Appunti Lezione – Grafi

Alberto Montresor

19 Agosto, 2014

1 DFS

Teorema 1 (Teorema delle parentesi)**.** In qualsiasi visita di profondità di un grafo $G = (V, E)$, per ogni coppia di vertici u, v, una sola delle seguenti condizioni è vera:

1. Gli intervalli $[dt[u], ft[u]]$ e $[dt[v], ft[v]]$ sono disgiunti;
   u, v non sono discendenti l'uno dell'altro nella foresta DF.
2. L'intervallo $[dt[u], ft[u]]$ è interamente contenuto in $[dt[v], ft[v]]$;
   u è discendente di v in un albero DF.
3. L'intervallo $[dt[v], ft[v]]$ è interamente contenuto in $[dt[u], ft[u]]$;
   v è discendente di u in un albero DF.

Dimostrazione.

- $dt[u] < dt[v]$: Ci sono due sottocasi da considerare:
  - $dt[v] < ft[u]$: questo significa che v è stato scoperto dopo il discovery time di u, ma prima del finish time di u; il che significa che v è un discendente di u. Inoltre, poichè v è stato scoperto più recentemente di u, verrà completato prima del completamento di u; ne segue che i vari tempi coinvolti seguono quest'ordine:

    $$dt[u] < dt[v] < ft[v] < ft[u]$$

    quindi l'intervallo $[dt[v], ft[v]]$ è completamente contenuto in $[dt[u], ft[u]]$.
  - $ft[u] < dt[v]$: in questo caso i due intervalli sono completamente disgiunti, in quanto l'ordine ottenuto è il seguente:

    $$dt[u] < ft[u] < dt[v] < ft[v]$$

    Ne segue anche che u e v non sono discendenti l'uno dell'altro.
- $dt[u] > dt[v]$: questo caso è simmetrico, con il ruolo di u e v scambiati.

□

Lemma 1. Durante la DFS di un grafo non orientato G, ogni arco è un arco dell'albero o un arco all'indietro.

Dimostrazione. Sia $[u, v]$ un arco di G e sia $dt[u] < dt[v]$. Allora v deve venir visitato e processato prima che si finisca di processare u (v è nella lista di adiacenza di u). Se l'arco $[u, v]$ viene esplorato prima da u a v, allora diventa un arco dell'albero. Se l'arco (u, v) viene esplorato prima da v a u, allora diventa un arco all'indietro (u è ancora aperto quando l'arco viene esplorato per la prima volta). □

Lemma 2. Teorema: Un grafo orientato è aciclico se e solo se DFS su G non trova alcun arco all'indietro.

Dimostrazione.

- **se**: Se esiste un ciclo, sia v il primo nodo del ciclo che viene visitato e sia (u, v) un arco del ciclo. Il cammino che connette v ad u verrà prima o poi visitato, e da u verrà scoperto l'arco all'indietro (u, v).
- **solo se**: Se esiste un arco all'indietro (u, v), dove v è un antenato di u, allora esiste un cammino da v a u e un arco da u a v, ovvero un ciclo.

□

Lemma 3. L'algoritmo basato su DFS produce correttamente l'ordinamento topologico di un DAG G.

Dimostrazione. Dobbiamo provare che, dopo una DFS, se (u, v) appartiene ad E, allora *ft*$[v]$ $<$ *ft*$[u]$, e quindi u compare prima di v nella lista. Possono darsi tre casi, non essendoci archi all'indietro:

- (u, v) è un arco dell'albero, quindi v è un discendente di u nell'albero DFS e quindi (per il teorema delle parentesi) [*dt*$[v]$,*ft*$[v]$] è contenuta in [*dt*$[u]$,*ft*$[u]$], e quindi *ft*$[v]$ $<$ *ft*$[u]$;
- (u, v) è un arco in avanti, quindi di nuovo v è un discendente di u nell'albero DFS e *ft*$[v]$ $<$ *ft*$[u]$;
- (u, v) è un arco di attraversamento, significa che v è stato già completamento visitato al momento che u viene scoperto, e quindi *ft*$[v]$ $<$ *ft*$[u]$

□

2 Strongly Connected Components

Lemma 4. Un grafo G e il suo trasposto G^T hanno le stesse SCC.

Dimostrazione. Se u, v appartengono alla stessa SCC, è possibile andare da u a v lungo il percorso p_{uv} e da v ad u lungo il percorso p_{vu}. Invertendo le direzioni degli archi, sarà possibile andare da v ad u lungo il cammino p^T_{uv} (l'inverso di p_{vu}), e da u a v lungo il cammino p^T_{vu}. □

2.1 Algoritmo

- chiama $DFS(G)$ e calcola i tempi di completamento ft
- calcola il trasposto G^T di G;
- calcola $DFS(G^T)$, ma nel ciclo principale di DFS, considera i vertici in ordine decrescente rispetto ai tempi ft (calcolati al primo passo)
- Gli insiemi di vertici di ciascun albero DF nella foresta DF generata al passo precedente rappresenta una distinta componenente fortemente connessa.

2.2 Come funziona l'algoritmo

Mostrare l'esempio di funzionamento che si può osservare in figura 2.

Definizione 1 (Grafo delle componenti)**.** Sia $G = (V, E)$ un grafo; il grafo delle componenti $G^{SCC} = (V^{SCC}, E^{SCC})$ è definito nel modo seguente. Sia $V^{SCC} = C_1, C_2, \ldots, C_k$ dove C_i è l'i-esima componente connessa di G. Un arco $(C_i, C_j) \in E^{SCC}$ (con $i \neq j$) se e solo se G contiene un arco orientato (x, y) per qualche $x \in C_i$ e qualche $y \in C_j$.

In altre parole, le componenti sono "contratte" in un singolo vertice, e così gli archi che vanno da componente a componente. Vedi figura 1.
Il concetto di grafo delle componenti è fondamentale per capire il funzionamento dell'algoritmo. Dobbiamo dimostrare una serie di lemmi.

Lemma 5 (Assenza di "doppi" cammini)**.** Siano C e C' due componenti fortemente connesse distinte nel grafo $G = (V, E)$. Se esiste un cammino da $u \in C$ a $u' \in C'$, allora non può esistere un cammino da $v' \in C'$ a $v \in C$.

Dimostrazione. Se tale cammino esistesse, allora esisterebbe (i) un cammino da qualunque nodo di C' a qualunque nodo di C; (ii) un cammino da qualunque nodo di C a qualunque nodo di C'. Quindi, C e C' sarebbero un'unica componente fortemente connessa, contrariamente all'ipotesi. □

Definizione 2 (Estensione delle notazioni ft,dt)**.** Sia U un sottoinsieme di V; definiamo $dt(U)$ come il primo tempo di scoperta e $ft(U)$ come l'ultimo tempo di completamento:

$$
\begin{aligned}
dt(U) &= \min\{\mathit{dt}[u] | u \in U\} \\
ft(U) &= \max\{\mathit{ft}[u] | u \in U\}
\end{aligned}
$$

In base a questo, possiamo enunciare un altro lemma.

Lemma 6. Siano C e C' due SCC distinte nel grafo orientato $G = (V, E)$. Supponiamo che esista un arco $(u, v) \in E$, dove $u \in C$ e $v \in C'$. Allora $ft(C) > ft(C')$.

Dimostrazione. Ci possono essere due casi: viene scoperto prima un nodo di C oppure di C'.

- $dt(C) < dt(C')$: al momento $dt(C)$ viene scoperto un nodo x; tutti i nodi di C e C' non erano ancora stati scoperti fino a quel momento. Poiché esiste un cammino fra x e tutti gli altri nodi di C e C', ne segue che sono tutti discendenti di x . Quindi ne segue che $ft(x)$ è maggiore di tutti gli ft sia di C che di C', quindi $ft(C) > ft(C')$.

- $dt(C') < dt(C)$: sia y il primo nodo scoperto in C'; tutti i nodi in C' sono raggiungibili da y (definizione di SCC) e tutti i nodi in C' non sono ancora stati scoperti; tutti i nodi in C' sono discendenti di y e quindi $ft(C') = ft(y)$. Inoltre, poichè esiste un cammino fra i nodi di C e C', non può esistere un cammino fra i nodi di C' e C. Quindi i nodi di C non sono raggiungibili da y, e verranno scoperti (e terminati) dopo. Quindi $ft(C) > ft(C')$.

□

Corollario 1. Siano C e C' due SCC distinte nel grafo orientato $G = (V, E)$. Supponiamo che esista un arco $(u, v) \in E^T$, dove $u \in C$ e $v \in C'$. Allora $ft(C) < ft(C')$.

Dimostrazione. Poiché $(u, v) \in E^T$, si ha $(v, u) \in E$. Poichè le SCC di G sono le stesse di G^T, il lemma precedente impica che $ft(C) < ft(C')$. □

Questo corollario è fondamentale per capire il funzionamento dell'algoritmo.

- Iniziamo con la componente connessa con tempo di completamento maggiore. Supponiamo che esista un vertice in E^T che va verso un'altra componente. Questo è assurdo, in quanto quest'altra componente dovrebbe avere tempo di completamento superiore, impossibile per definizione.

- La visita passa quindi per tutti i vertici della prima componente, e solo quelli.

- Nei passi successivi, verranno selezionato un vertice con tempo di completamento maggiore fra quelli non già visitati. Questo appartiene ad una componente che può avere cammini solo verso componenti con tempo di completamento superiore, quindi già visitati. Quindi verranno visitati solo i nodi di quella componente.