Rvalue rV ::= ... | $alloc^A(p^*)$

Allocation A

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

Memloc ℓ = $x \mid A.i \mid A[-]$

aliases : Place → {Memloc}

Θ : Memloc → {Allocation}

$x = alloc^A(0)$
$x = alloc^B(1)$ example

aliases(x) = {x}

aliases(x.0) = {A.0, B.0}
$\Theta(x)$ = {A, B}

$f(x = 0, \sigma) = \sigma[x \mapsto Z]$

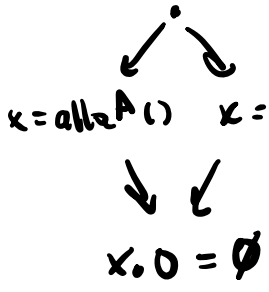
$f(x = y, \sigma) = \sigma[x \mapsto \sigma[y]]$

$\sigma \in \{Var \mapsto \alpha\}$

$\alpha = \{\bot, T, Z, NZ\}$

$\sigma \in \{Memloc \mapsto \alpha\}$

$f(p = 0, \sigma) =$
if |aliases(p)| = 1 :
  $\sigma[aliases(p)[0]] = Z$
else:
  for $\ell \in$ aliases(p):
    $\sigma = \sigma[\ell \mapsto \sigma[\ell] \sqcup Z]$



$f(p_{dst} = p_{src}, \sigma) =$

for $\ell_{dst} \in$ aliases(p_{dst}):
  for ℓ_{src} in aliases(p_{src}):
    $\sigma[\ell_{dst}] \sqcup = \sigma[\ell_{src}]$

Rvalue rV ::= ... | $f(x^*)$

Function F ::= fn f(x^*) { I^* }

Instr I ::= ... | return x

```
fn main() {
  x = 0
  y = id(x)
  z = 1/y
}
```

```
fn id(a) {
  return a
}
```

FF

$f(x = f(y_1, \ldots y_n), \sigma) =$

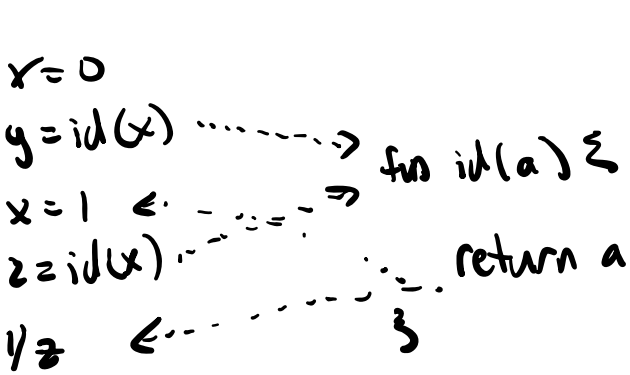
$\sigma[x \mapsto T]$ Approximation

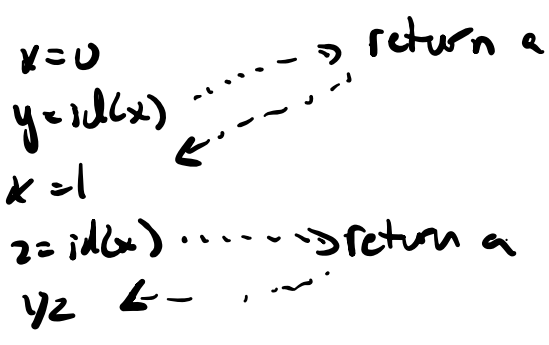
Context-sensitivity

$\sigma_{start} = \{param(i) \mapsto \sigma[y_i]\}$

result = Analyze(F, σ_{start})

postprocess(σ, result)






```
x = VecLight(&y, 0);
```

```
get<'a>(v: &'a VecCT>,
        i: usize) -> &'a T
```