Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Dmytro Chibisov
Sandeep Sadanandan

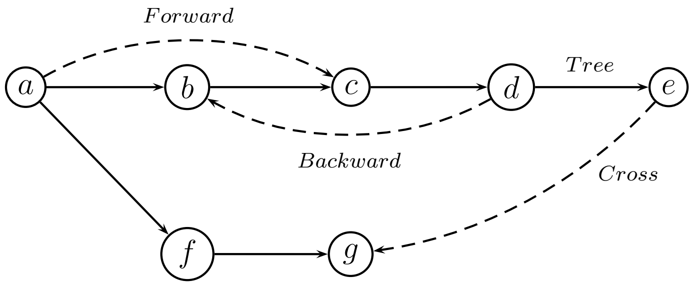
Winter Semester 2007/08
Solution Sheet 11
January 23, 2007

Fundamental Algorithms

Problem 1 (10 Points)

Explain the four different types of edges in a graph with a figure.

Solution



There are 4 different types of edges:

Tree Edges : The edges in the DFS tree

Back Edges : The edges connecting u to an ancestor

Forward Edges : The NON-TREE edges connecting u to a descendant

Cross Edges : All the other types

Problem 2 (10 Points)

What are the variables $DFSNum$ and $FinishNum$ in topological sorting?

Solution

For keeping track of the order in which DFS is happening, we assign something called $DFSNum$ and $FinishNum$ for each node of the graph.

$DFSNum$ is the variable which says when was the node first visited by the DFS. The initial node will have the smallest $DFSNum$ value. The node which was visited at the end will have the largest $DFSNum$ value.

$FinishNum$ is the variable which says when was a node finished. Which means the node and all its neighbors are visited and the DFS call is returning. It is evident that the DFS will return from the initial node at the end - which means, the initial node will have the largest value for $FinishNum$. And the node which was visited last will be the one which will be finished first – implying that it will have the smallest $FinishNum$ value.

There are two global variables $DFSPos$ and $DFSFinish$ which would be initialized to one. Every time a new node is visited, the value of $DFSPos$ is assigned to the $DFSNum$ of the node and $DFSPos$ is incremented. Similarly, when the search is leaving a node, the $DFSFinish$ value is assigned to $FinishNum$ and $DFSFinish$ is incremented.

Problem 3 (10 Points)

How can one classify an edge of a graph if the values of $DFSNum$ and $FinishNum$ of the nodes connected by it are known?

Solution

Let the edge be (u, v).

- Let (u, v) be a Tree edge. This means u was visited before v. But only after finishing v has the DFS come to u. Which simply means u will have a smaller $DFSNum$ while having a larger $FinishNum$.
- Let (u, v) be a Forward edge. This means again that u was visited before v. The case is exactly the same as above.
- (u, v) be a Backward edge. So v is an ancester of u. This implies u will have a larger $DFSNum$ while having a smaller $FinishNum$
- (u, v) be a Cross edge. Here v and u are in different subtrees of the DFS tree. It is also clear that v was visited before u; otherwise v would have been a descendant of u. So v has a smaller $DFSNum$. As they are in different subtrees, v was also finished before u was visited – implying that $FinishNum$ of v is smaller than that of u.

The result can be shown in a table as below.

Type	$u.DFSNum < v.DFSNum$	$u.FinishNum > v.FinishNum$
Tree	Yes	Yes
Forward	Yes	Yes
Backward	No	No
Cross	No	Yes

Problem 4 (10 Points)

Show that in an undirected graph, there are neither Forward nor Cross edges.

Solution

Let there be an edge (u, v). Without loss of generality, assume that $v.DFSNum$ is larger than $u.DFSNum$. That implies that u was visited before v.

Now there are only two cases:

1. The only path from u to v is (u, v).

   In this case, v will be visited directly from u and the edge is simply a Tree edge.

2. There is a path other than (u, v) from u to v.

   Here we have two subcases:

   (a) (u, v) was taken to visit v from u - which means (u, v) is a Tree edge.

   (b) The other path was chosen first and the DFS reached v. So now (u, v) will be used to traverse from v to u and will u will be recognised as an ancestor (because it is still not finished). This means (u, v) is now a Back edge.

Since these are the only two cases, there cannot be any other types of edges. So, Forward and Cross edges will be absent.

Problem 5 (10 Points)

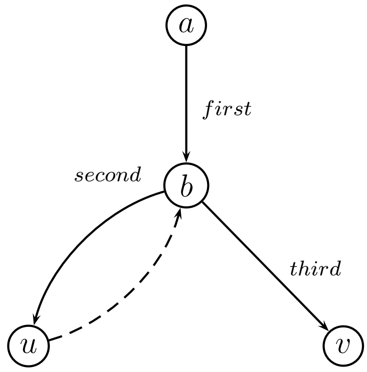
Disprove the conjectures:

1. If there is a path from u to v in a directed graph, and if $u.DFSNum < v.DFSNum$ then v is a descendant of u.

2. If there is a path from u to v in a directed graph, then any DFS must result in $v.DFSNum \leq u.FinishNum$.

Solution

1. $u.DFSNum = 3 < v.DFSNum = 4$ but v is not a descendant of u.



2. The same example which is given above is sufficient.

   $u.FinishNum = 1 < v.DFSNum = 4$.