Lecture #6 Data flow.

- $C(x, S, in)$ = value of x before S
- $C(x, S, out)$ = value of x after S

Rule 1: if $C(x, P_i, out) = *$ for any i
then $C(x, s, in) = *$

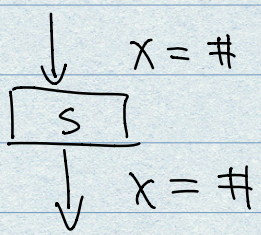
Rule 2: $C(x, P_i, out) = c$ &
$C(x, P_j, out) = d$ &
$d \neq c$
then $C(x, S, in) = *$

Rule 3: if $\forall i, C(x, P_i, out) = c$ or $\#$
then $C(x, S, in) = c$

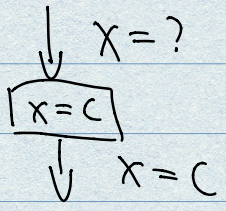
Rule 4: if $\forall i$, $C(x, P_i, out) = \#$
then $C(x, S, in) = \#$

↑ $C(x, P_i, out) \longrightarrow C(x, S, in)$

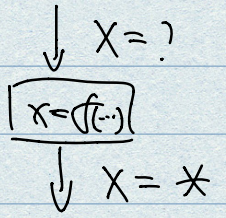
Rule 5.



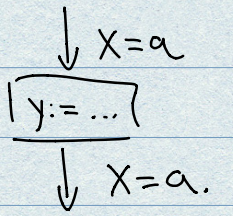
Rule 6.



Rule 7.



Rule 8.



$X \neq Y$.

An Algorithm:

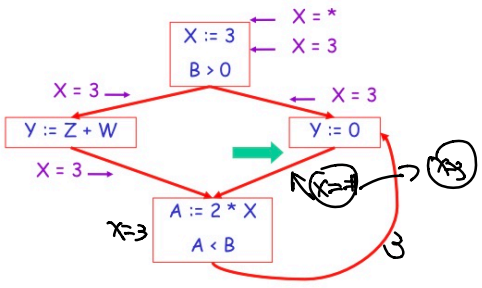
1. for every entry s to the program
   set $C(s, x, in) = *$

2. Set $C(s, x, in)$
   $= C(s, x, out)$
   $= \#$
   everywhere else

3. Repeat until all points satisfy 1-8.

The Value

- To understand why we need #, look at a loop



Dynamic dispatch : get type

→ get method / property.

Ordering.

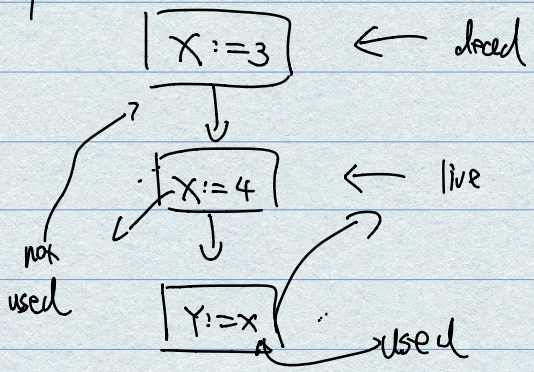
* is the greatest, # is the least.

the lub be the least-upper-bound.

Rule 1 - 4 →

$$C(x, s, in) = lub\{ C(p, x, out) \mid p \text{ is a predecessor of } s \}$$

Liveness Analysis.



live merged with live

→ live.