

Control dependence

#secure]

fn secure() $\rightarrow$ int {0}

fn main() {

println(secure())

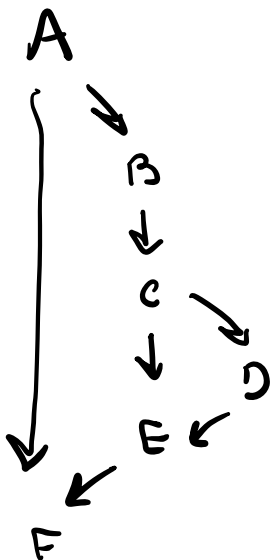
if secure() == 1 {

println("hello world")

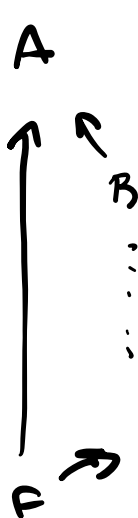
}
println("forber")

}

CFG:



Rev CFG:



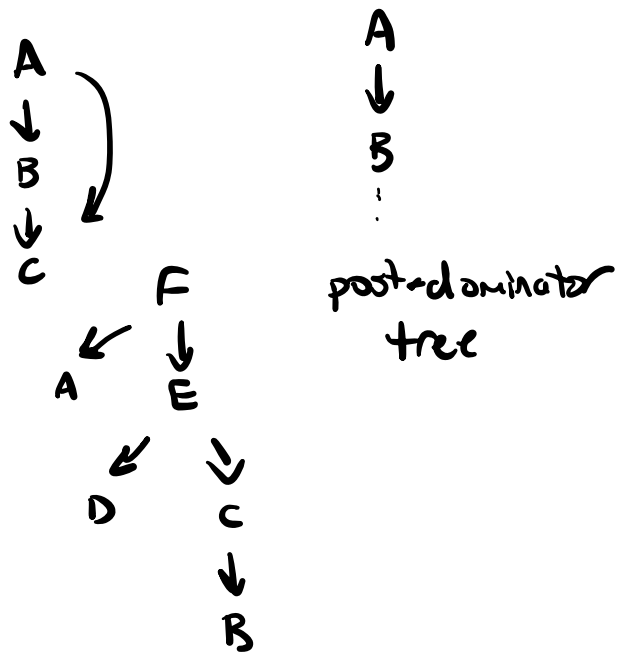
X is control-dependant (Cdep) on Y

if X post-dominates a child of Y,
but not Y

$\approx$ X post-dominates some, but not
all children of Y

X post-dominates Y if

X appears on all paths from Y
to the exit



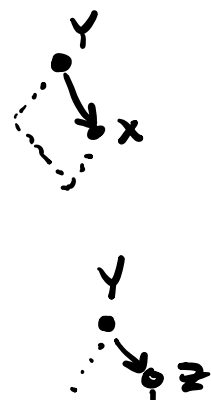
goal: Find nodes X Y s.t.
X pdom a child of Y but
not Y

for $x \in \text{rpo}(\text{pdom tree})$:

① for all edges $Y \rightarrow X$:

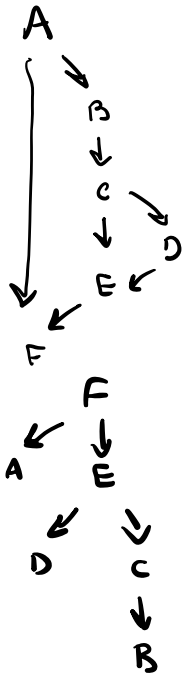
if X does not pdom Y then X cdep Y

② for each node Z immediately pdom by X:



for each node Y s.t. $Z \text{ cdep } Y$:

if X does not point Y , then $X \text{ cdep } Y$



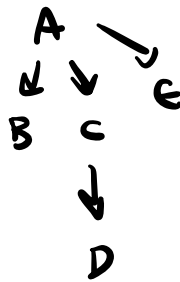
$B \text{ cdep } A$

$C \text{ cdep } A$

$D \text{ cdep } C$

$E \text{ cdep } A$

$F \dots$



control-dependence graph



Interprocedural Flow Analysis / Taint tracking

$\sigma \in \{ \text{Var} \}$

fn main() {

let $x = \text{scan}()$ in // $\sigma = \{x\}$

let $y = x$ in // $\sigma = \{x, y\}$

}

Dealing with pointers & control deps:

Mem Loc $l ::= x \mid A.i$

tainted(σ, x) = $\sigma \cap x \neq \emptyset$

Allocation A

Place $p ::= x(.i)^*$

$f(x = rv, \sigma) =$

if tainted(σ , uses(rv)):

add x to σ

↑ old analysis

$f(p_{dst} = rv, \sigma, i_{loc})$:

explicit = $\bigcup_{p_{src} \in \text{uses}(rv)} \text{aliases}(p_{src})$

implicit = $\bigcup_{j \text{ s.t. } i \text{ cdep } j} \bigcup_{p_{src} \in \text{uses}(P[j])} \text{aliases}(p_{src})$

new analysis
↓

$uas(if\ p\ goto\ i_1\ else\ i_2)$
 $= \{p\}$

if tainted (σ , implicit $\cup$ explicit):

for $last \in aliases(p_{last})$:

add $last$ to σ

Abstract allocations for interprocedural flows thru pointers:

$ctx: \{y\}$

$output: \{ (arg\ x.\emptyset, y) \}$

$fn\ set_val(x: (int), y: int) \{$

$\quad x.\emptyset = y$

$\}$

$fn\ set_ptr(x: ((int), \emptyset), y: (int, \emptyset)) \{$

$\quad x.\emptyset = y$

$\}$

$\theta[x] = \{arg\ x.\emptyset\}$
 $\quad \quad \quad \uparrow$
 $\quad \quad \quad \begin{matrix} arg\ x & arg\ y \\ arg\ x.\emptyset & \end{matrix}$

$f(x = f(y^*), \sigma)$

$\sigma^{start} = \{ f.params[i] \mapsto \sigma[y_i] \}$

$results = DATAFLOW(f, \sigma^{start})$

$output = \bigsqcup_{j \in returns(f)} results[j][ret]$

$\sigma[x \mapsto output]$

$fn\ main() \{$

$\quad let\ a = (5, \emptyset)\ in$

$\quad set_val(a, secure());$

$\quad let\ b = (5, \emptyset)\ in$

$\quad let\ c = (b, \emptyset)\ in$

$\quad set_ptr(c, a);$

$\quad println(c.\emptyset.\emptyset)$

Mem Loc $L ::= x \mid A.i \mid ret$

Allocation $A_i ::= local\ i$
 $\quad \quad \quad \mid arg\ p$