Section 9:

Optimizations Pt. 1

CS 164 @ UC Berkeley, Spring 2024

Reminders

WA 5 is due on Thursday, April 11 at 11:59 PM PST.

Pre-PA4 is released!

- Due on April 16 at 11:59 PM PST.

  (hard deadline)

Reminder to take care of yourselves, and to prioritize your health! WAs are worth 5% of your grade so don't stress too much about them!

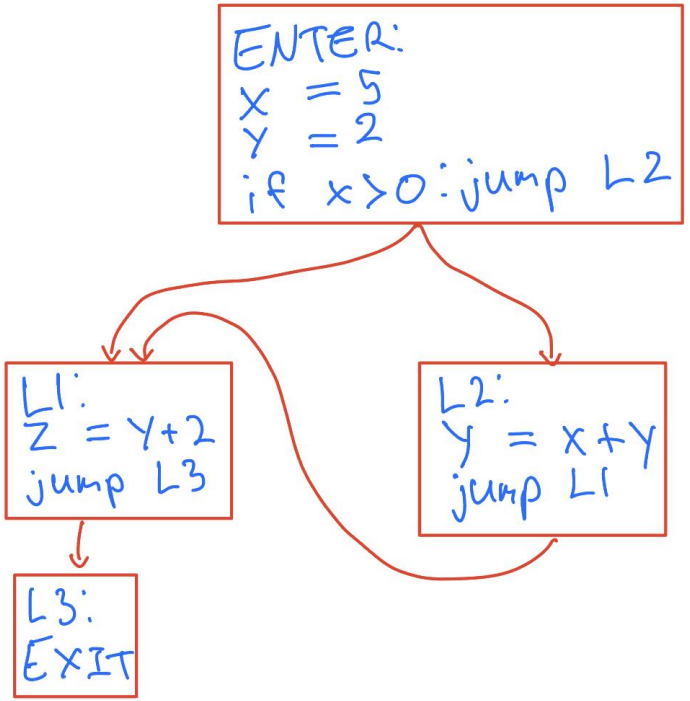
Control Flow Graphs

Directed graph of basic blocks

- Edges represent possible control flow
- Only first instruction can be a label
- Only last instruction can be a jump

Control Flow Graphs

```
ENTER:
    x = 5
    y = 2
    if x>0: jump L2
L1:
    z = y+2
    jump L3
L2:
    y = x+y
    jump L1
L3:
    EXIT
```



Dead Code Elimination

Are variables referenced after being assigned?

```
a := x ** 2
b := 3
c := x
d := c * c
e := c * c
f := a + d
g := e * f
```

```
a := x ** 2
~~b := 3~~
c := x
d := c * c
e := c * c
f := a + d
g := e * f
```

Common Subexpression Elimination

Do any assignments have the same right-hand side?

```
a := x ** 2
b := 3
c := x
d := c * c
e := c * c
f := a + d
g := e * f
```

```
a := x ** 2
b := 3
c := x
d := c * c
e := d
f := a + d
g := e * f
```

Copy Propagation

Do any variables copy other variables?

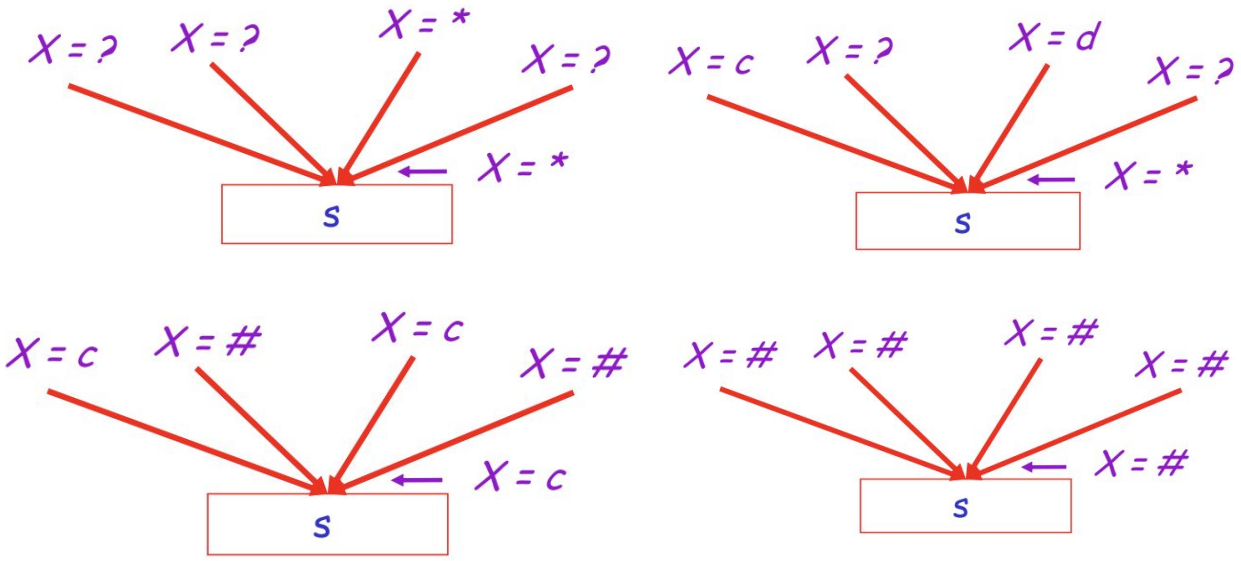
```
a := x ** 2
b := 3
c := x
d := c * c
e := c * c
f := a + d
g := e * f
```

```
a := x ** 2
b := 3
c := x
d := x * x
e := x * x
f := a + d
g := e * f
```

Flow Analysis for Constant Propagation

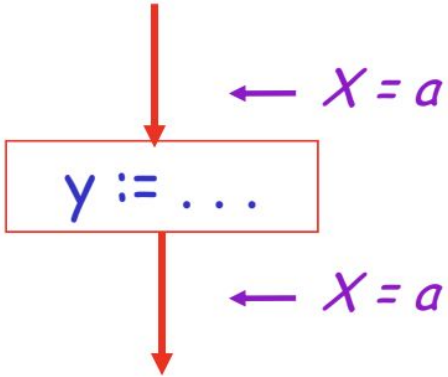
- Global optimizations require flow analysis
- Program points:
  - For each statement in a program, define a program point before and after it.
- Constant propagation:
  - If a variable is assigned a constant, replace every valid reference with the constant.
  - For each variable x, for each program point, assign value #, *, or a constant.
  - #: Program point is unreachable from where x is assigned.
  - *: Don't know if x is a constant at this program point.
  - Constant: x has a constant value at this program point.
- #, *, and the constants form a lattice where # < c < *.
  - Every finite set of values has a *least upper bound*.
  - Provide guarantees for termination of flow analysis.

Flow Analysis for Constant Propagation



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

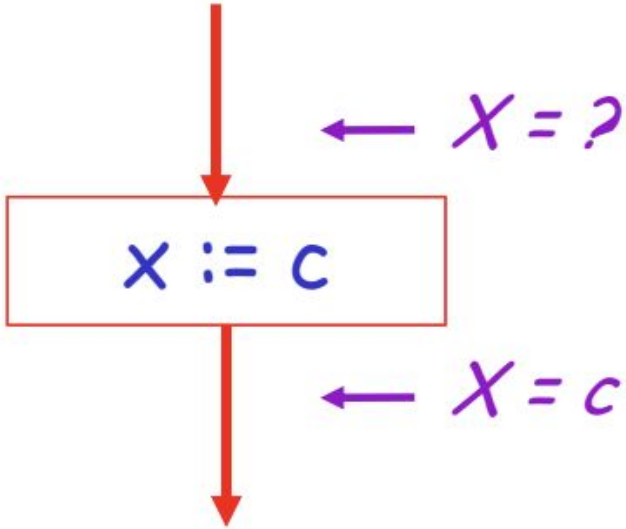
Flow Analysis for Constant Propagation



$$C_{out}(x, y := \ldots) = C_{in}(x, y := \ldots) \text{ if } x \neq y$$

Flow Analysis for Constant Propagation

(Picture slightly different from rule)



$$C_{out}(x, x := e) = eval(e, C_{in})$$

Anonymous feedback form:
http://tinyurl.com/SoraDisFeedback