedencias

Grafo acíclico directo

Definición de grafo acíclico directo

Un grafo acíclico directo, en inglés *directed acyclic graph* (DAG), es un digrafo sin ciclos.

Grafo de precedencia

Definición grafo de precedencia

Un grafo de precedencia es un grafo acíclico directo o DAG.

Ejemplos de un grafo de precedencia

- Pasos secuenciales para realizar una tarea con junto con la relación de precedencia entre los pasos.
- Plan de estudios con sus requisitos de materias.
- Representación de fórmulas matemáticas.
- Grafo de una relación de orden parcial.
- Representación de la planificación de un proyecto.

Ejemplo de un grafo de precedencia

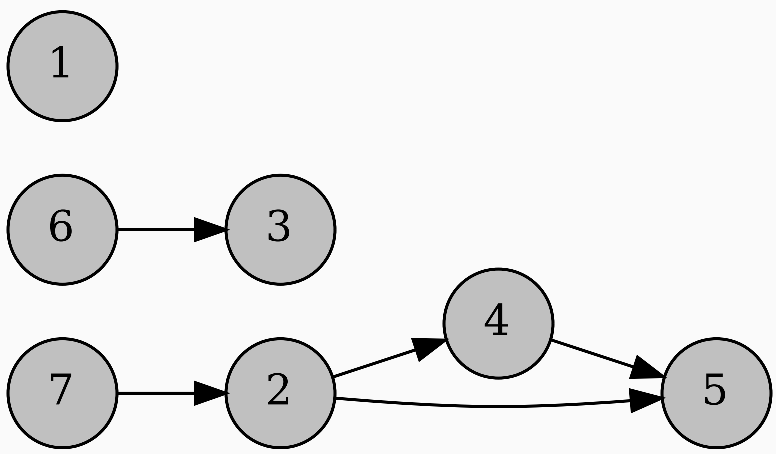


Figura 4: Ejemplo de un grafo de precedencia.

Ejemplo de un grafo de precedencia, continuación

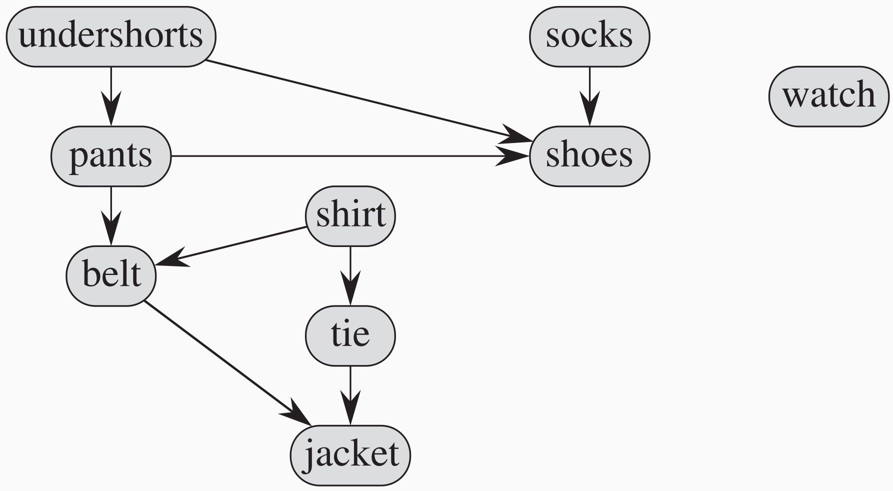


Figura 5: Ejemplo de un grafo de precedencia que representa el orden en que se colocan prendas de vestir. El lado (u, v) indica que la prenda u, se debe colocar antes de la prenda v. Fuente [1].

Ejemplo de un grafo de precedencia, continuación

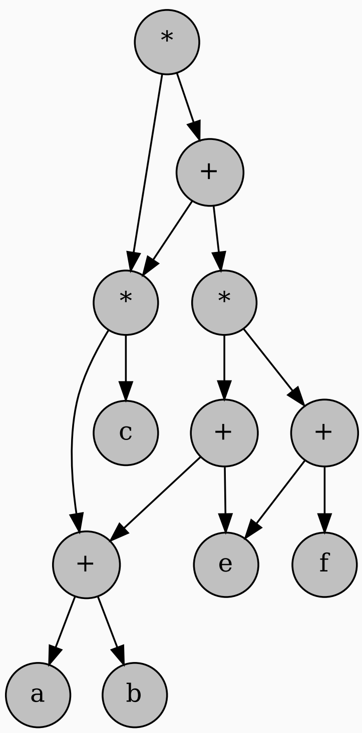


Figura 6: Ejemplo de un grafo de precedencia que representa la fórmula $((a + b) * c + ((a + b) + e) * (e + f)) * ((a + b) * c)$. Fuente [3].

Ejemplo de un grafo de precedencia, continuación

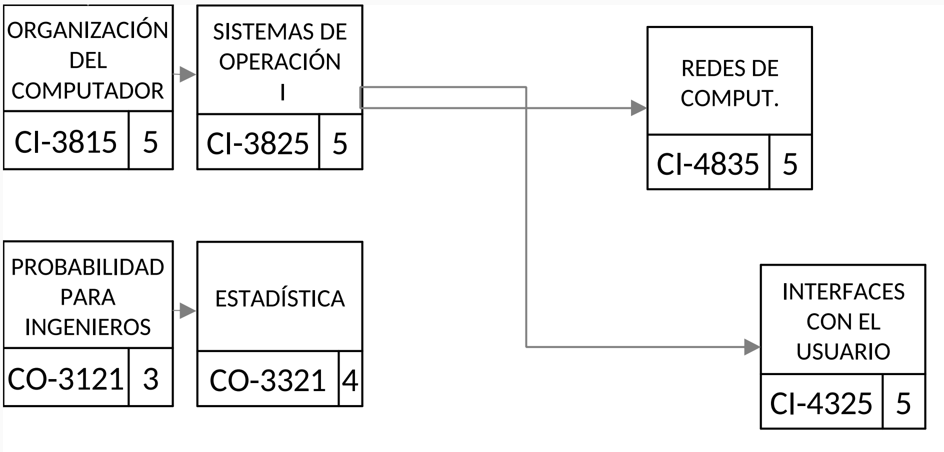


Figura 7: Ejemplo de un grafo de precedencia que representa materias de un curso y sus requisitos para cursarlas. Fuente [2].

Propiedades de los grafos de precedencia

1. Un digrafo G no posee ciclos si solo si, todo subgrafo de G no posee ciclos.
2. Un digrafo G no posee ciclos si solo si, el digrafo inverso G^{-1} no posee ciclos.

Componentes fuertemente conexas en un DAG

Proposición 1

Sea G un digrafo sin bucles. Se tiene que G no posee ciclos si y solo si, toda componente fuertemente conexa de G está compuesta por un solo vértice.

Prueba

- ($\Rightarrow$) Si una componente fuertemente conexa tiene más de un vértice, entonces necesariamente tiene que haber un ciclo para que todo vértice u sea alcanzable desde un vértice v, tal que $u \neq v$.
- ($\Leftarrow$) Si G tiene un ciclo, entonces este estará en la componente fuertemente conexa.

$\square$

Caminos en un DAG

Proposición 2

Un digrafo G es un grafo de precedencia si solo si todo camino entre dos de sus vértices, es un camino simple.

Prueba

Si hay un camino que no es simple, entonces se repiten vértices en el camino, por tanto hay ciclo en el camino. $\square$

Vértices fuentes y sumidero en DAG

Proposición 3

Todo DAG tiene un vértice fuente y un vértice sumidero

Caminos más largos en un DAG

Proposición 4

En un grafo de precedencia G, para todo vértice v de G, el vértice inicial del camino más largo que termina en v, es un vértice fuente de G.

Prueba

Si v es fuente, entonces se cumple. Sea w, tal que $w \neq v$ y el camino más largo comienza en w. Entonces el predecesor de w de pasa por el camino, pero como es un grafo de precedencia, entonces no hay ciclos y w no tienes predecesor, por lo tanto es vértice fuente. $\square$

Caminos más largos en un DAG, continuación

Proposición 5

En un grafo de precedencia G, para todo vértice v de G, el vértice terminal del camino más largo que comienza en v, es un vértice sumidero de G.

Prueba

Si a partir de G obtenemos G^{-1} y luego aplicamos Proposición 4, obtenemos que el camino más largo comienza en el vértice w que es un vértice fuente. Luego si a G^{-1} le obtenemos el digrafo inverso, entonces tenemos de nuevo G y en consecuencia w se convierte en un vértice sumidero $\square$

Partición de un DAG

Proposición 6

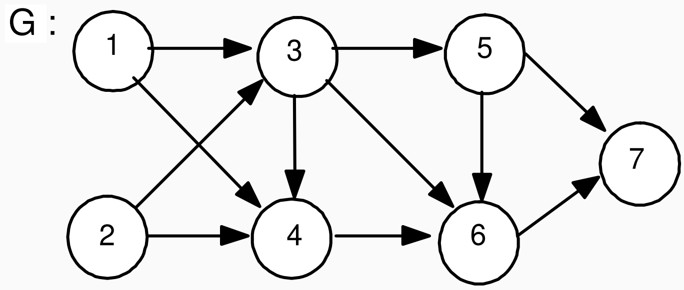
$G = (V, E)$ es un grafo de precedencia si solo si, se puede realizar una partición de V en conjuntos $V_0, V_1, \ldots, V_n$, tal que $\forall i, 0 \leq i \leq n$ se tiene que $v \in V_i$ si solo si, el camino más largo que termina en v tiene una longitud de i.

Partición en niveles de un DAG

Definición

A la partición de un grafo de precedencia $G = (V, E)$ en conjuntos $V_0, V_1, \ldots, V_n$, por la Proposición 6, la llamamos partición en niveles de un DAG. Este nombre se debe a que se tiene que para cada $v \in V_i$, se cumple que el nivel de v es i.

Ejemplo de la partición de niveles en un DAG



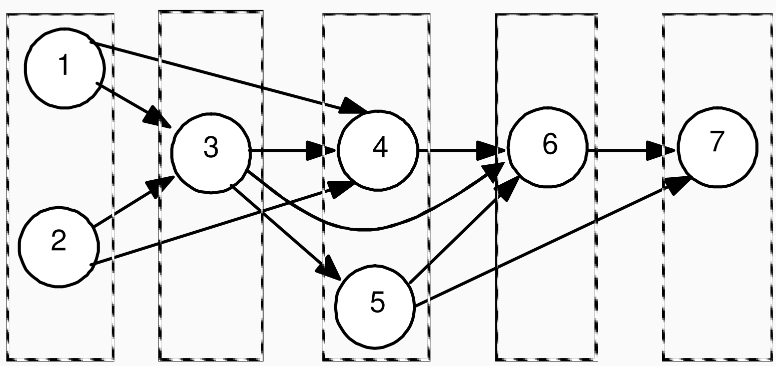


Figura 8: El grafo de precedencia contiene 5 particiones. Fuente [3]

Algoritmo de partición de niveles de un DAG

Función Particion-Niveles($G(V, E)$ sin arcos múltiples, *GradoInterior*)

1 **inicio**
2   **para** $i \leftarrow 0$ **a** $|V| - 1$ **hacer** $Particiones[i] \leftarrow \emptyset$;
3   $nvert \leftarrow 0$; $nivel \leftarrow 0$; $hayCiclo \leftarrow false$;
4   **para cada** $u \in V$ **hacer**
5   **si** $GradoInterior[u] = 0$ **entonces**
6       $Particiones[nivel].agregar(u)$;
7       $nvert \leftarrow nvert + 1$
8   **mientras** $(nvert < |V|) \wedge (Particiones[nivel] \neq \emptyset)$ **hacer**
9   **para cada** $u \in Particiones[nivel]$ **hacer**
10   **para cada** $v \in G.adyacentes[u]$ **hacer**
11         $GradoInterior[v] \leftarrow GradoInterior[v] - 1$;
12   **si** $GradoInterior[v] = 0$ **entonces**
13           $Particiones[nivel + 1].agregar(v)$;
14           $nvert \leftarrow nvert + 1$
15     $nivel \leftarrow nivel + 1$
16   $hayCiclo \leftarrow (nvert < |V|)$;
17   **devolver** $(Particiones, hayCiclo)$

Análisis del algoritmo de partición de niveles de un DAG

- Línea 2 inicialización de los conjuntos de *Particiones* es $O(|V|)$
- Línea 4-7 detección de vértices fuentes es $O(|V|)$
- Líneas 8-15 es $O(\max(|V|, |E|))$
- Por lo tanto, el tiempo del peor caso es $O(\max(|V|, |E|))$

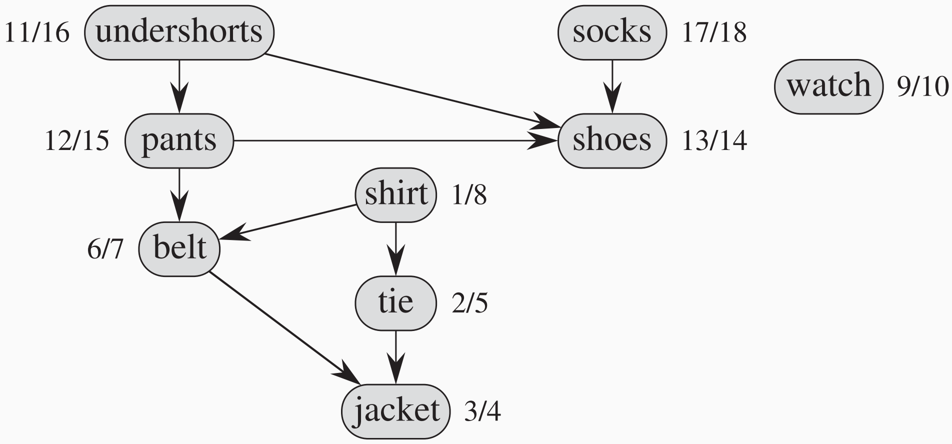
Ordenamiento topológico

Ordenamiento topológico de un DAG

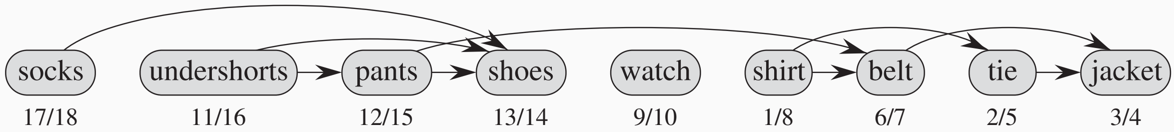
Definición de ordenamiento topológico

Un ordenamiento topológico de un DAG $G = (V, E)$, es un ordenamiento total de los vértices de G, tal que para cada arco $(u, v) \in E$, el vértice u precede al vértice v en el ordenamiento.

Ejemplo de un ordenamiento topológico



(a) Resultado de DFS en el DAG de la Figura 4.



(b) Ordenamiento topológico con el tiempo final de DFS.

Figura 9: Ordenamiento topológico de los vértices. Fuente [1].

Algoritmo de ordenamiento topológico de un DAG

Función Ordenamiento-Topológico($G(V, E)$)

inicio

Ejecute una versión modificada de $DFS(G)$, tal que inicialice una lista enlazada L vacía al comienzo del algoritmo, y que una vez que se obtenga el tiempo final de un vértice v, entonces agregue v en el frente de la lista enlazada L. ;

devolver L ; `/* Lista enlazada de vértices ordenados en orden topológico */`

Análisis del tiempo del ordenamiento topológico

- El tiempo de DFS es $\Theta(|V| + |E|)$.
- Inicializar la lista enlazada es $O(1)$.
- Agregar al frente en la lista enlazada es $O(1)$ y se hace en el momento de computar el tiempo final.
- Agregar todos los vértices a la lista enlazada es $\Theta(|V|)$.
- Por lo tanto, el tiempo del peor caso es $\Theta(|V| + |E|)$.

Lados de un DAG

Lema 1

Un digrafo G es acíclico si solo si la búsqueda en profundidad sobre G no genera ningún *back edge*.

Correctitud del algoritmo de ordenamiento topológico

Teorema 1

El algoritmo `Ordenamiento-Topológico` produce un ordenamiento topológico del digrafo acíclico dado como entrada.

Prueba de la correctitud de `Ordenamiento-Topológico`

Prueba del Teorema 1

- Es suficiente mostrar que para cada arco $(u, v) \in E$ se cumple que $v.tiempoFinal < u.tiempoFinal$
- Sea (u, v) un arco cualquiera explorado por DFS en este instante.
- Se tiene que v es de color GRIS, BLANCO o NEGRO
- v no puede ser de color GRIS, porque v podría ser ancestro de u y esto contradice Lema 1
- Si v es BLANCO, entonces u es ancestro de v, en consecuencia $v.tiempoFinal < u.tiempoFinal$
- Si v es NEGRO, entonces v ya finalizó y existe un $v.tiempoFinal$; se tiene que u va a finalizar después, en consecuencia $v.tiempoFinal < u.tiempoFinal$
- Por lo tanto, es cierto que para cada arco $(u, v) \in E$ después de aplicar DFS se tiene que $v.tiempoFinal < u.tiempoFinal$. $\square$

Referencias

[1] T. Cormen, C. Leirserson, R. Rivest, and C. Stein.
Introduction to Algorithms.
McGraw Hill, 3ra edition, 2009.

[2] C. de Ingeniería de Computación.
Plan de estudios de la carrera de ingeniería de la computación de la usb.
https://www.comp.coord.usb.ve/inicio/pensum-programas-ep-y-grafos/planes-de-estudios-de-ingenier%C3%ADa-en-computaci%C3%B3n#h.p_YU2VpUs5EG3g, November 2021.

[3] O. Meza and M. Ortega.
Grafos y Algoritmos.
Editorial Equinoccio, 2da edition, 2004.