GRAFI: COMPONENTI FORTEMENTE CONNESSE

[Deme, seconda edizione] cap. 12

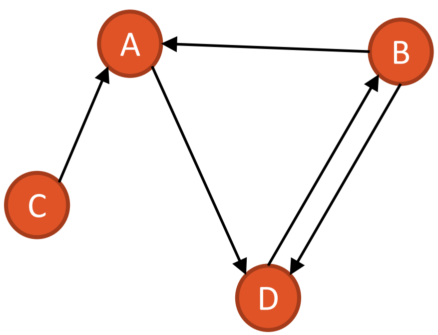
Sezione 12.5

[Cormen] Sezioni 22.4 e 22.5

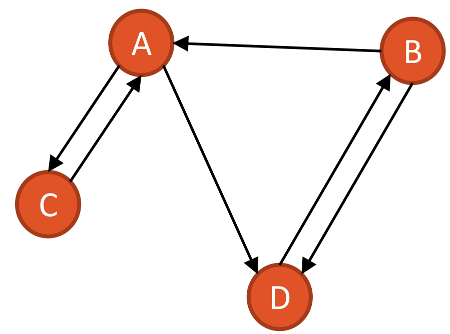
(dall’introduzione ai grafi) Grafo fortemente connesso

Se in un grafo G (**orientato**) esiste un cammino da ogni vertice verso ogni altro vertice, si dice che G è **fortemente connesso**.

Questo grafo orientato <u>non è</u> connesso (C non è raggiungibile da nessun nodo)



Questo grafo orientato è connesso



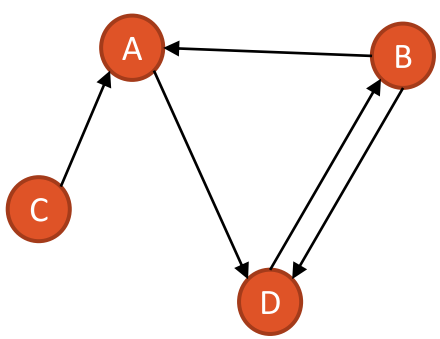
NOTA: non ci devono essere «tutti» gli archi perché il grafo sia fortemente connesso

Connessione forte (di vertici)

In un grafo **orientato** G:

due nodi **u**, **v** si dicono (fra loro) **mutuamente raggiungibili,** o **fortemente connessi,** se ognuno dei due è raggiungibile dall'altro, cioè se esiste un cammino da u a v e un cammino da v a u, cioè se **appartengono ad uno stesso ciclo.**

Indichiamo la notazione di connettività forte con a notazione u $\leftrightarrow$ v



A $\leftrightarrow$ B

Grafo fortemente connesso

(Dall’introduzione)

[…] Se in un grafo G (**orientato**) esiste un cammino da ogni vertice verso ogni altro vertice, si dice che G è **fortemente connesso**.

Riscriviamo la definizione

Un grafo orientate si dice fortemente connesso se tutti i suoi nodi sono fra loro **fortemente connessi**.

La connessione forte è una relazione di equivalenza

La relazione di connessione forte, che indichiamo con il simbolo infisso $\leftrightarrow$, è una relazione di equivalenza, perché è:

riflessiva: $u \leftrightarrow u$ (u raggiungibile da u con cammino nullo);

simmetrica: per definizione, se $u \leftrightarrow v$ allora $v \leftrightarrow u$.

transitiva: se $u \leftrightarrow v$ e $v \leftrightarrow w$ allora ovviamente $u \leftrightarrow w$.

Componenti fortemente connesse

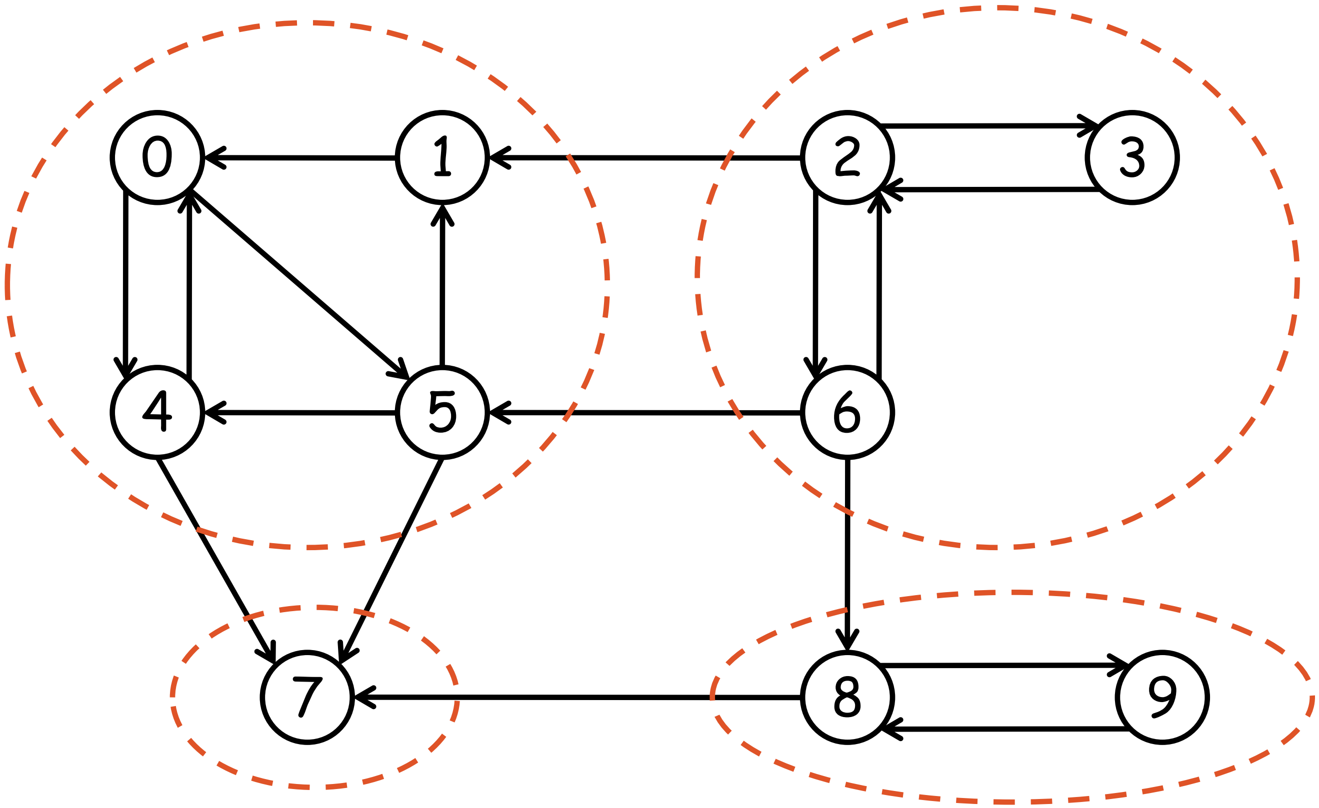
Le **componenti fortemente connesse (cfc)** di un grafo sono le **classi di equivalenza** della relazione di connessione forte, cioè:

Una **cfc** di un grafo **orientato** G è un sottografo G’ di G **fortemente connesso** e **massimale**

cioè un sottografo G’ di G tale che:

- i nodi di G' sono tutti fra loro **fortemente connessi** (cioè mutuamente raggiungibili);
- **nessun altro** nodo di G è fortemente connesso con nodi di G'.

Quali sono le cfc di questo grafo?



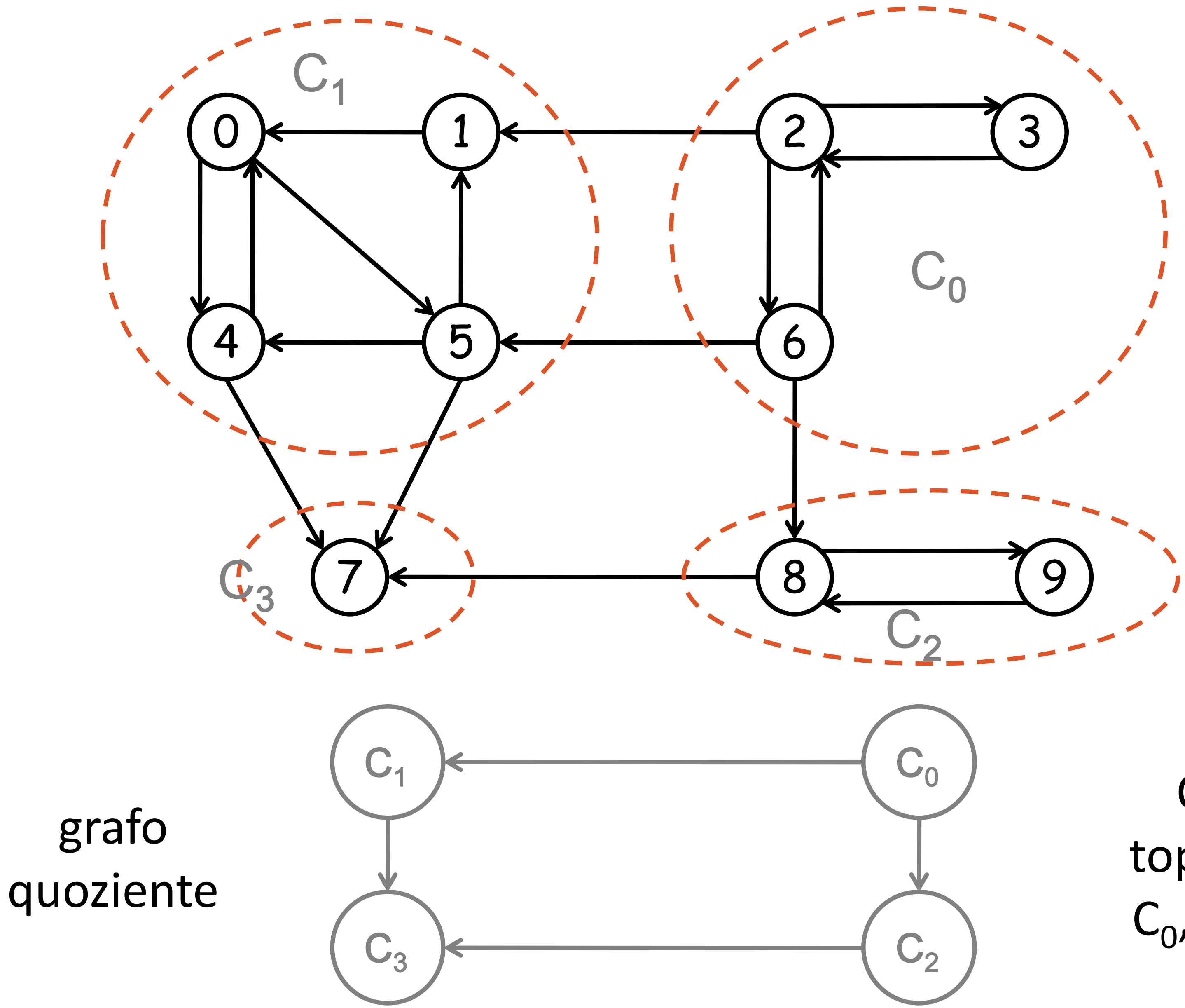
Una proprietà delle cfc

La relazione di connettività forte è una relazione di **equivalenza**, e che le cfc sono le sue **classi di equivalenza**.

Possiamo dire qualcosa sull'**insieme quoziente** di queste classi di equivalenza?

Sì, l'insieme quoziente è a sua volta **un grafo**. È **diretto** ed è facilmente dimostrabile che è **aciclico** (altrimenti, le cfc del ciclo si potrebbero «fondere» e non sarebbero massimali).

Quindi, l'insieme quoziente è un **DAG** ed è possibile calcolare su di esso un **ordine topologico**.



Ordine topologico: C_0, C_1,C_2,C_3

Come calcolare una componente fortemente connessa - idea

(Dalla definizione) la cfc di x conterrà **tutti** e **soli** i vertici **raggiungibili da x** **e** dai quali **x è raggiungibile**.

Allora calcoliamo i vertici raggiungibili da x, poi quelli da cui x è raggiungibile, e facciamo **l'intersezione dei due insiemi**

Calcolare una componente fortemente connessa (algoritmo semplice)

Per trovare la cfc contenente il vertice x:

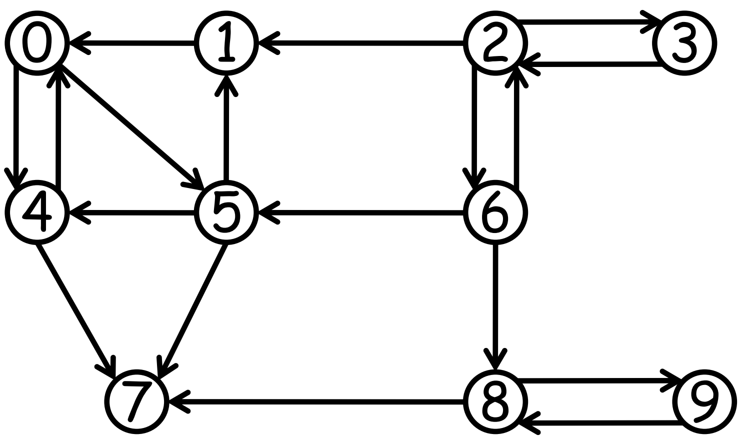
1. Calcoliamo i **discendenti** di x D(x), i vertici di G raggiungibili da x
2. Calcoliamo gli **antenati** di x A(x), i vertici di G che raggiungono x
3. cfc[x] ovvero la componente fortemente connessa contenente x, è data da **D(x) ∩ A(x)**

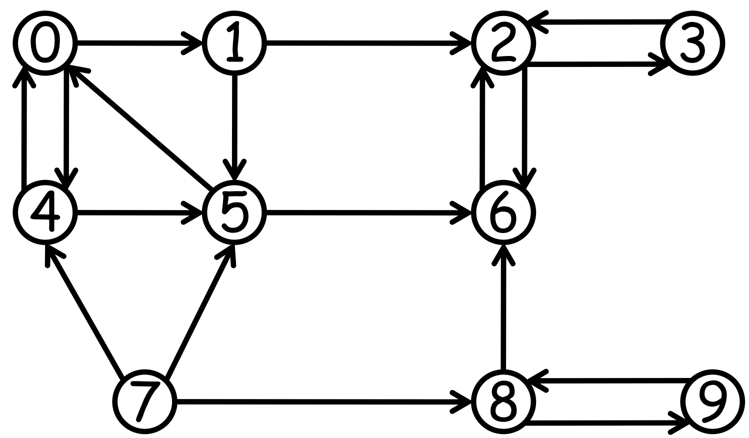
Il fatto che tutti i vertici di cfc[x] siano fortemente connessi tra di loro è banalmente verificabile considerando il fatto che tutti raggiungono x e tutti sono raggiungibili da x, quindi preso un u ed un v generici appartenenti a cfc[x], u raggiunge v con il cammino u → x → v e v raggiunge u con il cammino v → x → u.

Come calcolo D(x) e A(x)

Banalmente, D(x) si ottiene con una **visita** a partire da x.

A(x) si ottiene con una visita a partire da x del **grafo trasposto** di G, G^T. Ovvero il grafo in cui **tutti gli archi sono invertiti**.





D(1) = {0,1,4,5,7}

A(1) = {0,1,2,3,4,5,6}

Calcolo di tutte le cfc

Algoritmo:

for ogni vertice v di G non ancora marcato

calcola cfc[v]

COSTO:

Calcolo singola cfc:

1. Visita G $O(n+m)$
2. Calcola G^T $O(n+m)$
3. Visita G^T $O(n+m)$

TOT: **$O(n+m)$**

Per n vertici, quindi **$O(n^2+nm)$**

Di seguito vedremo che si può fare di meglio

Lemma del cammino fortemente connesso

Se due vertici x, y di un grafo sono in **una stessa cfc**, allora **nessun cammino tra di essi può abbandonare tale cfc**.

DIMOSTRAZIONE. x e y appartengano alla stessa cfc. Sia z tale che x → z e z → y, dimostriamo che $\mathbf{z \in cfc[x]}$

Per dimostrare che z appartiene alla stessa cfc di x (e quindi di y), ci basta dimostrare che **z è raggiungibile da x e viceversa**

z è banalmente raggiungibile da x per ipotesi **x → z**.

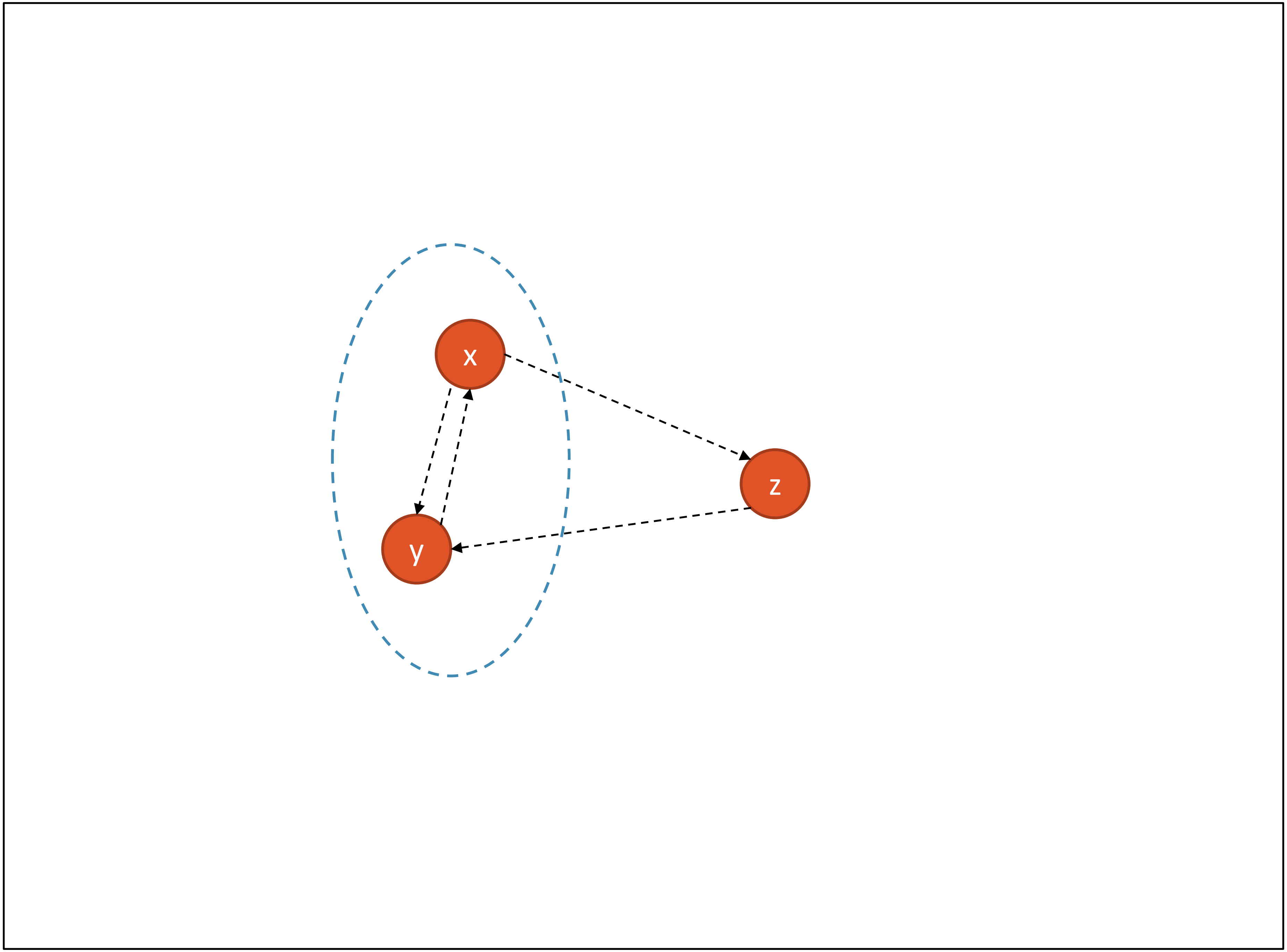
Siccome x e y appartengono alla stessa cfc, esisterà un cammino **y → x**

Esiste anche **z → y** per ipotesi.

Quindi, per concatenazione, esisterà un cammino **z → y → x**

Quindi esisterà un cammino **z → x**

CVD.



x
y
z

Teorema del sottoalbero fortemente connesso

TEOREMA. In una **qualunque DFS** di un grafo G orientato **tutti i vertici di una cfc vengono collocati in uno stesso sottoalbero**.

DIMOSTRAZIONE.

Sia r il primo vertice di una data cfc che viene scoperto nella DFS.

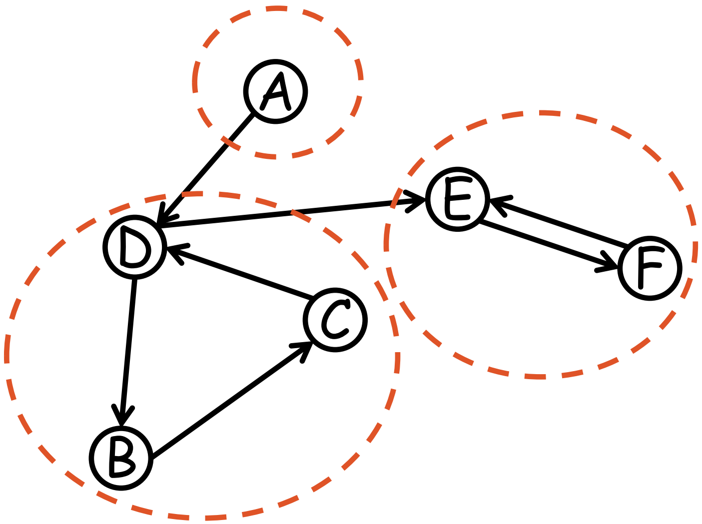
Da r sono raggiungibili tutti gli altri vertici della cfc (per definizione di cfc)

Poiché r è il primo, al momento della scoperta di r tutti gli altri vertici della cfc saranno bianchi.

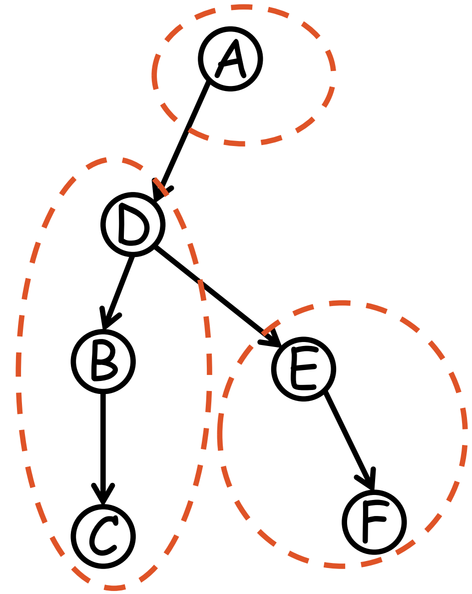
Inoltre, tutti i cammini da r agli altri vertici della cfc conterranno solo vertici bianchi che fanno parte della cfc (perché, per il lemma precedente, dentro a tali cammini non lasciano mai la cfc).

Allora (per il Teorema del cammino bianco) **tutti i vertici appartenenti alla cfc di r saranno discendenti di r nell'albero DFS**. CVD.

grafo G



albero di visita DFS



Le cfc del grafo sono tre:
{A}, {D, B, C}, {E, F}

Attenzione però: non vale l'inverso del teorema precedente, quindi nello stesso sottoalbero ci possono essere vertici di più componenti fortemente connesse

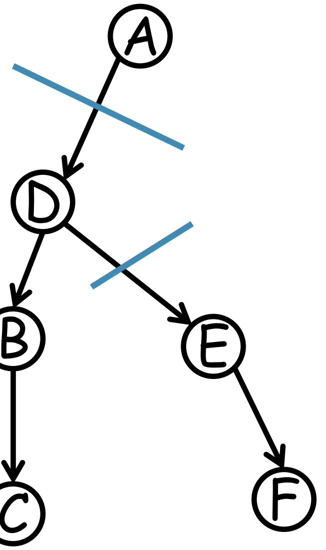
Idea

Ma allora, potremmo farci **guidare da una visita DFS per trovare le cfc**.

In particolare potremmo:

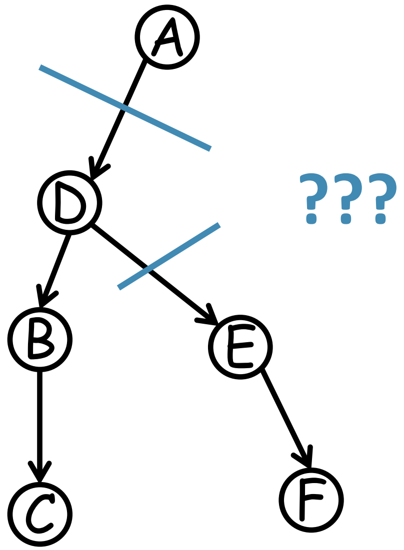
1. Trovare un albero DFS del grafo, contenente tutte le cfc
2. Tagliare l'albero in punti opportuni, partizionandolo nelle sue cfc

Si noti che partizionare l'albero nelle cfc è sempre possibile perché se in un albero della foresta u è discendente di v e u non appartiene a cfc[v], allora ogni discendente t di u, non può appartenere a cfc[v].



Problema: trovare i punti per tagliare l'albero

Ma noi non sappiamo dove tagliare l'albero!

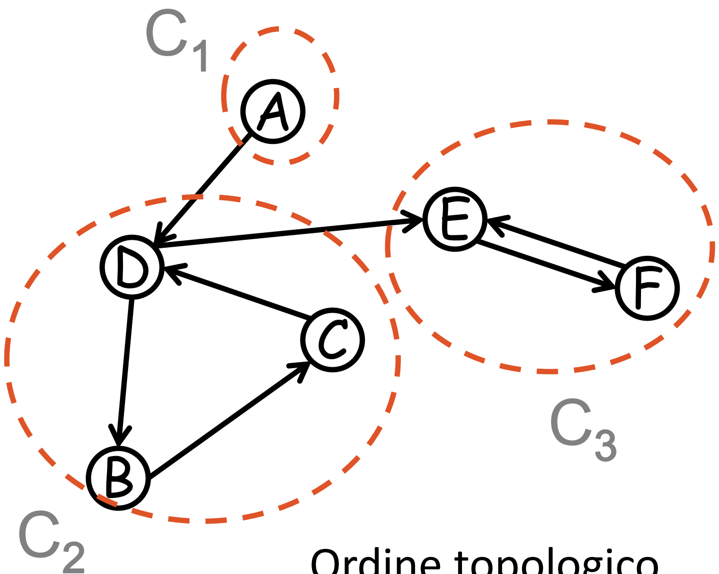


Per fortuna, ci aiutano 2 proprietà importanti.

Proprietà 1

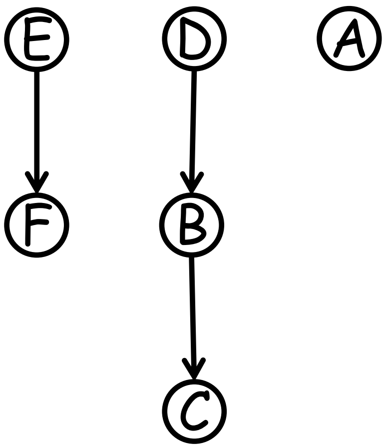
Esiste sempre, per ogni grafo diretto, almeno un **ordine di visita DFS** dei suoi nodi tale per cui **le cfc sono già separate** nella foresta di visita.

(è legato all'inverso di un ordine topologico del suo grafo quoziente, che noi però non abbiamo!)



Ordine topologico del grafo quoziente: C_1, C_2, C_3

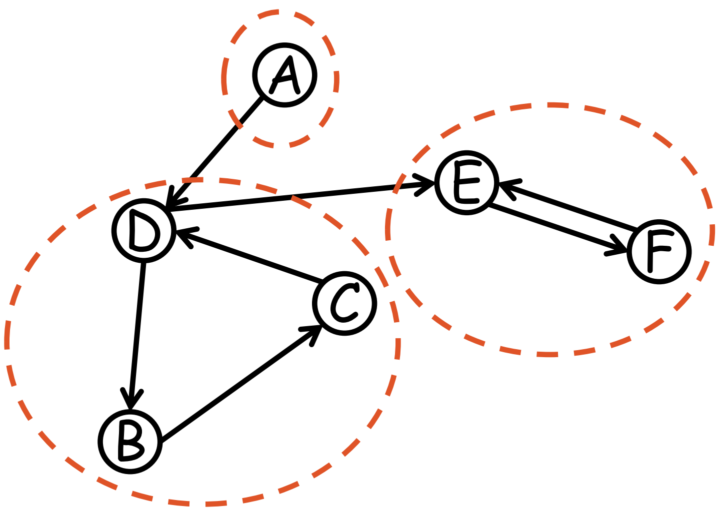
Visitandolo nell'ordine E F D B C A otteniamo



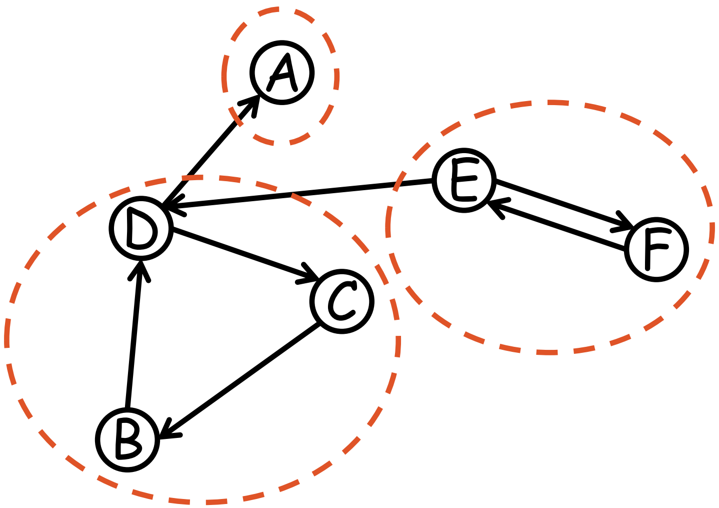
Proprietà 2

Un grafo G ed il suo trasposto G^T hanno **le stesse cfc**.

grafo G



grafo G^T



IDEA

G è un grafo diretto, quindi **la Proprietà 1 vale anche per G^T**.

Siccome G e G^T hanno le stesse cfc (Proprietà 2), cerchiamo di capire se possiamo sfruttare una DFS su G per trovare un **ordine di visita DFS di G^T** tale che la foresta DFS di G^T abbia le cfc tra di loro separate (Proprietà 1).

Se così fosse, con **2 visite DFS** avremmo risolto il problema.

IDEA (continua): Usare la prima visita per determinare un ordine di visita per la seconda

Dati due vertici x e y, quale visitare per primo nel grafo trasposto in maniera che x e y non stiano nello stesso sottoalbero?

Assumiamo che **x non sia nella stessa cfc di y**. Dopo la DFS sul grafo G si possono presentare i seguenti due casi:

1. **y è discendente di x** in un albero della foresta DFS di G
2. **x e y non sono uno discendente dell'altro** nella foresta DFS di G (sono in alberi o sottoalberi distinti della foresta)

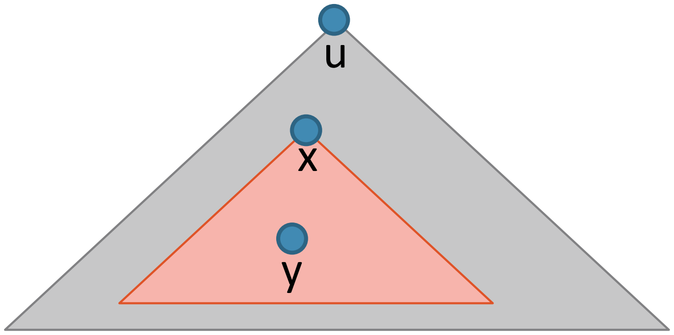
Capiamo come queste informazioni possono essere usate su G^T

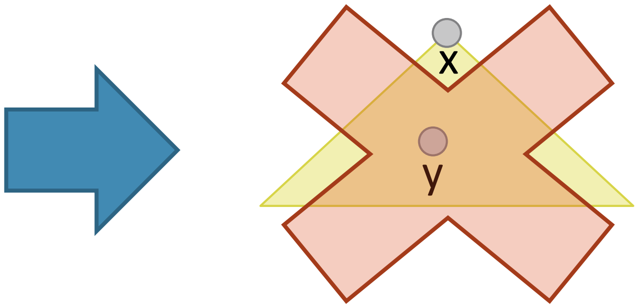
Caso 1

Dati due vertici x e y (che **non** appartengono alla stessa cfc), quale visitare per primo nel grafo trasposto?

y è discendente di x in un albero della foresta DFS di G

y non può essere discendente di x nella foresta DFS di G^T





Esiste in G un cammino da x a y.
Ma, **non esisterà il cammino da y a x in G**
(altrimenti x e y sarebbero nella stessa cfc).

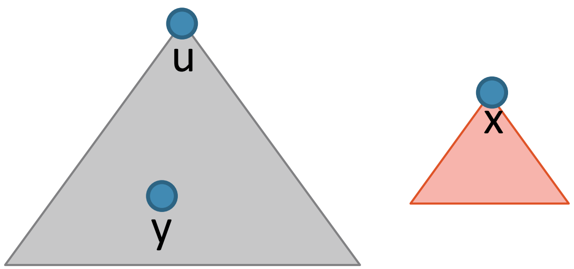
Non esisterà il cammino da x a y in G^T.
(e, se visito prima x, neanche quello da y a x nella visita)

Quindi, se x precede y nella visita DFS di G^T, sarò sicuro che x e y non staranno nello stesso albero

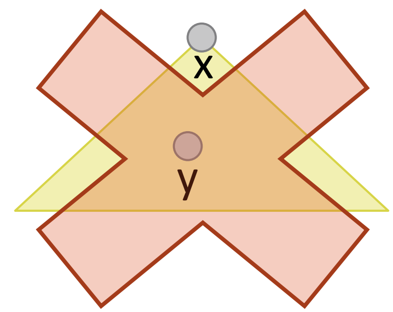
Caso 2

Dati due vertici x e y (che **non** appartengono alla stessa cfc), quale visitare per primo nel grafo trasposto?

In G x e y non sono uno discendente dell'altro



y non può essere discendente di x nella foresta DFS di G^T



può esistere un cammino da x a y (a causa di archi di attraversamento) in G, ma **non può esistere nessun cammino da y a x** (altrimenti x sarebbe nel sottoalbero di y)

Non esisterà il cammino da x a y in G^T.
(e, nella visita, neanche quello da y a x, se x viene visitato prima)

Quindi, se x precede y nella visita DFS di G^T, sarò sicuro che x e y non staranno nello stesso albero

Idea

In entrambi i casi allora sarà conveniente **che la visita che inizia da x G^T preceda quella che parte da y** perché, se i vertici non sono nella stessa cfc, non verranno collocati nello stesso albero.

Sembra allora che i vertici nella visita di G^T debbano essere considerati:

- **dall'alto verso il basso**, e
- **da destra verso sinistra**

Idea- II

Per capire come rispettare l'ordine della slide precedente, osserviamo gli intervalli di attivazione dei due vertici nei 2 casi

nel caso 1 si ha:

d[x] **d[y]** **f[y]** **f[x]**

nel caso 2 si ha:

d[y] **f[y]** **d[x]** **f[x]**

In entrambi i casi f[x] > f[y] (ricorda: sono i tempi della **prima** visita)

QUINDI **(nella seconda visita)** i vertici vanno considerati **in ordine decrescente di tempo di fine visita**

Un algoritmo per il calcolo delle cfc Algoritmo di Kosaraju

Dalle osservazioni precedenti ricaviamo il seguente algoritmo

Strongly-Connected-Components (G)

1. **Visita G** con l'algoritmo ***VISITA_TUTTI_I_VERTICI-DFS*** e costruisci una lista dei vertici in ordine decrescente dei tempi di fine visita
2. **Costruisci G^T**
3. **Visita G^T** con l'algoritmo ***VISITA_TUTTI_I_VERTICI-DFS*** **considerando**, nel ciclo principale dell'algoritmo, **i vertici nell'ordine trovato al passo 1**.

Costo: O(n+m) + O(n+m) + O(n+m) = **O(n+m)**

È il meglio che si possa fare? No, il libro suggerisce un algoritmo che effettua una sola visita in profondità (però con complessità sempre **O(n+m)**). Provate a leggerlo e capirlo.

Correttezza dell'algoritmo

Per dimostrare la correttezza dell'algoritmo ci servono le seguenti proprietà:

Teorema del sottoalbero fortemente connesso: In una qualunque DFS di un grafo G orientato **tutti i vertici di una cfc vengono collocati in uno stesso sottoalbero**. [già dimostrato]

Lemma 2. **Un grafo orientato e il suo trasposto hanno le stesse cfc**. [dimostrazione immediata dalla definizione di cfc]

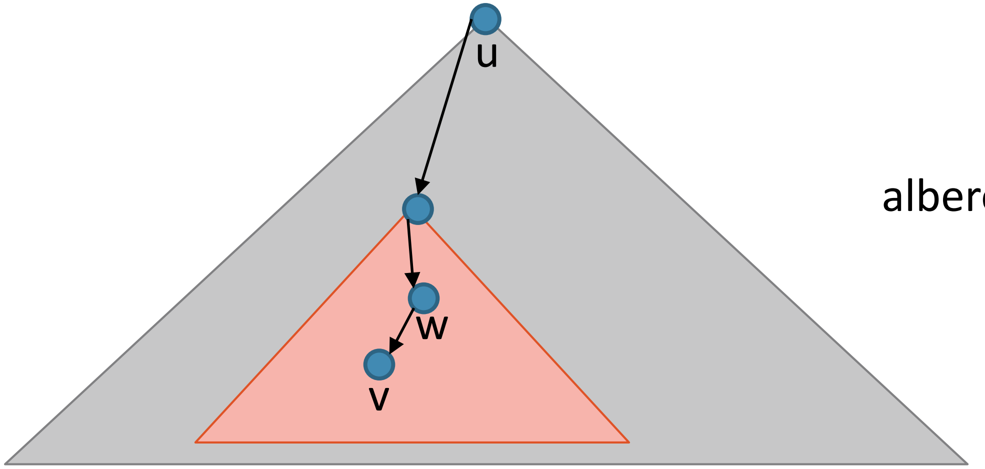
Lemma 3. Sia A^T un albero ottenuto con la visita in profondità di G^T, considerando i vertici in ordine decrescente dei tempi di fine visita su G, e sia u la sua radice. **Per ogni vertice v discendente di u in A^T, v e u appartengono alla stessa cfc**.

Dimostrazione lemma 3

Dimostriamo prima di tutto che **ogni discendente di u in A^T** è anche un **discendente di u** in un albero della foresta costruita dalla **visita in profondità su G**.

La dimostrazione è fatta per **assurdo**.

Consideriamo un cammino sull'albero A^T a partire dalla radice u; sia **v il primo vertice sul cammino per cui il lemma non vale** (cioè v non è discendente di u sulla visita di G) e sia w il suo predecessore sul cammino.



albero A^T di visita DFS di G^T

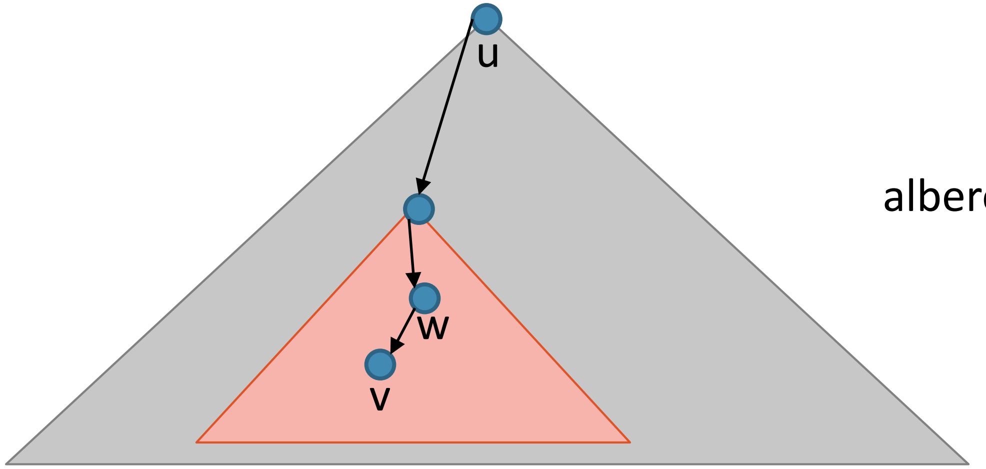
Dimostrazione lemma 3 – II

Poiché v è il primo vertice per cui il lemma non vale, **l'enunciato vale per w**, e quindi l'attivazione di w è dentro l'attivazione di u nella visita di G:

d[u] ≤ d[w] < f[w] ≤ f[u] (w può anche essere u)

Siccome la visita DFS di G^T considera i vertici in ordine **decrescente** di fine visita (su G), per ogni discendente di u in A^T, e quindi in particolare per v, vale

f[v] < f[u] nella visita di G



albero A^T di visita DFS di G^T

Dimostrazione lemma 3 – III

Per il teorema delle parentesi, **se v non è un discendente di u** nella prima visita, **i due intervalli di visita di v e u devono essere disgiunti**, cioè deve valere (nella visita di G):

d[v] < f[v] < d[u] < f[u]

d[v] f[v] d[u] d[w] f[w] f[u]

Ma questo è **impossibile**

v è adiacente a w in G^T quindi **w è adiacente a v in G**, e la visita di v non può terminare prima che sia iniziata la visita di un suo adiacente, cioè **f[v] non può precedere d[w]**.

Dimostrazione lemma 3 – IV

Terminata la dimostrazione per assurdo, sappiamo che **v è un discendente di u nella visita DFS di G**

Quindi in G **esiste un cammino da u a v.**

Siccome v è discendente di u in A^T, **esiste anche un cammino** da u a v in G^T, e quindi **da v a u in G**.

Quindi **v appartiene alla stessa cfc di u**.

CVD.

Dimostrazione correttezza dell'algoritmo Strongly-Connected-Components

I lemmi permettono di dimostrare che ogni albero della foresta ottenuta con la visita in profondità di G^T contiene:

- **tutti i vertici di una cfc di G** (Teorema del sottoalbero fortemente connesso e Lemma 2)
- **solo i vertici di una cfc di G** (Lemma 3)

⬇

L'algoritmo Strongly-Connected-Components è

corretto

ossia, quando applicato ad un grafo orientato, restituisce una foresta i cui alberi individuano le componenti fortemente connesse del grafo.

Cosa devo aver capito fino ad ora

- Connessione forte di vertici
- Componenti fortemente connesse
- Calcolare la cfc a cui appartiene un singolo vertice
  - Algoritmo
  - Costo
- Calcolare tutte le cfc di un grafo diretto. Algoritmo semplice e costo
- Lemma del cammino fortemente connesso
- Teorema del sottoalbero fortemente connesso
- Potatura dell’albero DFS

Cosa devo aver capito fino ad ora - II

- Trovare un ordine di visita del grafo trasposto per avere la foresta delle cfc
- Algoritmo di Kosaraju per il calcolo delle cfc
  - Costo
  - Correttezza
    - (Lemmi 2 e 3 con dimostrazione)
- Il grafo quoziente delle cfc è un DAG

…se non ho capito qualcosa

- Alzo la mano e chiedo
- Ripasso sul libro
- Chiedo aiuto sul forum
- Chiedo o mando una mail al docente