Grafos:

CFCs y Orden Topológico

Integrantes:
Andrés Monetta
Alejandro Clara
Sebastián Daloia

Indice

Ejercicio 3....3
Parte 1:....3
Proposición a:....3
Proposición b:....3
Proposición c-1:....3
Proposición c-2:....4
Parte 2:....4
Ejercicio 4....6
Indicaciones para que Topo encuentre colecciones....6
Algoritmo separación de artículos....6

Ejercicio 3

Parte 1:

Proposición a:

Una recorrida en G que empieza en un vértice de una CFC sumidero, visita todos los vértices de esa CFC y ningún otro.

Demostración:
Que la recorrida visita todos los vértices de la CFC sumidero es cierto por definición de CFC (existe un camino entre todo par de vértices).
Luego no visita ningún otro vértice de V por definición de CFC sumidero (las únicas aristas $(u,v) \in A$ *con* $u \in C$ *son tales que* $v \in C$).

Proposición b:

Una CFC fuente en G es sumidero en G^T.

Demostración:
Observación previa: Una componente CFC C en G con vértices $v_1, \dots, v_n$ es tal que existe un camino entre todo par de vértices.
Para estos vértices en G^T va a seguir existiendo un camino para todo par de vértices, pues los caminos de G en G^T se recorren en sentido inverso.
Con lo que G y G^T tienen los mismos vértices para cada CFC en cada uno de ellos.

Luego observando el grafo transpuesto del grafo de componentes G_C^T es tal que $(u_i, u_j) \in A_C^T$ *si* $(u_j, u_i) \in A_C$
Por lo que si una CFC era fuente en G_C entonces es sumidero en G_C^T .

Proposición c-1:

El grafo G_C se puede ordenar topológicamente como una secuencia $o = \{o_1, o_2, \dots, o_k\}$

Demostración:
Observación: Sabemos que existe orden topológico en G ↔ G es acíclico.

Supongamos que G_C tuviera un ciclo, digamos entre

$$v_i, v_j \rightarrow \exists (u_i, u_j) \text{ con } u_i \in CFC(v_i) \text{ y } u_j \in CFC(v_j)$$
$$\text{y } \exists (w_i, w_j) \text{ con } w_i \in CFC(v_i) \text{ y } w_j \in CFC(v_j)$$
$$\rightarrow CFC(u_i) \cup CFC(v_j) \text{ es CFC}$$

absurdo pues existen k componentes CFC y esto concluye que existe k-1.
Entonces G es acíclico, entonces existe al menos un orden topológico en G.

Proposición c-2:

$CFC(o_1)$ es sumidero en G^T

Demostración:
o_1 es tal que solo existen aristas $(o_1, o_j) j=2,\dots,k$
Luego $CFC(o_1)$ es una fuente para G entonces usando la proposición b
$CFC(o_1)$ es sumidero en G^T .

Proposición c-3:
Sea G' el grafo que se obtiene al excluir de G la componente $CFC(o_1)$ entonces
$o'=\{o_1, \dots, o_k\}$ es un ordenamiento topológico de G'_C .

Demostración:
$o=\{o_1, \dots, o_k\}$ es ordenamiento topológico $\rightarrow \nexists$ *aristas* $(o_j, o_i) . \because j>i \, j, i=1,\dots,k$
Luego $o'=\{o_2, \dots, o_k\}$ es ordenamiento topológico pues $o'=o \setminus \{o_1\}$ es decir solo estan siendo removidas aristas del tipo $\{o_1, o_j\}$.

Parte 2:

Dada una secuencia u1, ..., uk de vértices de G, con ui ∈ CFC(oi), 1 ≤ i ≤ k, siendo o = (o1, ..., ok)
un ordenamiento topológico de G_C , describa y justifique un algoritmo que permita encontrar cada
una de las CFCs de G.

```
encontrar_CFC(G: Grafo)
    Dado G expreso su grafo traspuesto G^T
    Para cada u de {u_1,...,u_k}  /* con  u_i ∈ CFC(o_i)  */
        o[j]: DFS(u) en G^T /*Devuelve secuencia que visita todos los
                              vértices de CFC(o_i)*/
        remover vértices marcados en G
    retornar o
```

El algoritmo encuentra las CFC de G debido a que el primer elemento $u1$ pertenece a la primera componente del ordenamiento topológico de G_C este primer ordenamiento es fuente en G y sumidero en G^T (proposición b y proposición c-2), entonces en el grafo transpuesto puedo recorrer todos los

vértices de esa componente y solo de esa (proposición a). Luego removiendo los vértices del ordenamiento primero, repito el mismo procedimiento para el nuevo ordenamiento $o'=\{o_{2,}\ldots,o_k\}$ (proposición c-3).

Ejercicio 4

Indicaciones para que Topo encuentre colecciones

Juanes y Topo Lógico saben que para los artículos de una colección, las referencias que estos tienen permiten comenzar a leer uno de los artículos y encontrar todos los artículos que pertenecen a esa colección siguiendo las referencias. Además existen referencias de articulos de una colección a otra anterior en el tiempo.

Si consideramos los artículos como vértices de un grafo, y a las referencias como las aristas del mismo, y que existe un camino de referencias entre cada par de artículos de una colección entonces podemos afirmar que éstas son componentes fuertemente conexas.

Topo Logico ya cuenta con un diagrama (grafo de componentes) con las colecciones de artículos y las referencias entre ellas.

Debería crear un orden topológico para este diagrama(ya que existe al menos uno), eligiendo en primer lugar las fuentes que haya.
Luego construir el grafo transpuesto del diagrama y recorrer la colección que aparezca en primer lugar en el orden (ahora sumidero en el grafo transpueto), Topo se asegura con esto recorrer todos los artículos de esa colección sin pasar por ninguna referencia a otra colección.
Una vez logre enumerar los artículos de esta colección, la retira del diagrama y realiza el mismo procedimiento tomando ahora como fuente la siguiente colección existente en el orden topológico. Procediendo de la misma forma que arriba podrá rearmar todas las colecciones nuevamente.
Si hubiese más de una fuente en el diagrama, procedería de la misma manera.

Este grafo tiene un orden topológico

Algoritmo separación de artículos

```
PROCEDIMIENTO BFS (v: Vértice, lista_distancias :Lista)

VAR   Q:Cola de vértice
      u,w: vértice

Comienzo

     CrearCola(Q)
     Marcar(v)
```

```
     setNivel(v, 0)
     InsBack(Q,v) /*”encolar”*/


     Mientras No-Vacia(Q)

          u = primero(Q)

          <Procesar(u)>

          Q = resto(Q)
          Para cada w adyacente a u
               Si w no marcado entonces
                    Marcar(w)
                    setNivel(w,getNivel(u) + 1)
                    agregarALista(lista_distancias,w,getNivel(u) + 1)
                    InsBack(Q,w)

               Fin si
          Fin para
     Fin Mientras
Fin
```