Ordinamento topologico

Vediamo in maggior dettaglio come si risolve il problema dell'ordinamento topologico mediante una visita in profondità. Supponiamo che $G = (V, E)$ sia un grafo diretto con n vertici. Di seguito viene proposta una versione più pulita dello pseudo-codice presentato nelle slide. Si noti che supponiamo che a ogni vertice sia associato un campo `rank` destinato a contenere il suo numero topologico.

```
Algorithm topologicalDFS(G){
    // Input dag G
    // Output topological ordering of G
    n = G.numVertices();
    for (u in G.vertices())
        setLabel(u, UNEXPLORED);
    for (v in G.vertices())
        if (getLabel(v) = UNEXPLORED)
            topologicalDFS(G, v);
}

Algorithm topologicalDFS(G, v) {
    // Input graph G and a start vertex v of G
    // Output labeling of the vertices of G
    // in the connected component of v
    setLabel(v, EXPLORING);
    for (e in G.outEdges(v)) { // outgoing edges
        w = opposite(v,e);
        if (getLabel(w) == UNEXPLORED) // e is a discovery edge
            topologicalDFS(G, w);
        // Altrimenti e è un arco forward o cross in un DAG
    }
    setLabel(v, EXPLORED); // Qui avviene in effetti la visita di v
    v.rank = n; // Il numero topologico di v è n
    n = n - 1; // Variabile globale
}
```

Di seguito dimostriamo la seguente affermazione:

Se G è un grafo diretto aciclico, l'algoritmo topologicalDFS ne calcola un possibile ordinamento topologico.

Prova. La prova consiste nel mostrare che, se $(u, v) \in E$, allora $u.rank < v.rank$. Ricordiamo innanzi tutto che, esaminiamo l'albero della DFS, ogni vertice w è visitato *dopo* tutti i suoi discendenti. Ricordiamo anche la classificazione degli archi di un grafo rispetto alla visita DFS. Un arco (w, z) può essere dei tipi seguenti:

- *Arco tree:* quando (w, z) è *anche* un arco dell'albero della DFS (in sostanza un arco di tipo *discovery*). Si noti che in tal caso z è visitato prima di w nella DFS.
- *Arco forward*: in tal caso z è un *discendente* di w nell'albero DFS, ma non appartiene all'albero. Si noti che anche in tal caso z è visitato prima di w nella DFS.
- *Arco cross*: in tal caso i sotto-alberi radicati in w e z sono disgiunti (w e z non sono discendenti l'uno dell'altro). Si noti che in tal caso, z è stato visitato prima ancora che si iniziasse la visita del sotto-albero avente radice w.

- *Arco back*: in tal caso z è un antenato di w nell'albero DFS. Si noti che l'esistenza di un arco di tale tipo implica quella di un ciclo. Ciò significa che non possono esistere archi di tipo back nella DFS effettuata su un grafo diretto aciclico.

A questo punto possiamo riprendere la nostra prova, considerando i tre casi possibili:

1. Se (u, v) è un arco di tipo cross, allora `topologicalDFS(G, v)` è stata completata prima ancora che `topologicalDFS(G, u)` iniziasse e dunque $rank.v > rank.u$.
2. Negli altri due casi (archi tree e forward), v è un discendente (nel primo caso diretto) di u e quindi è visitato prima di quest'ultimo. Per rendersene conto è sufficiente guardare il codice presentato sopra: la visita (e assegnazione del relativo numero topologico) di un vertice avvengono *dopo* che tutti i vertici raggiungibili da quel vertice e non ancora esplorati sono stati visitati e numerati, ricevendo quindi numeri topologici più alti di quelli del vertice considerato.