Lecture 20

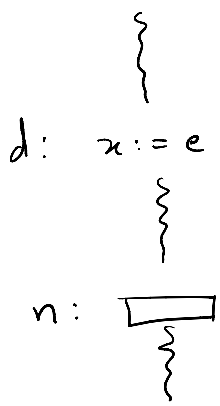
Monday, April 6, 2020 4:19 PM

- Hello!
- Use the shared Google Doc as always
- Link to Friday office hours:

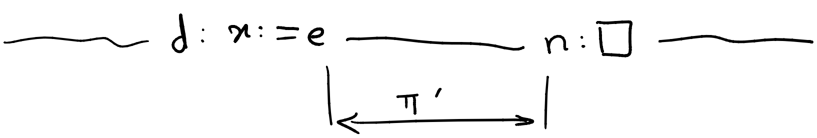
  https://usc.zoom.us/j/928467677

  (Or send me email)

- Dataflow analysis
- Reaching Definitions



The definition d <u>reaches</u> n if there is some program execution π



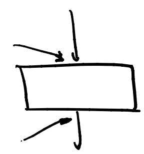
s.t. x is never reassigned in the subpath π'.

d: x := source()
y := x
y := 5
n: $\boxed{z := x}$
	d: x := source()
	x := 5
	n: $\boxed{z := x}$
		y := input()
		d: x := source()
		if (y > 100) {
		x := 3
		} if (y ≤ 100) {
		x := 7
		}
		n: $\boxed{z := x}$

There is no feasible path from d to n.

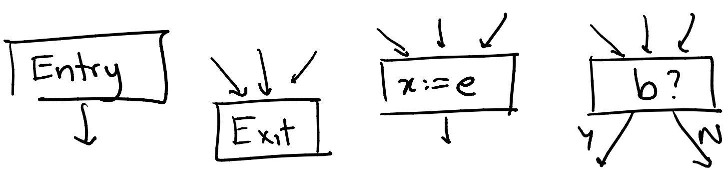
RD[n] = set of all definitions d that can reach n.

IN[n] = set of all definitions that can reach the beginning of n (= RD[n])

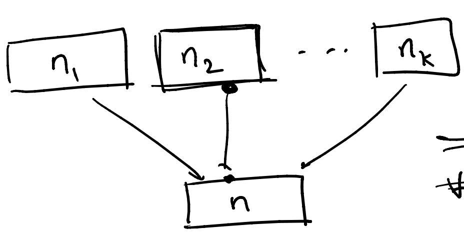


OUT[n] = set of all definitions that can reach the end of n.

Control flow graphs:



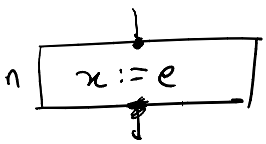
Claim: ① OUT[Entry] = ∅

② 

$IN[n] = \bigcup_i OUT[n_i]$

$\cancel{\forall i.\ OUT[n_i] \subseteq IN[n]}$

(3)



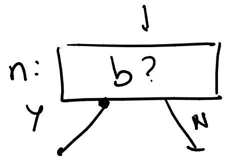
$$OUT[n] = (IN[n] \setminus KILL[n]) \cup GEN[n]$$

Every dataflow fact involving x $\rightarrow$ $KILL[n] = \{ d \mid d : x := e' \}$

Every assignment statement in the program with x on the left

The statement $x := e$ $\rightarrow$ $GEN[n] = \{ n \}$

(4)



$$OUT[n] = (IN[n] \setminus KILL[n]) \cup GEN[n]$$

Conditional statements don't kill any facts; don't give rise to any dataflow facts. $\rightarrow$ $\begin{cases} KILL[n] = \emptyset \\ GEN[n] = \emptyset \end{cases}$

Algorithm Compute-RD (Chaotic Iteration Algorithm)

(1) For each statement n,

initialize $IN[n], OUT[n] := \emptyset$.

(2) Until fixpoint: (Iter # k)

For each non-entry stmt n:

- Update $IN^{(k)}[n] := \bigcup_i OUT^{(k-1)}[n_i]$

i ← ranging over all predecessors

- Update $OUT^{(k)}[n] := (IN^{(k)}[n] \setminus KILL[n]) \cup GEN[n]$

Specified as part of the input; Computed at the beginning of time.

<u>Claim</u> 5 When Algorithm - Compute RD terminates claims 1 — 4 are all satisfied.

6 After the $(k+1)$-th iteration, $\forall n$

$$IN^{(k)}[n] \subseteq IN^{(k+1)}[n] \quad OUT^{(k)}[n] \subseteq OUT^{(k+1)}[n].$$

7 Algorithm - Compute RD terminates.

<u>Ex</u>: Algorithm - Shortest Paths $(G = (V, E))$

1 For each pair of vertices $(u\ v)$ initialize $sp(u\ v) := \underline{\infty}$.

2 For each vertex v, update $sp(v\ v) := 0$.

3 Until fixpoint:

- For each edge $e = (u\ v)$ update: $sp(u\ v) := \min(sp(u\ v)\ \ w_e)$

update

- For each triple of vertices (u v w)

  update sp(u w) := min(sp(u w), sp(u v) + sp(v w))

<u>Termination argument</u>

1. Define ranking fn.
2. Show that the ranking fn monotonically decreases

- e: Entry
- d: $x :=$ source()
- a: $y := x$
- b: $y := 3$
- n: $z := x$

	Initially	After iter 1	After iter 2	After iter 3	After iter 4	After iter 5	After iter 6	After iter 7	After iter 8
OUT[e]	∅	∅	∅	∅	∅	∅	∅	∅	∅
IN[d]	∅	∅	∅	∅	∅	∅	∅	∅	∅
OUT[d]	∅ ⊆	{d} ⊆	{d} ⊆	{d}	{d}	{d}	{d}	{d}	{d}
IN[a]	∅	∅ ⊆	{d}	{d}	{d}	{d}	{d}	{d}	{d}
OUT[a]	∅	{a} ⊆	{a}	{a,d}	{a,d}	{a,d}	{a d}	{a d}	{a d}
IN[b]	∅	∅	{a}	{a}	{a,d}	{a,d}	{a d}	{a d}	{a d}
OUT[b]	∅	{b}	{b}	{b}	{b}	{b,d}	{b d}	{b d}	{b d}
IN[n]	∅	∅	{b}	{b}	{b}	{b}	{b d}	{b d}	{b d}
OUT[n]	∅	{n}	{n}	{b,n}	{b n}	{b n}	{b n}	{b n d}	{b n d}

Expressing dataflow analyses in Datalog

Claim (Lower bounds on IN & OUT)

<u>Claim</u>: ① OUT[Entry] = ∅

Claim (Lower bounds on IN & OUT)

(2') $\forall_i^n \ \mathrm{OUT}[n_i] \subseteq \mathrm{IN}[n]$

$\forall_i^n \ d, \ d \in \mathrm{OUT}[n_i] \Rightarrow d \in \mathrm{IN}[n]$

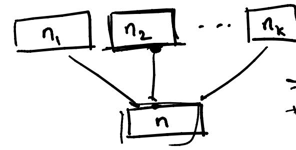
(3') $\forall n, \ (\mathrm{IN}[n] \setminus \mathrm{KILL}[n]) \cup \mathrm{GEN}[n] \subseteq \mathrm{OUT}[n]$

(4') $\forall n \ d, \ d \in \mathrm{IN}[n] \setminus \mathrm{KILL}[n]$

$\Rightarrow d \in \mathrm{OUT}[n]$

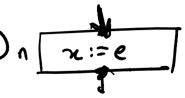
$\forall n \ d, \ d \in \mathrm{GEN}[n] \Rightarrow d \in \mathrm{OUT}[n]$.

Claim: (1) $\mathrm{OUT}[\mathrm{Entry}] = \emptyset$

(2) 

$\mathrm{IN}[n] = \bigcup_i \mathrm{OUT}[n_i]$

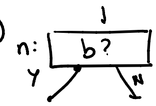
~~$\forall i. \ \mathrm{OUT}[n_i] \subseteq \mathrm{IN}[n]$~~

(3) n 

$\mathrm{OUT}[n] = (\mathrm{IN}[n] \setminus \mathrm{KILL}[n]) \cup \mathrm{GEN}[n]$

Every dataflow fact involving x → $\mathrm{KILL}[n] = \{d \mid d: x := e\}$

The statement x:=e → $\mathrm{GEN}[n] = \{n\}$

(4) n: 

$\mathrm{OUT}[n] = (\mathrm{IN}[n] \setminus \mathrm{KILL}[n]) \cup \mathrm{GEN}[n]$

Conditional statements don't kill any facts; don't give rise to any dataflow facts. → $\mathrm{KILL}[n] = \emptyset$, $\mathrm{GEN}[n] = \emptyset$

The output of the chaotic iteration algorithm is the smallest solution to claims (2') (3') (4').