A Worklist Algorithm for Dominators

The data flow equations we have developed for dominators can be evaluated using a simple Worklist Algorithm.

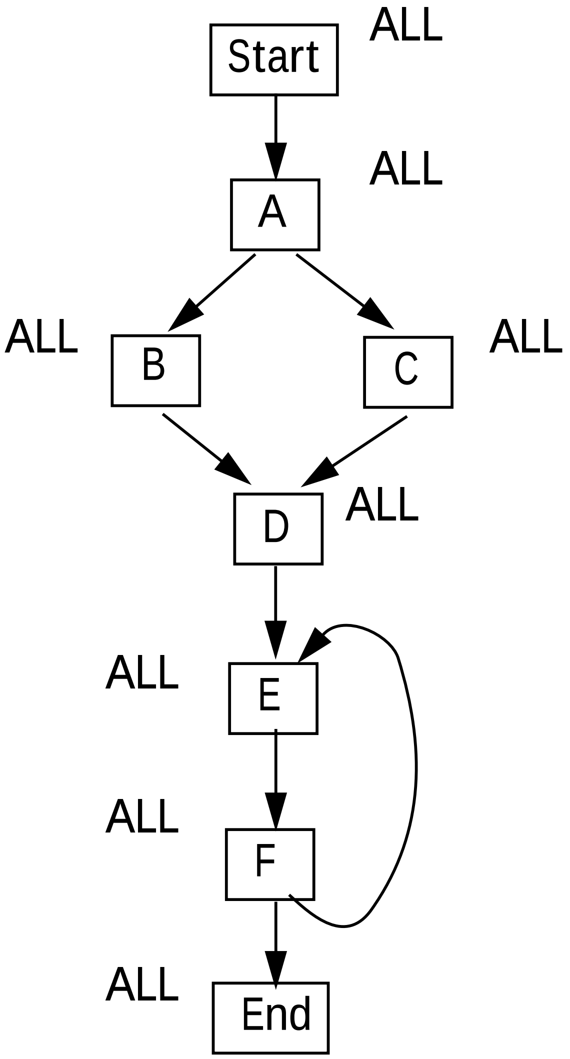
Initially, each node’s dominator set is set to the set of all nodes. We add the start node to our worklist.

For each node on the worklist, we reevaluate its dominator set. If the set changes, the updated dominator set is used, and all the node’s successors are added to the worklist (so that the updated dominator set can be propagated).

The algorithm terminates when the worklist becomes empty, indicating that a stable solution has been found.

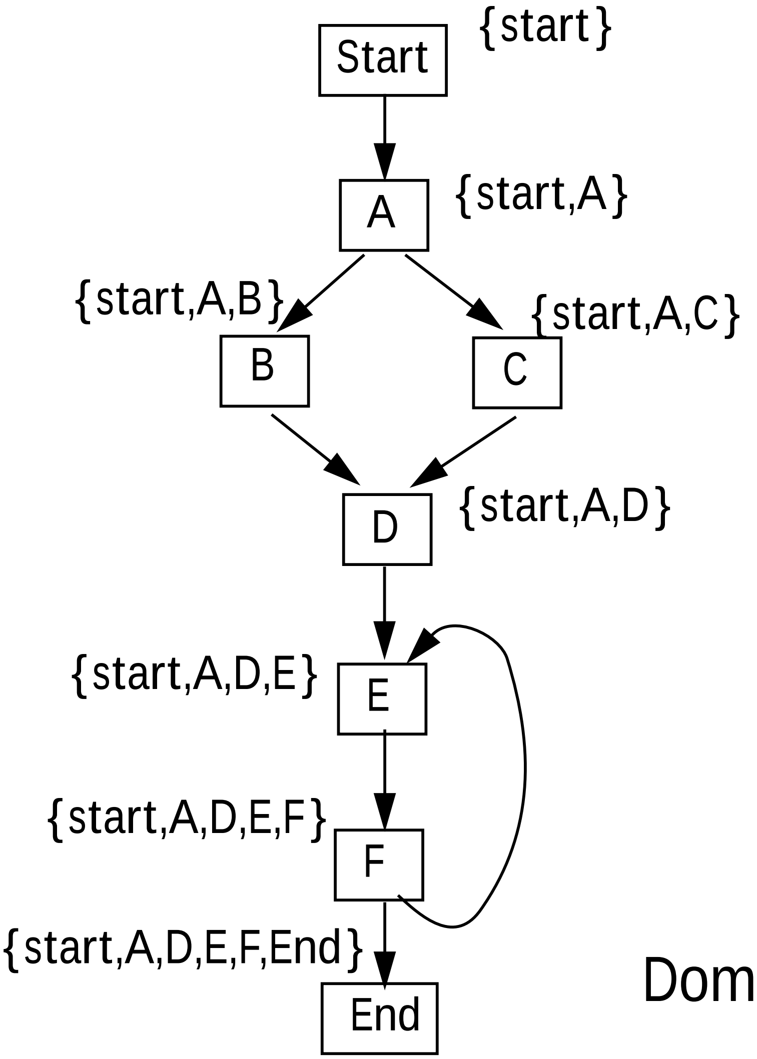
```
Compute Dominators(){
  For (each $n \in$ NodeSet)
      Dom(n) = NodeSet
  WorkList = {StartNode}
  While (WorkList $\neq \phi$) {
    Remove any node Y from WorkList
    New = {Y} U $\bigcap_{X \in \text{Pred}(Y)}$ Dom(X)
    If New $\neq$ Dom(Y) {
       Dom(Y) = New
       For (each $Z \in$ Succ(Y))
           WorkList = WorkList U {Z}
}}}
```

Example

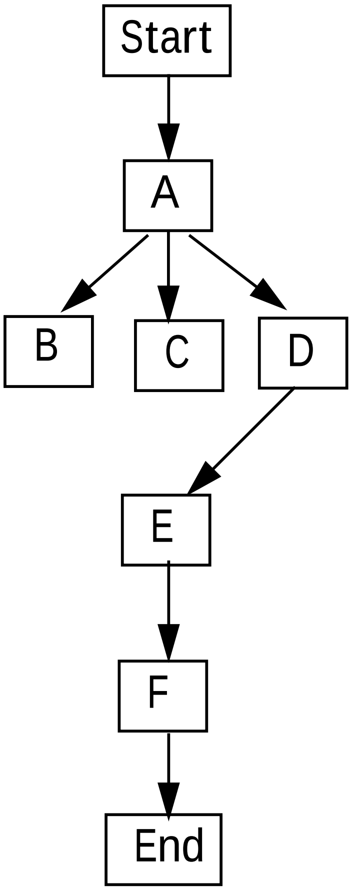


Initially the WorkList = {Start}.

Be careful when Pred(Node) = ϕ.



Control Flow Graph

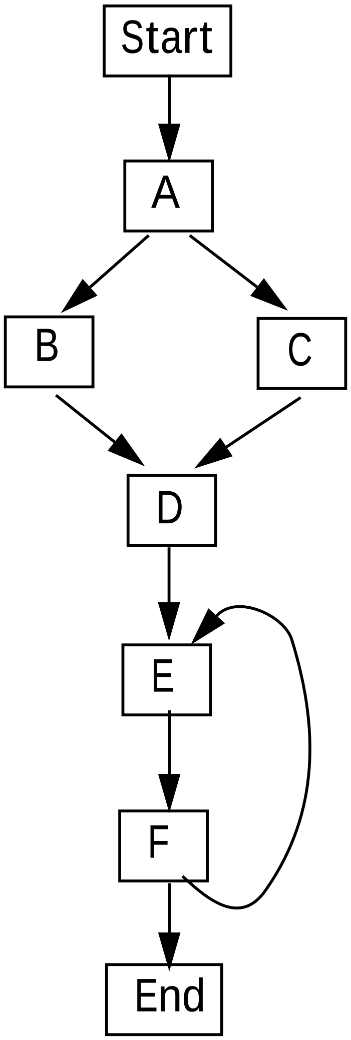


Dominator Tree

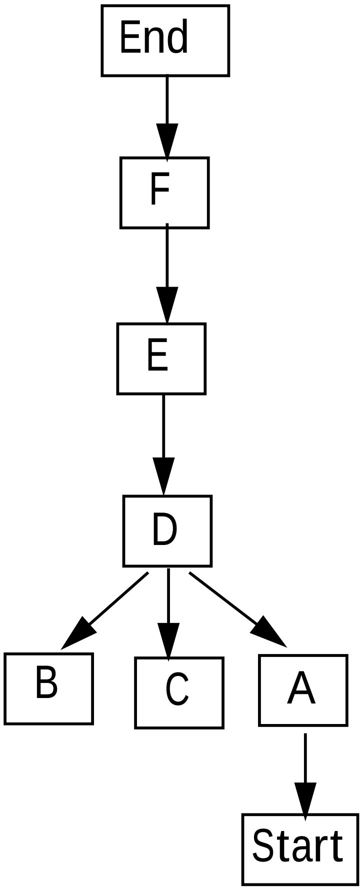
Postdominance

A block Z *postdominates* a block Y (Z pdom Y) if and only if all paths from Y to an exit block must pass through Z. Notions of immediate postdominance and a postdominator tree carry over.

Note that if a CFG has a single exit node, then postdominance is equivalent to dominance if flow is reversed (going from the exit node to the start node).



Control Flow Graph



Postdominator Tree