APA Modulo 1 Lezione 8

Elena Zucca

25 marzo 2020

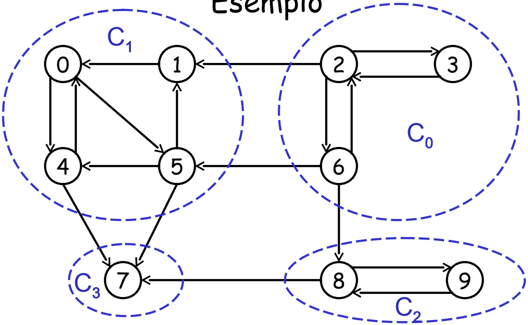
Componenti fortemente connesse

Definizione

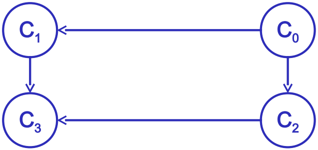
In un grafo orientato G, due nodi u e v si dicono mutuamente raggiungibili, o fortemente connessi, se ognuno dei due è raggiungibile dall'altro, ossia se esistono un cammino da u a v e un cammino da v a u (u e v appartengono allo stesso ciclo).

- la relazione di mutua raggiungibilità o connessione forte, che indichiamo con $\leftrightarrow$, è una relazione di equivalenza, ossia:
  - riflessiva $x \leftrightarrow x$
  - simmetrica $x \leftrightarrow y$ implica $y \leftrightarrow x$
  - transitiva $x \leftrightarrow y$ e $y \leftrightarrow z$ implica $x \leftrightarrow z$
- le componenti fortemente connesse sono le classi di equivalenza della relazione $\leftrightarrow$, ossia i sottografi massimali di G i cui nodi sono tutti fortemente connessi tra loro

Esempio



grafo quoziente



Grafo quoziente

- nodi = componenti fortemente connesse
- esiste un arco $(\mathcal{C}, \mathcal{C}')$ se e solo se esiste un arco da un nodo in $\mathcal{C}$ a un nodo in $\mathcal{C}'$
- chiaramente è aciclico quindi su di esso esiste un ordine topologico

Verso un algoritmo per trovare le c.f.c.

Definizione

Una componente fortemente connessa di un grafo orientato è una sorgente o un pozzo se è, rispettivamente, un nodo sorgente o pozzo nel grafo quoziente.

Verso un algoritmo per trovare le c.f.c.

In una visita in profondità di un grafo G, il tempo di fine visita $end(\mathcal{C})$ di una componente fortemente connessa $\mathcal{C}$ è il massimo dei tempi di fine visita dei nodi appartenenti a $\mathcal{C}$.

Proprietà della visita

Se $\mathcal{C}$ e $\mathcal{C}'$ sono due componenti fortemente connesse di G, ed esiste un arco da (un nodo di) $\mathcal{C}$ a (un nodo di) $\mathcal{C}'$, allora in qualunque visita in profondità di G si ha:

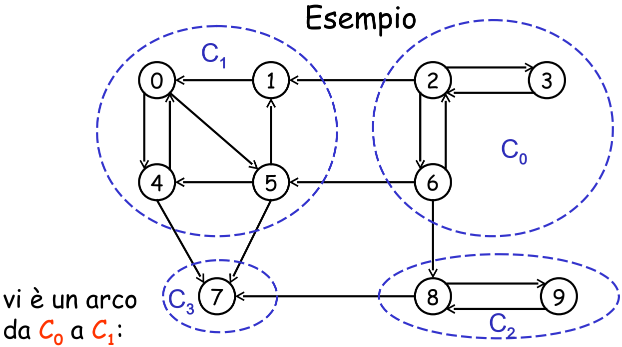
$$end(\mathcal{C}') < end(\mathcal{C})$$

Prova (semi-formale)

Due casi:

- Il primo nodo visitato di $\mathcal{C}$ e $\mathcal{C}'$ è in $\mathcal{C}$, sia u.
  Allora tutti i nodi di $\mathcal{C}$ e $\mathcal{C}'$ sono visitati prima di terminare la visita di u:
  $$end(\mathcal{C}') < end(u) = end(\mathcal{C})$$
- Il primo nodo visitato di $\mathcal{C}$ e $\mathcal{C}'$ è in $\mathcal{C}'$. Non può esistere un cammino da $\mathcal{C}'$ a $\mathcal{C}$, quindi tutti i nodi di $\mathcal{C}'$ sono visitati prima di qualunque nodo di $\mathcal{C}$:
  $$end(\mathcal{C}') < end(\mathcal{C})$$

Graficamente



primo nodo visitato:

- in C_0: la chiamata ricorsiva termina solo dopo la fine delle chiamate ricorsive su tutti gli altri nodi di C_0 e su tutti i nodi di C_1
- in C_1: la chiamata ricorsiva termina prima di qualunque chiamata ricorsiva su nodi di C_0

Verso un algoritmo per trovare le c.f.c.

Useremo Teorema 1 + Teorema 2 + Osservazione

Teorema 1

In una visita in profondità di un grafo orientato il nodo avente il massimo tempo di fine visita appartiene a una componente fortemente connessa sorgente.

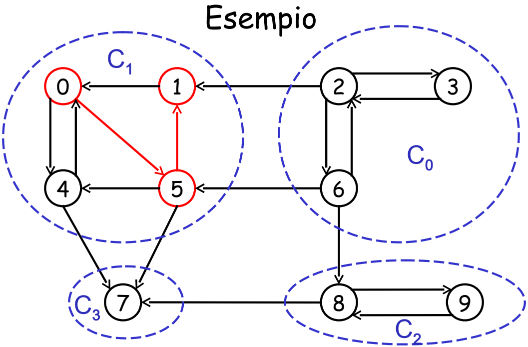
Dimostrazione.

Sia u il nodo avente il massimo tempo di fine visita, appartenente a $\mathcal{C}$. Se $\mathcal{C}$ non fosse sorgente, ci sarebbe un arco da un altra c.f.c. $\mathcal{C}'$ a $\mathcal{C}$. Per il lemma precedente, avremmo $end(\mathcal{C}) < end(\mathcal{C}')$, quindi $end(u)$ non sarebbe il massimo tempo di fine visita. $\square$

Osservazione

- non vale la proprietà simmetrica: il nodo avente il minimo tempo di fine visita non appartiene necessariamente a una componente fortemente connessa pozzo
- non è quindi possibile con una visita in profondità di un grafo trovare le componenti fortemente connesse pozzo

Controesempio



se la visita inizia dal nodo 0, e poi fra gli adiacenti di 0 visita per primo il nodo 5, e fra gli adiacenti di 5 visita per primo il nodo 1, allora, poiché 1 non ha nodi adiacenti non visitati, la visita di 1 termina subito

il nodo col minimo end è quindi 1, che si trova nel cfc C_1, che non è un pozzo

11

Verso un algoritmo per trovare le c.f.c.

Teorema 2

In una visita in profondità di un grafo orientato la visita di un nodo appartenente a una componente fortemente connessa pozzo $\mathcal{C}$ visita tutti e soli i nodi di $\mathcal{C}$.

Dimostrazione.

Ovvio, perché tutti i nodi raggiungibili vengono visitati e non c'è nessun arco uscente da una componente fortemente connessa pozzo. $\square$

Verso un algoritmo per trovare le c.f.c.

Osservazione

Una componente fortemente connessa sorgente in un grafo orientato G è una componente fortemente connessa pozzo nel grafo trasposto G^T, cioè nel grafo che si ottiene da G invertendo l'orientamento degli archi.

Mettendo insieme le cose

Teorema 1 ci fornisce un modo per trovare via via le componenti fortemente connesse sorgente, quindi in un ordine topologico

Teorema 2 ci fornisce un modo per trovare tutti i nodi che stanno in una componente fortemente connessa, ma ci serve che sia pozzo

Osservazione ci permette di vedere le componenti sorgente via via trovate come pozzi, quindi di usare Teorema 2

Algoritmo a parole

- si effettua una visita in profondità inserendo i nodi in una sequenza `Ord` in ordine di fine visita
- si estrae via via da `Ord` l'ultimo nodo u, per il Teorema 1 u si trova in una c.f.c. sorgente nel grafo ottenuto da G non considerando le c.f.c. precedenti
- per trovare tutti i nodi di tale c.f.c., che per l'Osservazione è una c.f.c. pozzo nel grafo trasposto, per il Teorema 2 basta effettuare una visita dei nodi (non ancora visitati) raggiungibili da u nel grafo trasposto G^{T}
- non è necessario durante le visite modificare i grafi G e G^{T} perché basta, come al solito, marcare i nodi visitati

Pseudocodice

```
SCC(G)
  DFS(G,Ord) //aggiunge i nodo visitati a Ord in ordine
  //non occorre calcolare i tempi di fine visita
  G^T= grafo trasposto di G
  Ord↔= sequenza vuota//ordine topologico delle c.f.c.
  while (Ord non vuota)
    u= ultimo nodo non visitato in Ord//in sorgente
    𝒞= insieme di nodi vuoto
    DFS(G^T,u,𝒞)//aggiunge nodi visitati in 𝒞
    Ord↔.add(𝒞)//aggiunge in fondo
  return Ord↔
```

Osservazioni

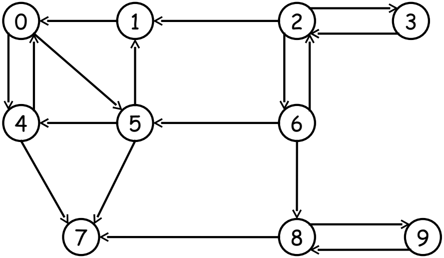
- la sequenza delle componenti fortemente connesse ottenute è in ordine topologico, perché ogni volta si individua una componente fortemente connessa a cui arriva un arco da una delle precedenti
- se il grafo è aciclico le componenti fortemente connesse sono i singoli nodi, quindi l'algoritmo restituisce semplicemente un ordine topologico dei nodi

Complessità

- visita in profondità del grafo: $O(n+m)$
- generazione del grafo trasposto: $O(n+m)$
- successive visite del grafo trasposto: $O(n+m)$

quindi complessivamente $O(n+m)$

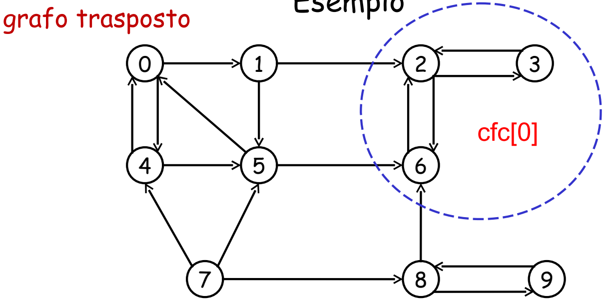
Esempio



- visita in profondità a partire dal nodo 0: la sequenza delle chiamate ricorsive e dei rispettivi ritorni può essere
  (0 (5 (1) (7) (4))) (2 (3) (6 (8 (9))))

la corrispondente sequenza S dei nodi in ordine <u>crescente</u> di fine visita è: 1 7 4 5 0 3 9 8 6 2

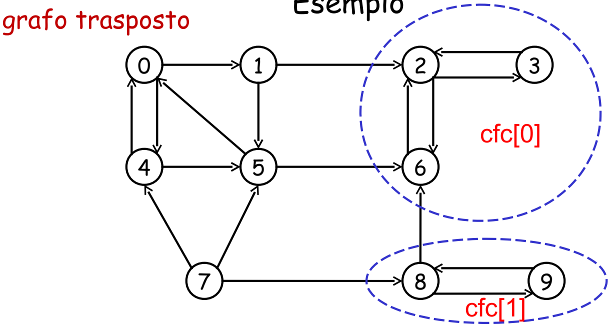
Esempio



1 7 4 5 0 3 9 8 6 2

- prendo l'ultimo nodo della sequenza S, cioè 2
- nel grafo trasposto visito tutti i nodi raggiungibili da esso, e li inserisco nel primo cfc: cfc[0] = {2, 3, 6}

Esempio

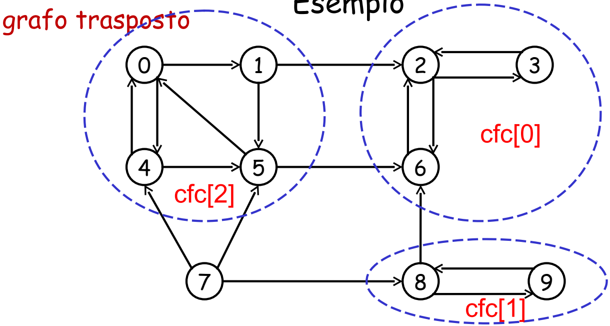


1 7 4 5 0 3 9 8 6 2

• percorrendo la sequenza S all'indietro, prendo il successivo nodo non visitato, cioè 8

• nel grafo trasposto visito tutti i nodi raggiungibili da esso, e li inserisco nel secondo cfc: cfc[1] = {8, 9}

Esempio

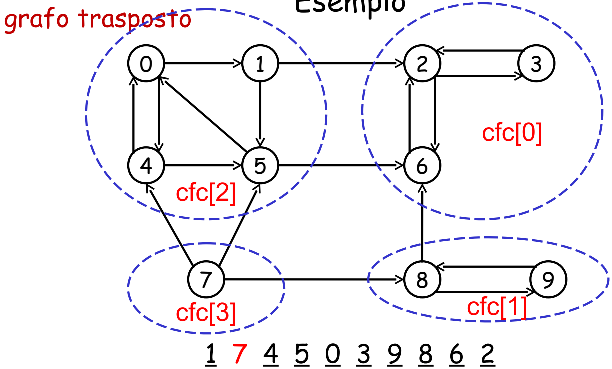


1 7 4 5 0 <u>3</u> <u>9</u> <u>8</u> <u>6</u> <u>2</u>

• percorrendo la sequenza S all’indietro, prendo il successivo nodo non visitato, cioè 0

• nel grafo trasposto visito tutti i nodi raggiungibili da esso, e li inserisco nel terzo cfc: cfc[2] = {0, 1, 5, 4}

Esempio



• percorrendo la sequenza S all'indietro, prendo il successivo nodo non visitato, cioè 7

• nel grafo trasposto visito tutti i nodi raggiungibili da esso, e li inserisco nel terzo cfc: cfc[3] = {7}