

Creating an entry store
for a bb counts as
changing it.

entry : \$jump bb2
bb2 : $x = \$copy 2$
~~\$ret~~

bb2 store

entry : $x = \$copy 2$
~~\$branch x bb1 bb2~~

reg zero pos

$x \mapsto \text{pos}$
 $y \mapsto \text{neg}$
 $z \mapsto \text{zero}$

\sqcup

$x \mapsto \text{reg}$
 $y \mapsto \text{neg}$

$x \mapsto T$
 $y \mapsto \text{neg}$
 $z \mapsto \text{zero}$

abstract semantics

'global-ints'

prep :

- compute set of int-typed global variables
- compute set of addr-taken int-typed local variables

L needs to include global variables ← 'addr-of-ints'

Γ needs to include global variables $\leftarrow \text{'addrof_ints'}$

non-ptr, non-call related instructions



- $x = \$\text{copy } op$ $\leftarrow \text{'operand' = variable or a constant}$

\hookrightarrow if x is not 'int', ignore

\hookrightarrow translate op to abstract value
using Δ or Γ

$$\hookrightarrow \Gamma' = \Gamma[x \mapsto \text{abs}(op)]$$

$$\Gamma$$

- $x = \$\text{arith } \langle op \rangle op1 op2$

$$\hookrightarrow \Gamma' = \Gamma[x \mapsto \text{abs}(op1) \stackrel{\wedge}{\text{op}} \text{abs}(op2)]$$

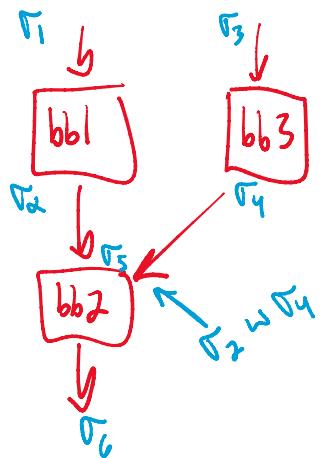
- $x = \$\text{cmp } \langle op \rangle op1 op2$

\hookrightarrow if $op1, op2$ are not int, $\Gamma' = \Gamma[x \mapsto T]$

- $\$jump bb$

- $\$branch op bb1 bb2$

\hookrightarrow propagate to $bb1$ and/or $bb2$
depending on abstract value of op



ptr-related instructions

$$a = \$\text{copy } 2$$

$$b = \$\text{copy } 3$$

ptr-related instructions

- $x = \$load y$

↳ if x not an int, ignore

↳ $\Gamma' = \Gamma[x \mapsto T]$

- $\$store x op$

↳ if op is not int, ignore

↳ $\forall var \in \text{address_ints}, \Gamma' = \Gamma' \cup \Gamma \cup [var \mapsto \text{abs}(op)]$

- $x = \$alloc op [id]$

$x = \$addr y$

$x = \$ gep y op$

$x = \$ gfp y field$

} all have non-int x ,
so ignore

call-related instructions

- $[x =] \$call-ext id (op1, ...)$

- $[x =] \$call-dir id (op1, ...)$

- $[x =] \$call-idr fp (op1, ...)$

↳ $\forall var \in \text{global_ints}$, update Γ to map $var \mapsto T$

↳ if x is int, set $x \mapsto T$

| : if new argument is a ptr that can reach

$$\begin{aligned} a &= \$copy 2 \\ b &= \$copy 3 \\ x &= \$addrof a \\ y &= \$addrof L \end{aligned}$$

⋮
 $\$store x - 4 | 2$

$\Gamma' = \Gamma \cup [a \mapsto ny]$

$\Gamma'' = \Gamma' \cup [b \mapsto ny]$

— . . .

L if any argument is a ptr that can reach
an int (eg. `&&int`), then $\forall \text{var} \in \text{argsOfInts}$,
set $\text{var} \mapsto T$

- \$ ret op[?]
ignore