Depth first search

Julián Mestre

In this note we review some of the properties of depth first search (DFS) and describe a linear-time DFS-based algorithm for computing strongly connected components.

The high idea behind the DFS traversal is very simple: explore edges that lead to unvisited vertices; if a "dead end" is reached, backtrack. Below is the pseudo-code for the DFS routines.

```
DFS(G)

counter = 0
For u ∈ V[G]
  Mark u as not visited
For u ∈ V[G]
  If u was not visited
    parent[u] = nil
    DFS-visit(u)
```

```
DFS-visit(u)

Mark u as visited
counter = counter + 1
d[u] = counter
For v adjacent to u
  If v was not visited
    parent[v] = u
    DFS-visit(v)
counter = counter + 1
f[u] = counter
```

Define the DFS forest as $\{(\texttt{parent}[u], u) \mid \forall u \in V \ \texttt{parent}[u] \neq nil\}$. In class we proved many interesting properties about $\texttt{f}[u]$ and $\texttt{d}[u]$, the *finishing and discovery time* of a vertex u. The following lemmas hold for both directed and undirected graphs.

Lemma 1 (nesting property)**.** *Let u be a vertex in the graph, and $v_1, \ldots, v_k$ the children of u in the DFS forest. The intervals $[\texttt{d}[v_i], \texttt{f}[v_i]]$ are pair-wise disjoint and span $(\texttt{d}[u], \texttt{f}[v])$.*

Lemma 2 (descendant property)**.** *Let u and v be vertices in the graph. Vertex v is a descendant of u in the DFS forest if and only if $\texttt{d}[u] < \texttt{d}[v] < \texttt{f}[v] < \texttt{f}[u]$.*

Lemma 3 (unvisited path property)**.** *Suppose that right before making the call `DFS-visit(u)` there was a path $u \rightsquigarrow v$ using unvisited vertices. Then v will become a descendant of u in the DFS forest.*

Let us use the above properties to argue the correctness of the following elegant algorithm for computing the strongly connected components (SCC) of a directed graph.

```
Find strongly connected components(G)
```
i) call DFS(G) to compute finishing times $f[u]$ for every vertex u.

ii) call DFS(G^{rev}) processing vertices in the main loop in decreasing order of $f[u]$.

iii) output each tree in the second DFS forest as a separate component.

Because of Lemma 3 we know that each SCC is contained within a single tree in the DFS forest. This is good, because we do not want a component to be split across different trees.

But note that a given tree may, in principle, contain more than one SCC; we must show that at most one is contained.

Lemma 4. *Every tree in the second DFS forest corresponds to exactly one SCC.*

Proof. Let T be a tree in the DFS forest. Suppose that r, the root of T, belongs to the strongly connected component C. Thus $C \subseteq V[T]$, we need to argue that $C = V[T]$.

How can DFS exit component C? By taking an edge $(u, v) \in E[G^{rev}]$ such that $u \in C$ and v belongs to some other component C'. However, the edge will not be used if component C' had already been visited before the the call `DFS-visit(r)` was made. One way this could happen is if

$$\max_{w \in C'} \mathtt{f}[w] > \max_{w \in C} \mathtt{f}[w]. \tag{1}$$

Let us see what happens in the first DFS execution. Let $x \in C'$ and $y \in C$ be the vertices in C' and C with smallest discovery time. By Lemmas 2 and 3 it follows that $\mathtt{f}[x] = \max_{w \in C'} \mathtt{f}[w]$ and $\mathtt{f}[y] = \max_{w \in C} \mathtt{f}[w]$. Thefore, condition (1) holds provided

$$\mathtt{f}[x] > \mathtt{f}[y]. \tag{2}$$

Consider the following two cases:

case 1: $\mathtt{d}[x] < \mathtt{d}[y]$. Because C' and C are strongly connected when the call `DFS-visit(`x`)` was made there was a path $x \rightsquigarrow u \rightarrow v \rightsquigarrow y$ using unvisited nodes. Applying Lemmas 2 and 3 we get that $\mathtt{f}[y] < \mathtt{f}[x]$.

case 2: $\mathtt{d}[y] < \mathtt{d}[x]$. Suppose $\mathtt{f}[x] < \mathtt{f}[y]$, by Lemma 2 vertex x must be a descendant of y; this means there is path $y \rightsquigarrow x$, but we just argued that there is a path $x \rightsquigarrow y$. This contradicts the fact that x and y are in different strongly connected components. It must be that $\mathtt{f}[y] < \mathtt{f}[x]$.

Hence DFS never leaves C and the Lemma follows. □