Lecture 6: Data-Flow Analysis Termination and Complexity of the Worklist Algorithm

17-355/17-665/17-819: Program Analysis

Rohan Padhye

February 3, 2022

* Course materials developed with Jonathan Aldrich and Claire Le Goues

Worklist Algorithm [Kildall'73]

```
worklist = ∅
for  Node n in cfg
     input[n] = output[n] = ⊥
     add n to worklist
input[0] = initialDataflowInformation

while worklist is not empty
     take a Node n off the worklist
     output[n] = flow(n, input[n])
     for Node j in succs(n)
           newInput = input[j] ⊔ output[n]
           if newInput ≠ input[j]
                 input[j] = newInput
                 add j to worklist
```

Worklist Algorithm [Kam & Ullman'76]

```
worklist = ∅
for Node n in cfg
    input[n] = output[n] = ⊥
    add n to worklist
output[programStart] = initialDataflowInformation

while worklist is not empty
    take a Node n off the worklist
    input[n] = ⊔_{k∈preds(n)} output[k]
    newOutput = flow(n, input[n])
    if newOutput ≠ output[n]
        output[n] = newOutput
        for Node j in succs(n)
            add j to worklist
```

Recall: Fixed point of Flow Functions

$$(\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_n) \xrightarrow{f_Z} (\sigma'_0, \sigma'_1, \sigma'_2, \dots, \sigma'_n)$$

$$\sigma'_0 = \sigma_0$$

$$\sigma'_1 = f_Z[\![x := 10]\!](\sigma_0)$$

$$\sigma'_2 = f_Z[\![y := 0]\!](\sigma_1)$$

$$\sigma'_3 = \sigma_2 \sqcup \sigma_7$$

$$\sigma'_4 = f_Z[\![\text{if } x = 10 \text{ goto } 7]\!]_F(\sigma_3)$$

$$\vdots$$

$$\sigma'_8 = f_Z[\![\text{if } x = 10 \text{ goto } 7]\!]_T(\sigma_3)$$

$$\sigma'_9 = f_Z[\![x := y]\!](\sigma_8)$$

Fixed point!

$$(\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_n) = f_Z(\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_n)$$

Correctness theorem:
If data-flow analysis is well designed*, then any fixed point of the analysis is sound.

* Lattice has finite height and flow functions are monotonic.

Successive applications for the whole-program flow function results in an ascending chain

Base case: $(\sigma_0, \bot, \bot, \dots, \bot) \xrightarrow{f_Z} (\sigma'_0, \sigma'_1, \sigma'_2, \dots, \sigma'_n)$

Inductive case: $(\sigma_0, \sigma_1, \sigma_2, \dots, \sigma_n) \xrightarrow{f_Z} (\sigma'_0, \sigma'_1, \sigma'_2, \dots, \sigma'_n)$