Static Program Analysis Part II

Control Flow Graph

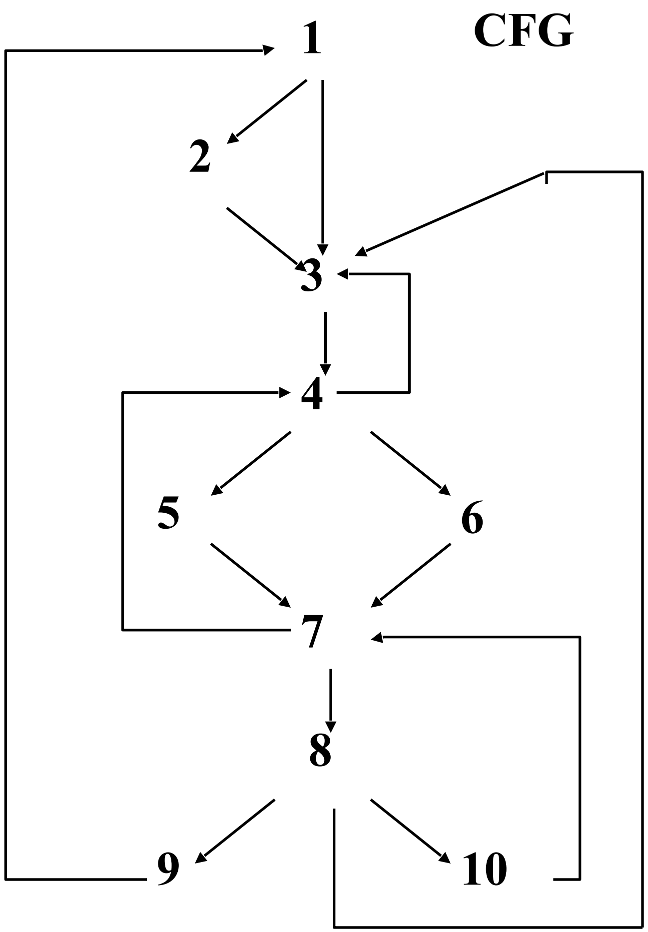
- A control flow graph $CFG = (N, E)$ is a directed graph
- $N = \{n_1, n_2, \ldots n_k\}$ is a finite set of nodes (basic blocks of a program)
- $E = \{(n_i, n_j) \mid n_i, n_j\ N$ *& the flow of control goes from* n_i *to* $n_j\}$

Dominators

- Given a Control Flow Graph (CFG) with nodes D and N:
  - D dominates N if every path from the initial node to N goes through D

- Properties of dominance:
  1. Every node dominates itself
  2. Initial node dominates all others

Dominators - example

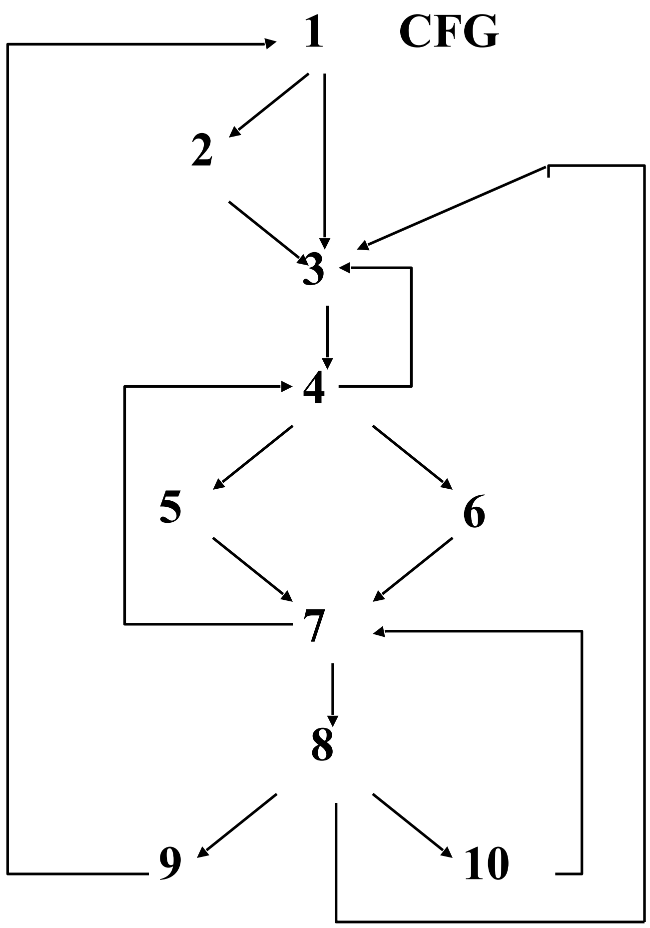


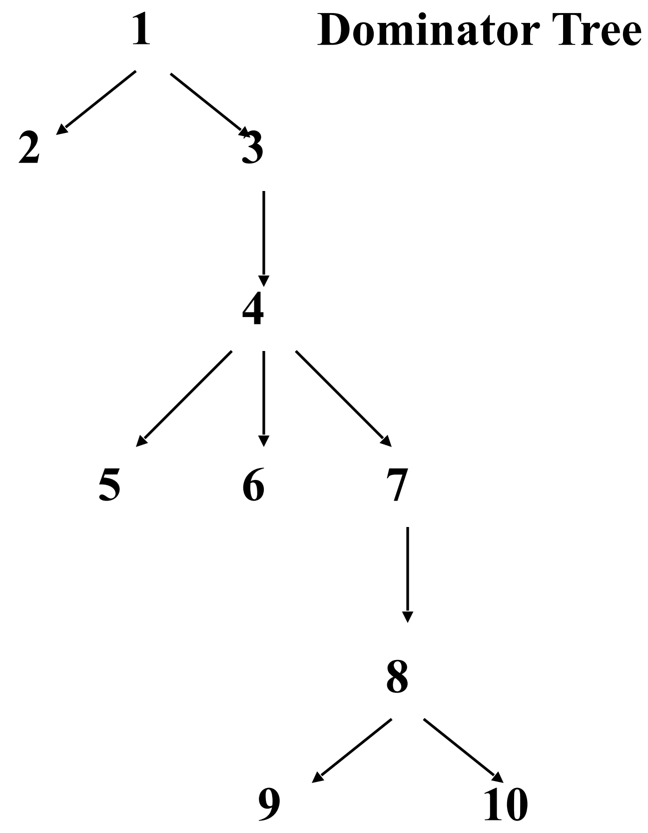
Node	Dominates
1	1,2,…,10
2	2
3	3,4,5,6,7,8,9,10
4	4,5,6,7,8,9,10
5	5
6	6
7	7,8,9,10
8	8,9,10
9	9
10	10

Dominator Trees

- In a dominator tree
  - The initial node *n* is the root of the Control Flow Graph
  - The parent of a node n is its *immediate dominator* (i.e., the last dominator of *n* on any path); the immediate dominator for *n* is unique

Dominators - dominator tree example



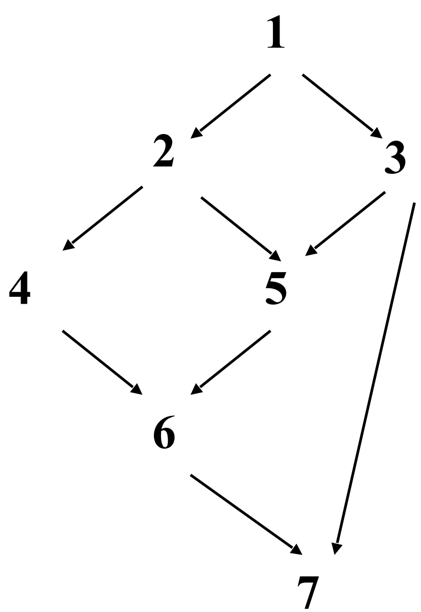


Post-Dominators

- Given a Control Flow Graph with nodes PD and N:
  - PD post dominates N if every path from N to the final nodes goes through PD

Post-Dominators - Example

CFG



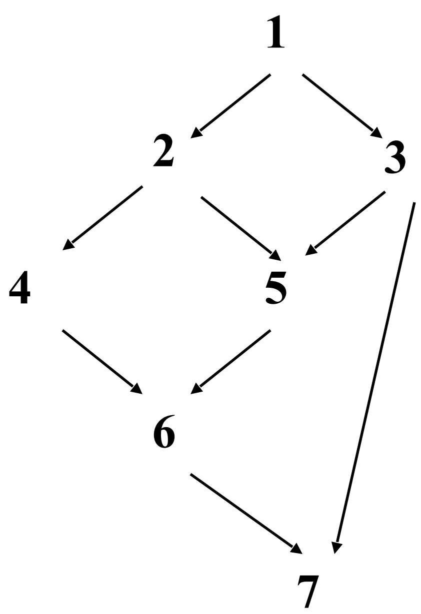
Node	Postdominates
1	--
2	--
3	--
4	--
5	--
6	2,4,5
7	1,2,3,4,5,6

Post Dominators - Dominator Tree

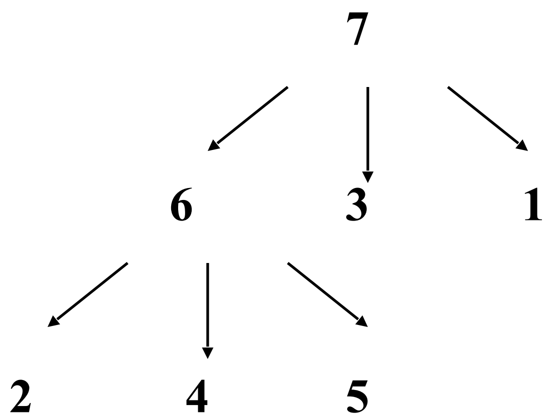
- In a post dominator tree
  - The initial node *n* is the exit node of the Control Flow Graph
  - The parent of a node n is its *immediate post dominator* (i.e., the first post dominator of *n* on any path); the immediate post dominator for *n* is unique

Post Dominators - Dominator Tree Example

CFG

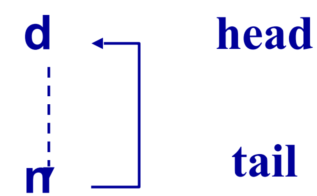


Post dominator Tree



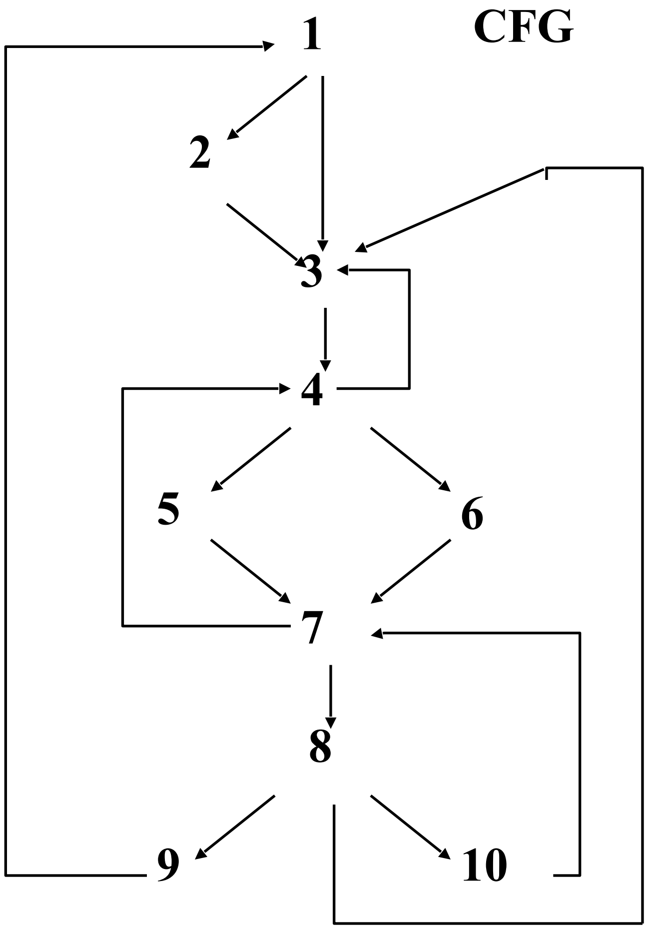
Finding Loops

- We'll consider what are known as natural loops
  - Single entry node (header) that *dominates* all other nodes in the loop
  - The nodes in the loop form a strongly connected component, that is, from every node there is at least one path back to the header
  - There is a way to iterate - there is a back edge (n,d) whose target node d (called the head) dominates its source node n (called the tail)



- If two back edges have the same target, then all nodes in the loop sets for these edges are in the same loop

Loops - Example



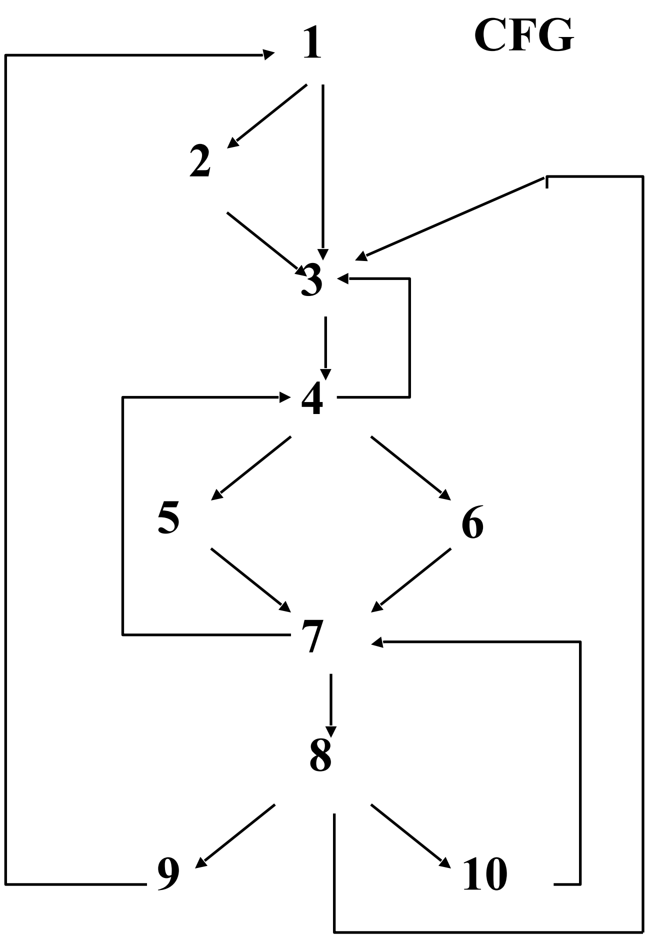
Which edges are back edges?

4 → 3
7 → 4
10 → 7
9 → 1
8 → 3

Construction of loops

1. Find dominators in Control Flow Graph
2. Find back edges
3. Traverse back edge in reverse execution direction until the target of the back edge is reached; all nodes encountered during this traversal form the loop. The result is all nodes that can reach the source of the edge without going through the target

Loops - Example



Back Edge	Loop Induced
4 → 3	{3,4,5,6,7,8,10}
7 → 4	{4,5,6,7,8,10}
10 → 7	{7,8,10}
8 → 3	{3,4,5,6,7,8,10}
9 → 1	{1,2,…,10}

Applications of Control Flow

- Complexity
  - Cyclomatic (McCabe's) - Indication of number of test case needed; indication of difficulty of maintaining
- Testing
  - branch, path, basis path
- Program understanding
  - program structure and flow is explicit