18 Sharir's SCC algorithm

Let $G = (V, E)$ be a directed graph. The relation on V, 'v is reachable from u' graph is reflexive and transitive. The relation 'v is reachable from u and u is reachable from v' is an equivalence relation on V.

Let $V = V_1 \cup \ldots V_k$ be the partition of V into equivalence classes under this relation. For each j, the subgraph of G spanned by V_k, i.e.,

$$(V, \{(u, v) \in E : \; u \in V_k \wedge v \in V_k\})$$

is called a *strong component* of V.

We shall concentrate on finding the sets V_j of vertices of strong components without constructing the graphs they span, and refer to V_j as strong components.

(18.1) Lemma *Suppose that V_k is a strong component and depth-first search is performed. Let $u \in V_k$ be the first vertex in V_k visited during dfs. Then every vertex in V_k is a descendant of u in the depth-first forest.*

Proof. When dfs(u) begins, no other vertex of V_k has been visited, and every vertex of V_k is reachable from u by a path which contains no vertex visited before u.

By the main DFS property, every vertex in V_k will be a descendant of u after dfs(u) ends. ∎

It follows that for every tree T of the depth-first forest, the vertices in the tree are a union of SCCs, i.e.,

$$V_{i_1} \cup \ldots \cup V_{i_\ell}$$

for some $i_1, \ldots, i_\ell$.

18.1 Sharir's algorithm

Sharir's algorithm is as follows. To determine the SCCs (vertex sets) of $G = (V, E)$,

- Perform a full DFS of G, for one purpose: to obtain the resulting postorder ranking of the vertices in G.
- Make a reversed copy G' of G. That is, with $G = (V, E)$,

$$G' = (V, \{(v, u) : \; (u, v) \in E\})$$

- Perform a full DFS of G', but the order in which you select new vertices for DFS is in *reversed postorder* constructed from the full DFS of G.
- Let $T_1, \ldots, T_k$ be the trees constructed in the second full DFS. For $1 \leq j \leq k$, the vertices in T_j span one of the strong components.

Figure 1: Directed graph and its reversal for Sharir's algorithm

18.2 Example of Sharir's algorithm

```
6 7
0 2 2 5
1 1 3
2 1 4
3 1 5
4 1 0
5 1 1
  dfs(0) begins
    dfs(2) begins
      dfs(4) begins
      dfs(4) ends
    dfs(2) ends
    dfs(5) begins
      dfs(1) begins
        dfs(3) begins
        dfs(3) ends
      dfs(1) ends
    dfs(5) ends
  dfs(0) ends
   digraph extras, 6 vertices
   vertex    parent  pre_rank post_rank
        0        -1         0         5
        1         5         4         3
        2         0         1         1
        3         1         5         2
        4         2         2         0
        5         0         3         4

Notice that all vertices are reachable from 0, so
there is just on depth-first tree.  However, there
are two SCCs.
```

```
6 7 (reversed digraph)
0 1 4
1 1 5
2 1 0
3 1 1
4 1 2
5 2 0 3
The following is constructed from the original
postorder rank.

verts in reversed postorder 0 5 1 3 2 4
  2nd dfs(0) begins
    2nd dfs(4) begins
      2nd dfs(2) begins
      2nd dfs(2) ends
    2nd dfs(4) ends
  2nd dfs(0) ends
SCC 0 4 2 --- first tree in second dfs
  2nd dfs(1) begins
    2nd dfs(5) begins
      2nd dfs(3) begins
      2nd dfs(3) ends
    2nd dfs(5) ends
  2nd dfs(1) ends
SCC 1 5 3 --- second tree in second dfs
   digraph extras, 6 vertices
   vertex    parent  pre_rank post_rank
        0        -1         0         2
        1        -1         3         5
        2         4         2         0
        3         5         5         3
        4         0         1         1
        5         1         4         4
%
```

18.3 Correctness of Sharir's algorithm

(18.2) Lemma *Sharir's algorithm is correct in the sense that for every depth-first tree T built in the second dfs, the vertices in T coincide with the vertices in one of the SCCs of G.*

Proof. G and its reversal G' may have different edge-sets, but if W is the set of nodes spanning a SCC of G then it spans a SCC of G', and vice-versa.

Therefore every such tree T includes one or more SCCs of G. Suppose that some tree includes more than one SCC.

Let u be the root of T, belonging to an SCC U.

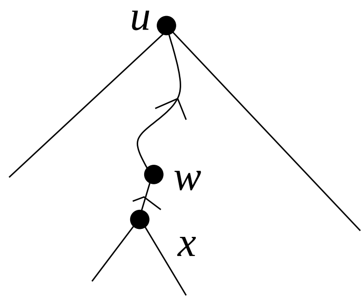

Figure 2: Correctness of Sharir's algorithm

- Every vertex in T is connected to u by a path in G, because every descendant of u is reachable from u by a path in G'.
- Let x be a vertex in T which belongs to a different SCC than u, and suppose x is as close to u as possible: so the parent w of x, and all ancestors of w in T, are in U.
- By choice of u in the second dfs, u has first-dfs postorder rank greater than all other vertices in U. But then it is a first-dfs ancestor of every vertex in U, and has minimal preorder rank.
- Since (x, w) is an edge of the original digraph, if w had been visited after x in the first DFS, then w, and every vertex in U, would have been a descendant of x in the first dfs, and x would have had higher postorder rank than u.
- Therefore x is visited after w in the first dfs, and w has lower (first dfs) preorder rank than x. But the (first dfs) preorder rank of u is $\leq$ that of w, because $w \in U$.

  So, after the first dfs, u precedes x in preorder and, by choice of u over x, follows x in postorder. This means that x is a descendant of u, and that x is reachable from u in G.

  Since x is reachable from u in G', u is reachable from x in G.

  Therefore x and u are in the same SCC U, a contradiction. ∎