Dominators

Definition

In a CFG, node a dominates b if every path from the start node to b passes through a. Node a is a dominator of b.

Property

The dominance relation is a partial order.

Definition

Node a strictly dominates b if $a \neq b$ and a dominates b.

Dominators

Theorem

IF a and b both dominate c,
THEN either a dominates b or b dominates a.

Dominators

Theorem

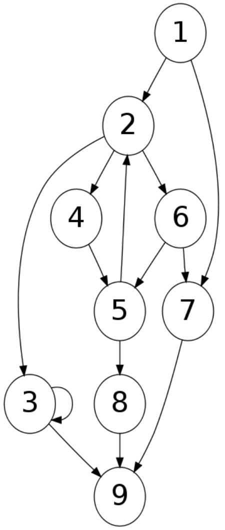
IF a and b both dominate c,
THEN either a dominates b or b dominates a.

Corollary

Every node n has at most one immediate dominator $\text{idom}(n)$ such that

- $\text{idom}(n) \neq n$
- $\text{idom}(n)$ dominates n, and
- $\text{idom}(n)$ does not dominate any other dominator of n.

Dominator Example



Computing Dominators

As a dataflow analysis

1. Forwards
2. Lattice is $(\mathcal{P}(Stmts), \supseteq)$
3. $\cap$
4. $\text{out}_\ell = \text{in}_\ell \cup \{\ell\}$
5. start node value is $\{\}$
6. $\bot = \{\text{all statements}\}$

Computing Dominators

As a dataflow analysis

1. Forwards
2. Lattice is $(\mathcal{P}(Stmts), \supseteq)$
3. $\cap$
4. $\text{out}_\ell = \text{in}_\ell \cup \{\ell\}$
5. start node value is $\{\}$
6. $\bot = \{\text{all statements}\}$

More efficient approaches

- Lengauer-Tarjan: see Appel book section 19.2
- Cooper, Harvey, Kennedy: `http://www.hipersoft.rice.edu/grads/publications/dom14.pdf`

Dominance Frontier

Definition

A node w is in the dominance frontier of x if:

- x does not strictly dominate w, and
- x dominates a predecessor of w.

Computing Dominance Frontier

$DF_{local}(x)$: the successors of x not strictly dominated by x.
$DF_{up}(y)$: nodes in $DF(y)$ not strictly dominated by $\mathrm{idom}(y)$.
$DF(x) = DF_{local}(x) \cup \bigcup_{\{y \mid \mathrm{idom}(y)=x\}} DF_{up}(y)$.

Computing Dominance Frontier

Algorithm $\mathrm{DF(x)}$:

```
S = {}
for all nodes w ∈ succ(x) do
    if idom(w) ≠ x then
        S ∪= {w}
/* S is now DF_local(x) */
for all nodes y for which idom(y) = x do
    /* below we compute DF_up(y) */
    for all nodes w ∈ DF(y) do
        if x does not dominate w or x = w then
            S ∪= {w}
return S
```

Computing Dominance Frontier (Alternative)

Restatement of definition of DF

$w \in DF(x)$ for every x that dominates a predecessor of w, but does not strictly dominate w.

Algorithm COMPUTE DFS():

1: **for all** nodes w **do**
2:     **for all** $p \in \text{preds}(w)$ **do**
3:         $x = p$
4:         **while** $x \neq \text{idom}(w)$ **do**
5:             $DF(x) \cup = \{w\}$
6:             $x = \text{idom}(x)$