Constant Propagation
overview of algorithm

• Build control flow graph

• Perform symbolic evaluation
  • Keep track of whether variables are constant or not

• Replace constant-valued variable uses with their values, try to simplify expressions and control flow
overview of algorithm

• Build control flow graph

• Perform symbolic evaluation
  • Keep track of whether variables are constant or not

• Replace constant-valued variable uses with their values, try to simplify expressions and control flow
build control flow graph

x = 1;
y = read()
if (y > x)
    y = 5;
    x = y + 1;
else
    y = 3;
    x = y + 3;
if (x < 6)
    print(y);

When evaluating a piece of code, we care about individual statements, not basic blocks.

- Need to know the value of variables right before a statement executes to figure out what the statement does.

Create a new version of the CFG with one node per statement instead of per basic block.
- Also helpful to explicitly mark where control flow paths merge.
x = 1;
y = read();
if (y > x)
    y = 5;
    x = y + 1;
else
    y = 3;
    x = y + 3;
if (x < 6)
    print(y);
When we concretely execute a CFG, we are executing a program.

Keep track of values of variables before each statement.

Execute statement to determine values after the statement executes.

Evaluate conditionals to choose which path to take.

```
x = 1
y = read()
(y > x) ?
1  x = 1
2  y = read()
3  (y > x) ?
4  y = 5
5  x = y + 1
6  y = 3
7  x = y + 3
(x < 6) ?
8  (x < 6) ?
9  print(y)
print(y)
merge
merge
merge
halt
```
executing a cfg

• When we concretely execute a CFG, we are executing a program

• Keep track of values of variables before each statement

• Execute statement to determine values after the statement executes

• Evaluate conditionals to choose which path to take

```
1  x = 1
2  y = read()
3  (y > x) ?

4  y = 5
5  x = y + 1

6  y = 3
7  x = y + 3

* merge
8  (x < 6) ?

9  print(y)

* merge
* merge

10  halt
```
executing a cfg

- When we concretely execute a CFG, we are executing a program
- Keep track of values of variables before each statement
- Execute statement to determine values after the statement executes
- Evaluate conditionals to choose which path to take
executing a cfg

- No matter what line 2 does, x always has the value 6 at line 8
- print statement never executes
- How can we figure this out?
overview of algorithm

• Build control flow graph

• Perform symbolic evaluation
  • Keep track of whether variables are constant or not

• Replace constant-valued variable uses with their values, try to simplify expressions and control flow
next: symbolic evaluation