Class 5

- Review; questions
- Assign (see Schedule for links)
  - Slicing overview
  - Problem Set 2: due 9/3/09
  - Problem Set 3: due 9/8/09

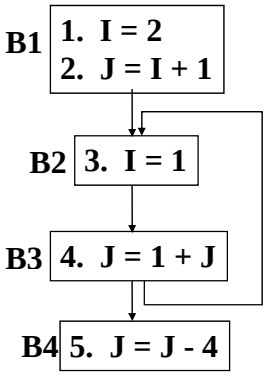
1

Data-flow Analysis Review

Start at slide 21 of Basic Analysis 3

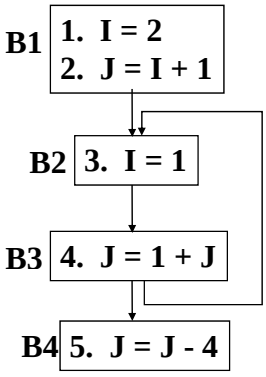
Static Single Assignment

Static Single Assignment (SSA)



What is static single assignment?

Static Single Assignment (SSA)

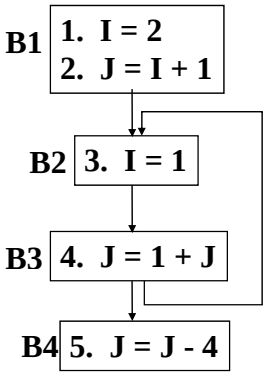


Static Single Assignment is an intermediate representation in which every variable assigned a value occurs as the target of only one assignment

Makes some optimizations more effective

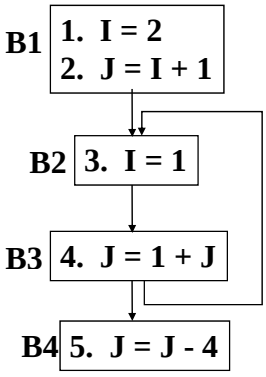
Suggested as IR for pointer analysis but no empirical results show that it is effective

Static Single Assignment (SSA)

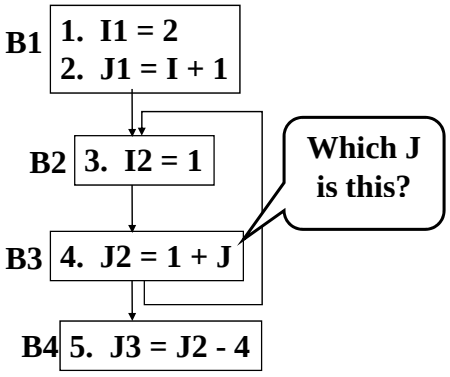


Translate the program to SSA form

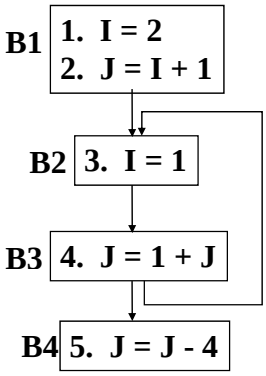
Static Single Assignment (SSA)



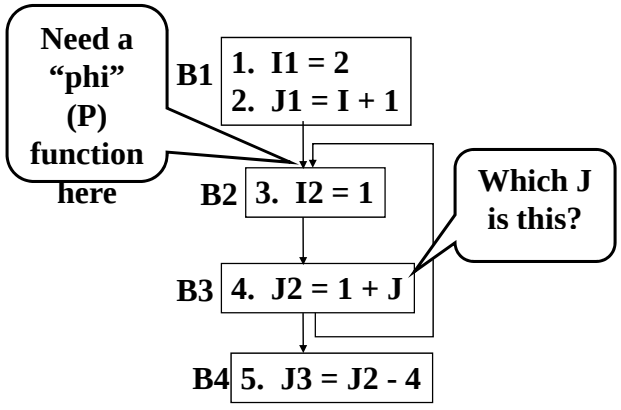
Translate the program to SSA form



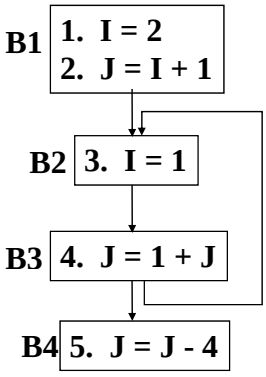
Static Single Assignment (SSA)



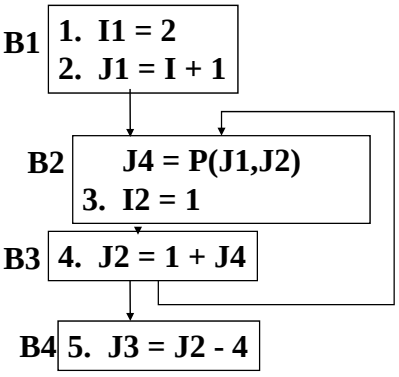
Translate the program to SSA form



Static Single Assignment (SSA)

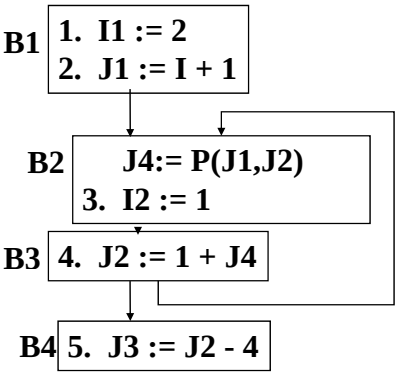


Translate the program to SSA form



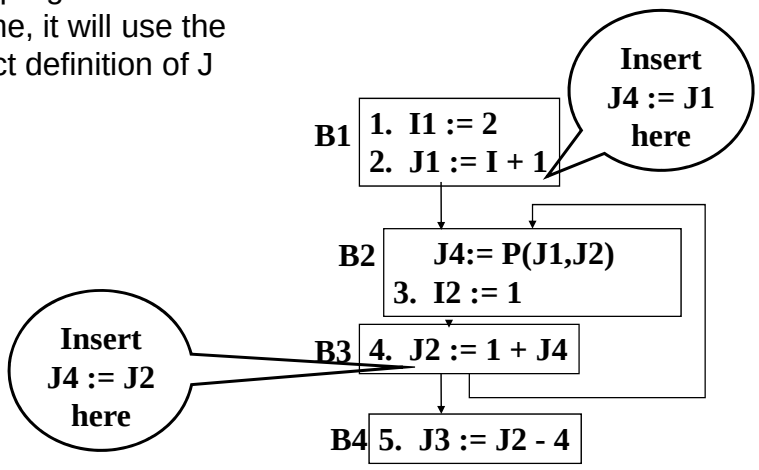
Static Single Assignment (SSA)

Rewrite program so that at runtime, it will use the correct definition of J

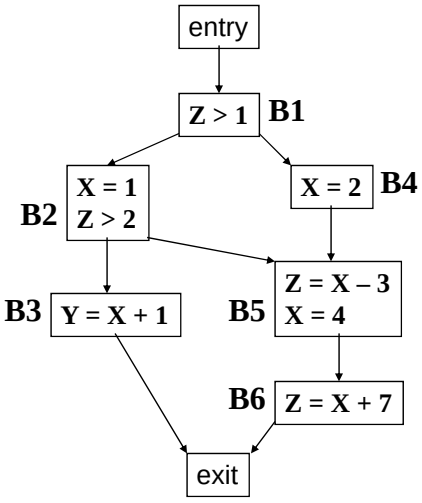


Static Single Assignment (SSA)

Rewrite program so that at runtime, it will use the correct definition of J



Static Single Assignment (SSA)

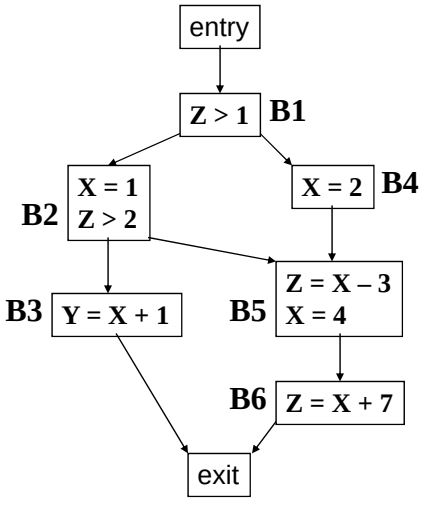


What if program is more complicated, how can we determine where to place "phi" functions?

Can use dominance frontiers

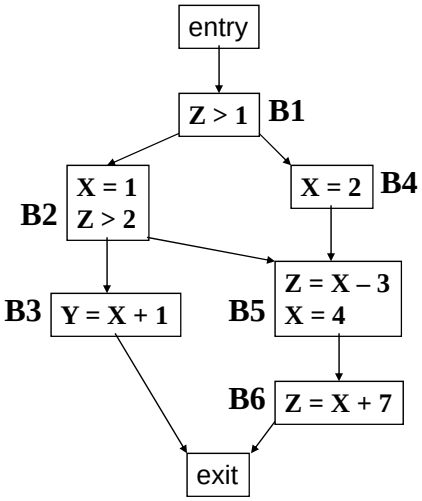
Dominance Frontiers

Dominance Frontiers



For a CFG, the dominance frontier of x, written DF(x), is the set of all nodes y in the CFG such that x dominates an immediate predecessor of y but does not strictly dominate y (what does it mean to strictly dominate?)

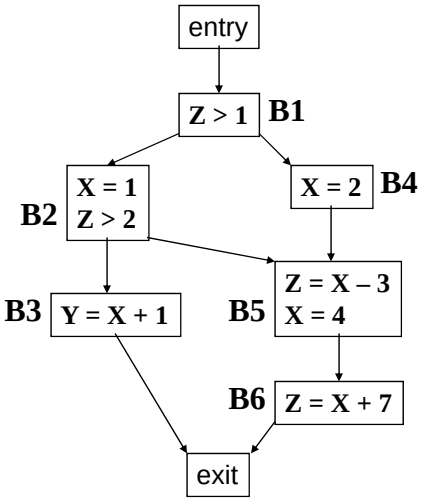
Dominance Frontiers



What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

First, get dominator tree

Dominance Frontiers

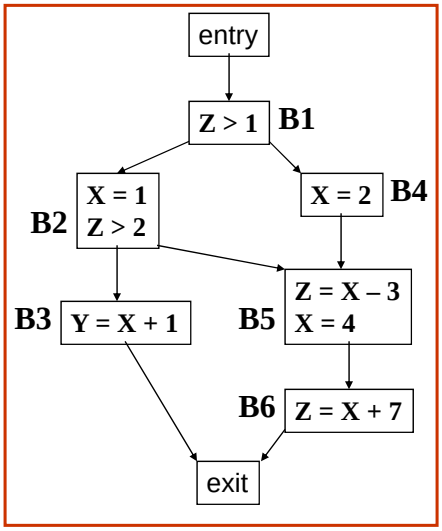


What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

entry:

B1:

Dominance Frontiers

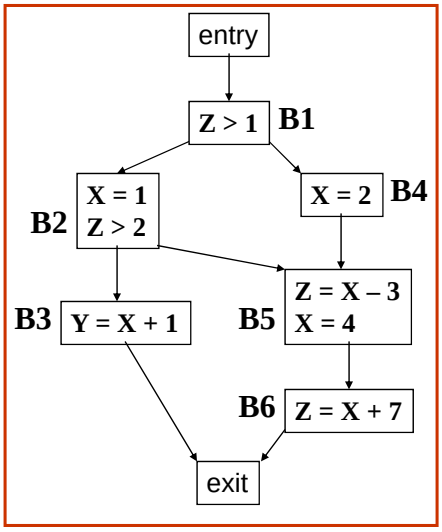


What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

entry:

B1:

Dominance Frontiers



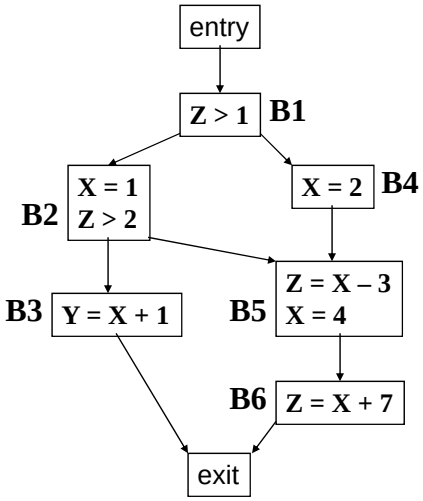
What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

entry:

entry dominates all nodes
→DF(entry) is empty

B1:

Dominance Frontiers



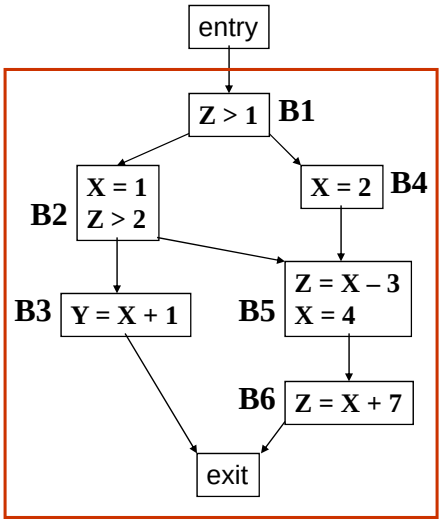
What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

entry:

entry dominates all nodes
→DF(entry) is empty

B1:

Dominance Frontiers



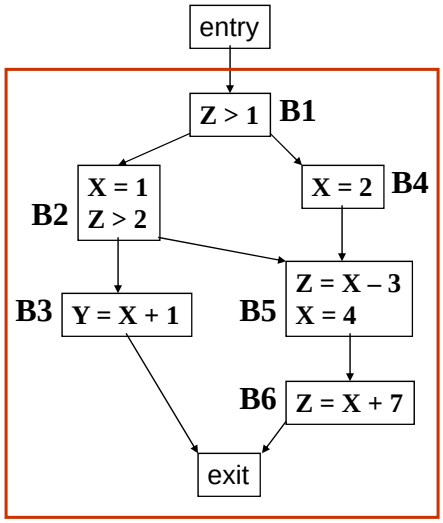
What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

entry:

entry dominates all nodes
→DF(entry) is empty

B1:

Dominance Frontiers



What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

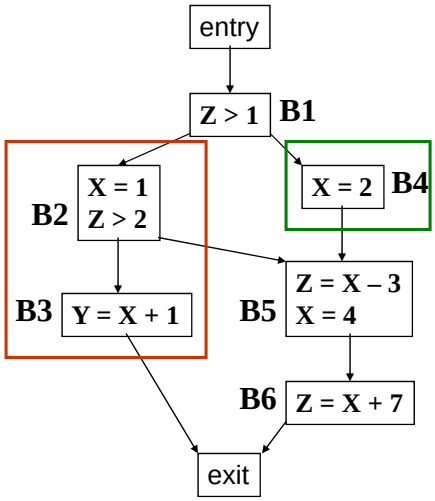
entry:

entry dominates all nodes
→DF(entry) is empty

B1:

B1 dominates B1…exit → DF(B1) is empty

Dominance Frontiers



What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

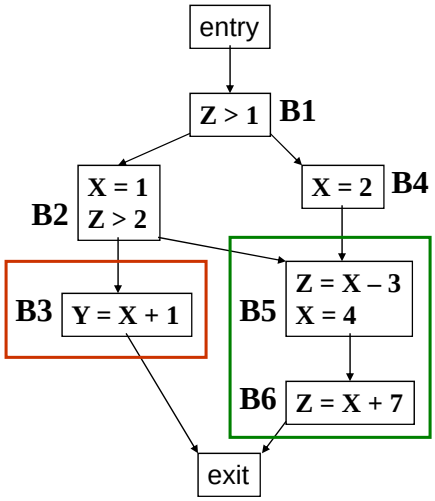
B2:

B2 dominates B2, B3, successor nodes outside this region are B5, exit → DF(B2) is {B5, exit}

B4:

B4 dominates B4, successor node outside this region is B5
→ DF(B4) is {B5}

Dominance Frontiers



What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

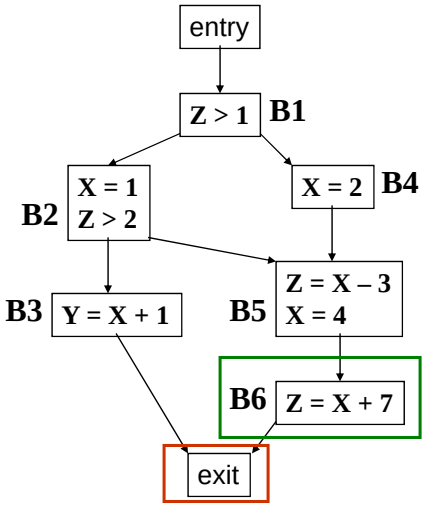
B3:

B3 dominates B3, successor node outside this region is exit → DF(B3) is {exit}

B5:

B5 dominates B5, B6, successor node outside this region is exit → DF(B5) is {exit}

Dominance Frontiers



What are the dominance frontiers for basic blocks entry, 1, …, 6, exit?

exit:

exit dominates exit, no successor node outside this region → DF(exit) is empty

B6:

B6 dominates B6, successor node outside this region is exit → DF(B6) is {exit}

Dominance Frontiers

```
algorithm Dom_Front
Input: Dom, dominator tree for CFG
output  DF[X] for each node X in CFG
Method:
    foreach X in a bottom-up traversal of the Dom do
        DF[X] = empty
        foreach Y in Succ(X) do       // Succ() is immediate successor
          if idom(Y) != X then DF(X) = DF(X) union {Y} endif      // local
        endfor
        foreach Z in Children(X) do
          foreach Y in DF(Z) do
              if idom(Y) != X then DF(X) = DF(X) union {Y} endif       // up
          endfor
    end Dom_Front
```

Computing SSA

So far, we know

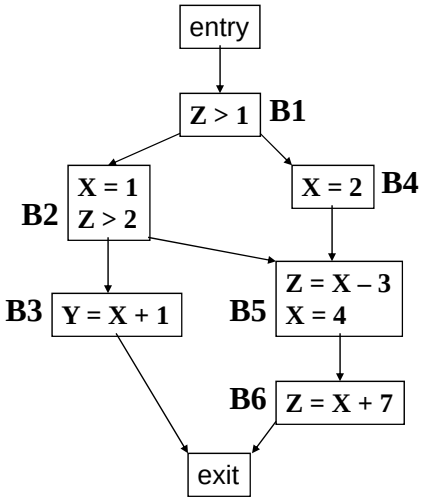
- What SSA is
- How to remove phi functions
- That there's a way to optimize placement of these phi functions using dominance frontiers

Still need to know where to place the phi functions

- Will use iterated dominance frontiers (IDF)
- $IDF(X) = DF^{+}(X) = \lim DF^{i}(X)$ where

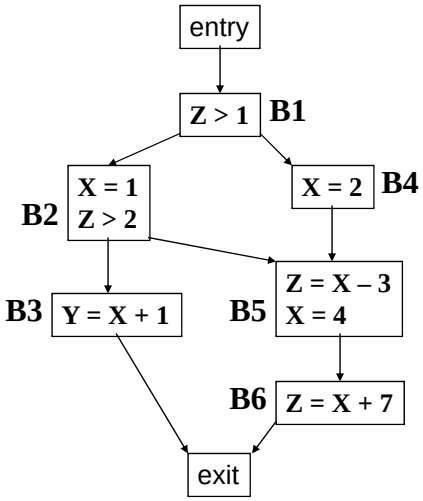
$$DF^{i+1} = DF(X \cup DF^{i}(X))$$

Iterated Dominance Frontiers



What are the iterated dominance frontiers for each variable:

Iterated Dominance Frontiers



What are the iterated dominance frontiers for each variable:

For x:

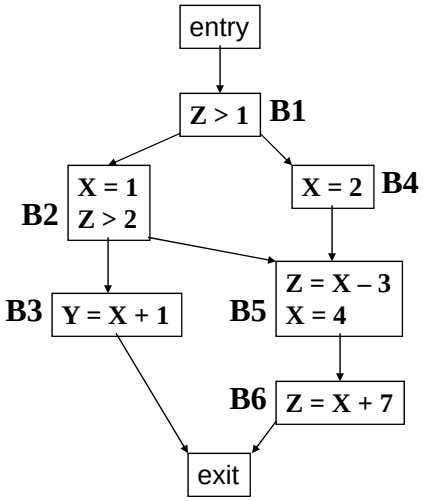
$DF^1(2,4,5)=\{5,exit\}$

$DF^2(2,4,5)=DF(2,4,5 \cup DF^1(2,4,5))$

$= DF(2,4,5,exit)$

$= \{5, exit\}$

Iterated Dominance Frontiers



What are the iterated dominance frontiers for each variable:

For x: {5,exit}

For y:

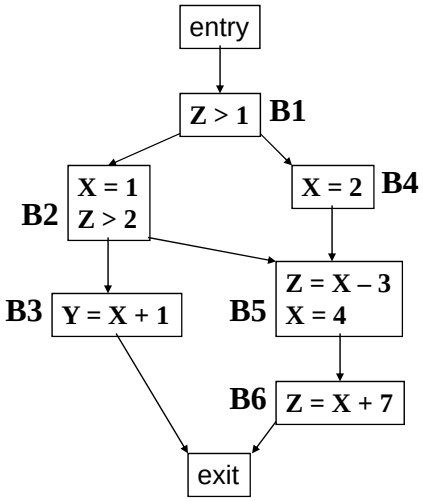
$DF^1(3)=\{exit\}$

$DF^2(3)=DF(3U\ DF^1(3))$

$= DF(3,exit)$

$= \{exit\}$

Iterated Dominance Frontiers



What are the iterated dominance frontiers for each variable:

For x: {5,exit}

For y: {exit}

For z:

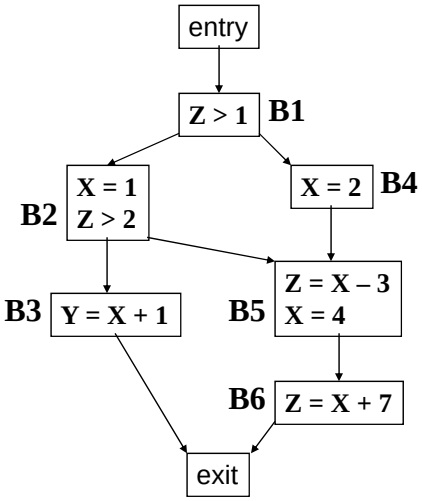
$DF^1(5,6)=\{exit\}$

$DF^2(5,6)=DF(5,6U\ DF^1(5,6))$

$= DF(5,6,exit)$

$= \{exit\}$

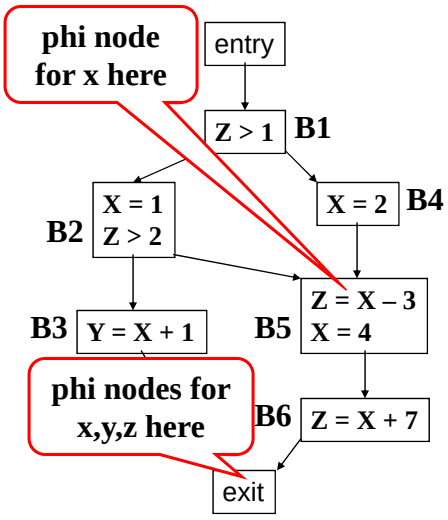
Iterated Dominance Frontiers



What are the iterated dominance frontiers for each variable:

For x: {5,exit}
For y: {exit}
For z: {exit}

Iterated Dominance Frontiers



What are the iterated dominance frontiers for each variable:

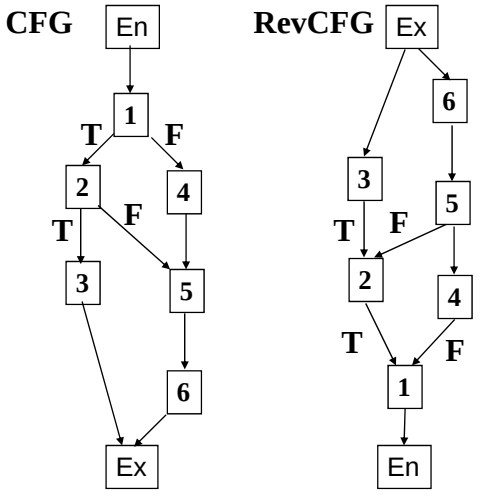
For x: {5,exit}
For y: {exit}
For z: {exit}

Dominance Frontiers & Control Dependence

Control dependence can be computed

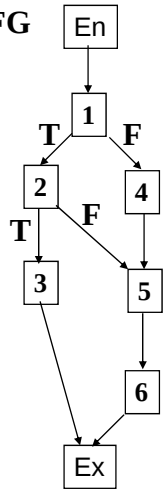
- Ferrante, Ottenstein, and Warren algorithm—last Thursday
- Using dominance frontiers
  - Control dependencies are dominance frontiers on the reverse control-flow graph

Example

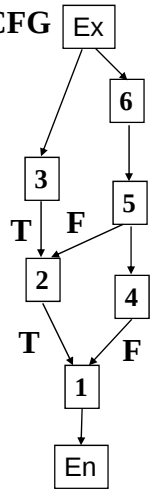


Example

CFG



RevCFG



DF(En) = {}
DF(1) = {}
DF(2) = {1} 1T
DF(3) = {2} 2T
DF(4) = {1} 1F
DF(5) = {1,2} 1F, 2F
DF(6) = {1,2} 1F,2F
DF(Ex) = {}