```
void search(n) {
    mark n "visited";
    for (each successor s of n)
        if (s is "unvisited") {
            add edge n → s to T;
            search(s);
        }
    dfn[n] = c;
    c = c − 1;
}

main() {
    T = ∅; /* set of edges */
    for (each node n of G)
        mark n "unvisited";
    c = number of nodes of G;
    search(n0);
}
```

Figure 9.43: Depth-first search algorithm

3. There are edges $m \to n$ such that neither m nor n is an ancestor of the other in the DFST. Edges $2 \to 3$ and $5 \to 7$ are the only such examples in Fig. 9.42. We call these edges *cross edges*. An important property of cross edges is that if we draw the DFST so children of a node are drawn from left to right in the order in which they were added to the tree, then all cross edges travel from right to left.

It should be noted that $m \to n$ is a retreating edge if and only if $dfn[m] \geq dfn[n]$. To see why, note that if m is a descendant of n in the DFST, then $search(m)$ terminates before $search(n)$, so $dfn[m] \geq dfn[n]$. Conversely, if $dfn[m] \geq dfn[n]$, then $search(m)$ terminates before $search(n)$, or $m = n$. But $search(n)$ must have begun before $search(m)$ if there is an edge $m \to n$, or else the fact that n is a successor of m would have made n a descendant of m in the DFST. Thus the time $search(m)$ is active is a subinterval of the time $search(n)$ is active, from which it follows that n is an ancestor of m in the DFST.

9.6.4 Back Edges and Reducibility

A *back edge* is an edge $a \to b$ whose head b dominates its tail a. For any flow graph, every back edge is retreating, but not every retreating edge is a back edge. A flow graph is said to be *reducible* if all its retreating edges in any depth-first spanning tree are also back edges. In other words, if a graph is reducible, then all the DFST's have the same set of retreating edges, and

```
void search(n) {
        mark n "visited";
        for (each successor s of n)
                if (s is "unvisited") {
                        add edge n → s to T;
                        search(s);
                }
        dfn[n] = c;
        c = c − 1;
}

main() {
        T = ∅; /* set of edges */
        for (each node n of G)
                mark n "unvisited";
        c = number of nodes of G;
        search(n_0);
}
```

Figure 9.43: Depth-first search algorithm

3. There are edges $m \to n$ such that neither m nor n is an ancestor of the other in the DFST. Edges $2 \to 3$ and $5 \to 7$ are the only such examples in Fig. 9.42. We call these edges *cross edges*. An important property of cross edges is that if we draw the DFST so children of a node are drawn from left to right in the order in which they were added to the tree, then all cross edges travel from right to left.

It should be noted that $m \to n$ is a retreating edge if and only if $dfn[m] \geq dfn[n]$. To see why, note that if m is a descendant of n in the DFST, then $search(m)$ terminates before $search(n)$, so $dfn[m] \geq dfn[n]$. Conversely, if $dfn[m] \geq dfn[n]$, then $search(m)$ terminates before $search(n)$, or $m = n$. But $search(n)$ must have begun before $search(m)$ if there is an edge $m \to n$, or else the fact that n is a successor of m would have made n a descendant of m in the DFST. Thus the time $search(m)$ is active is a subinterval of the time $search(n)$ is active, from which it follows that n is an ancestor of m in the DFST.

9.6.4 Back Edges and Reducibility

A *back edge* is an edge $a \to b$ whose head b dominates its tail a. For any flow graph, every back edge is retreating, but not every retreating edge is a back edge. A flow graph is said to be *reducible* if all its retreating edges in any depth-first spanning tree are also back edges. In other words, if a graph is reducible, then all the DFST's have the same set of retreating edges, and