Aman Choudhary 17920

1) Let $d \in IA$, be input state to the transfer function.

i) $x = x + y$

$f(d) = d'$ where,

$$\begin{cases} d' = \perp, & \text{if } d \text{ is } \perp \\ d'(x) = [u_x + u_y, l_x + l_y], \; d'(y) = d(y) & \text{if } d(x) = [u_x, l_x] \text{ and, } d(y) = [u_y, l_y] \end{cases}$$

NOTE: we are assuming, $x + \infty = \infty$, if $x \neq -\infty$
and, $x - \infty = -\infty$, if $x \neq \infty$

ii) $x = y + 1$

$f(d) = d'$ where,

$$\begin{cases} d' = \perp, & \text{if } d \text{ is } \perp \\ d'(x) = [u_y + 1, l_y + 1], \; d'(y) = d(y) & \text{if } d(y) = [u_y, l_y] \end{cases}$$

iii) Let $d \in IA$, such that $d(x) = [u_x, l_x]$ and $d(y) = [u_y, l_y]$

$$\gamma_I(d) = \{(i, j) \mid i \in [u_x, l_x] \text{ and } j \in [u_y, l_y]\}$$
$$= \phi \text{ , if } d \text{ is } \perp$$

iv) Let c' be an element of the concrete lattice such that,

$$c' = \{(i, j) \mid i \in C_X, j \in C_Y \text{ where } C_X, C_Y \subseteq Z\}$$

$\alpha_I(c') = d$ where,

$$d(x) = \begin{cases} [\min(C_X), \max(C_X)], & \text{if } C_X \neq \phi \\ [-\infty, \infty], & \text{otherwise} \end{cases}$$

$$d(y) = \begin{cases} [\min(C_Y), \max(C_Y)], & \text{if } C_Y \neq \phi \\ [-\infty, \infty], & \text{otherwise} \end{cases}$$

$$d = \perp \text{ , if } c' = \phi$$

(2)

i) $\alpha_{IC}(d) = \bigcup_{v \in Var} C_v$ where,

$$C_v = \begin{cases} \{(v, u_v)\}, & \text{if } d(v) = [u_v, l_v] \text{ and } u_v = l_v \\ \phi, & \text{if } d(v) = [u_v, l_v] \text{ and } u_v < l_v \end{cases}$$

$\alpha_{IC}(d) = \perp$, if d is $\perp$

ii) $\gamma_{CI}(c) = d$ where for each $v \in Var$

$$d(v) = \begin{cases} [u_v, u_v], & \text{if } (v, u_v) \in c, \text{ and } u_v \in \mathbb{Z} \\ [-\infty, \infty], & \text{otherwise} \end{cases}$$

$\gamma_{CI}(c) = \perp$, if c is $\perp$

iii) Let, us assume we have a program with two variables x and y,

$d \in IA$ such that, $d(x) = [2, 3]$, $d(y) = [-\infty, \infty]$

$\alpha_{IC}(d) = \phi$

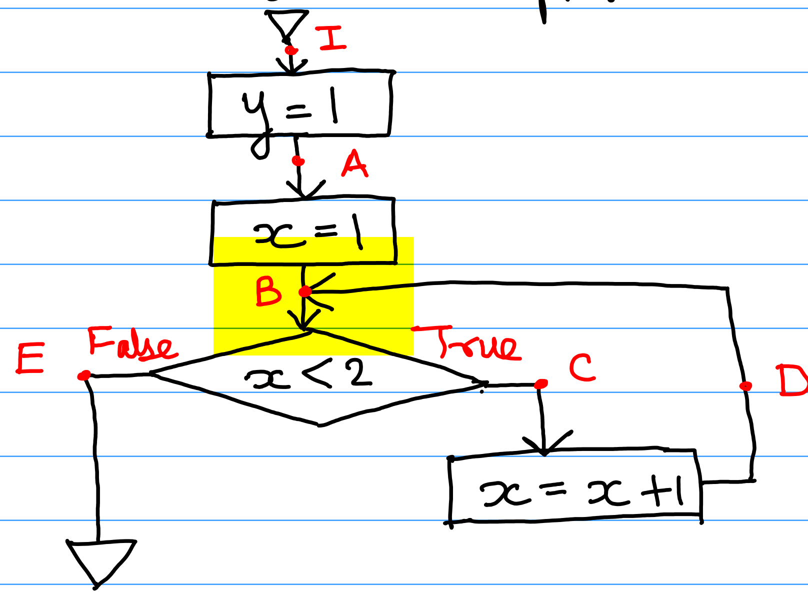
$\gamma_{CI}(\alpha_{IC}(d)) = d'$ where $d'(x) = d'(y) = [-\infty, \infty]$

Clearly $d' \sqsupset d$ because $d'(x) \supset d(x)$

③ Program:

```
y = 1 ;
x = 1 ;
while (x < 2){
    x = x+1 ;
}
```

Control Flow Graph:



Abstract states:

Path	IA lattice	CP lattice
I	$d[x \to [-\infty,\infty], y \to [-\infty,\infty]]$	$\{\ \}$
IA	$d[x \to [-\infty,\infty], y \to [1,1]]$	$\{(y,1)\}$
✓ IAB	$d[x \to [1,1], y \to [1,1]]$	$\{(x,1),(y,1)\}$
IABC	$d[x \to [1,1], y \to [1,1]]$	$\{(x,1),(y,1)\}$
IABCD	$d[x \to [2,2], y \to [1,1]]$	$\{(x,2),(y,1)\}$
✓ IABCDB	$d[x \to [2,2], y \to [1,1]]$	$\{(x,2),(y,1)\}$
IABCDBE	$d[x \to [2,2], y \to [1,1]]$	$\{(x,2),(y,1)\}$

NOTE: If $d_1(x) = [u_1, l_1]$ and $d_2(x) = [u_2, l_2]$, then $d_1 \sqcup d_2 = d'$ and $d'(x) = [\min(u_1,u_2), \max(l_1,l_2)]$

At point B,

d_I = Abstract JOP using IA lattice $= d[x \to [1,2], y \to [1,1]]$

d_C = Abstract JOP using CP lattice $= \{(y,1)\}$

$\gamma_I(d_I) = \{(1,1),(2,1)\}$

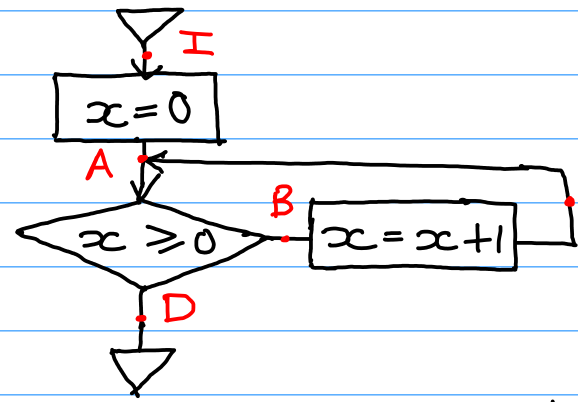
$\gamma_C(d_C) = \{(i,1) \mid i \in \mathbb{Z}\}$

Clearly, $\gamma_I(d_I) \subset \gamma_C(d_C)$

i) ④ **Program:**

```
x = 0;
while (x >= 0){
    x = x + 1;
}
```

Control Flow Graph:



Kildall's Algo

$(d_I, d_A, d_B, d_C, d_D) \in \overline{IA}$ represents the current abstract states across all program points as we run Kildall's algorithm for above program.

Step 1: $(d[x \to [-\infty, \infty]], \perp, \perp, \perp, \perp)$ (I, A, B, C, D)

Status of each program point, red: marked, black: unmarked

Step 2: $(d[x \to [-\infty, \infty]], \underline{d[x \to [0,0]]}, \perp, \perp, \perp)$ (I, A, B, C, D)

Step 3: $(d[x \to [-\infty, \infty]], d[x \to [0,0]], \underline{d[x \to [0,0]]}, \perp, \perp)$ (I, A, B, C, D)

Step 4: $(d[x \to [-\infty, \infty]], d[x \to [0,0]], d[x \to [0,0]], \underline{d[x \to [1,1]]}, \perp)$ (I, A, B, C, D)

Step 5: $(d[x \to [-\infty, \infty]], \underline{d[x \to [0,1]]}, d[x \to [0,0]], d[x \to [1,1]], \perp)$ (I, A, B, C, D)

Step 6: $(d[x \to [-\infty, \infty]], d[x \to [0,1]], \underline{d[x \to [0,1]]}, d[x \to [1,1]], \perp)$ (I, A, B, C, D)

Step 7: $(d[x \to [-\infty, \infty]], d[x \to [0,1]], d[x \to [0,1]], \underline{d[x \to [1,2]]}, \perp)$ (I, A, B, C, D)

Step 8: $(d[x \to [-\infty, \infty]], \underline{d[x \to [0,2]]}, d[x \to [0,1]], d[x \to [1,2]], \perp)$ (I, A, B, C, D)

⋮

and so on...

⋮

As we can see, the program is running in an infinite loop, and the Kildall's algorithm will always encounter an unmarked point. Thus, it will never terminate.

ii) Let us look at a particular chain in the IA lattice:

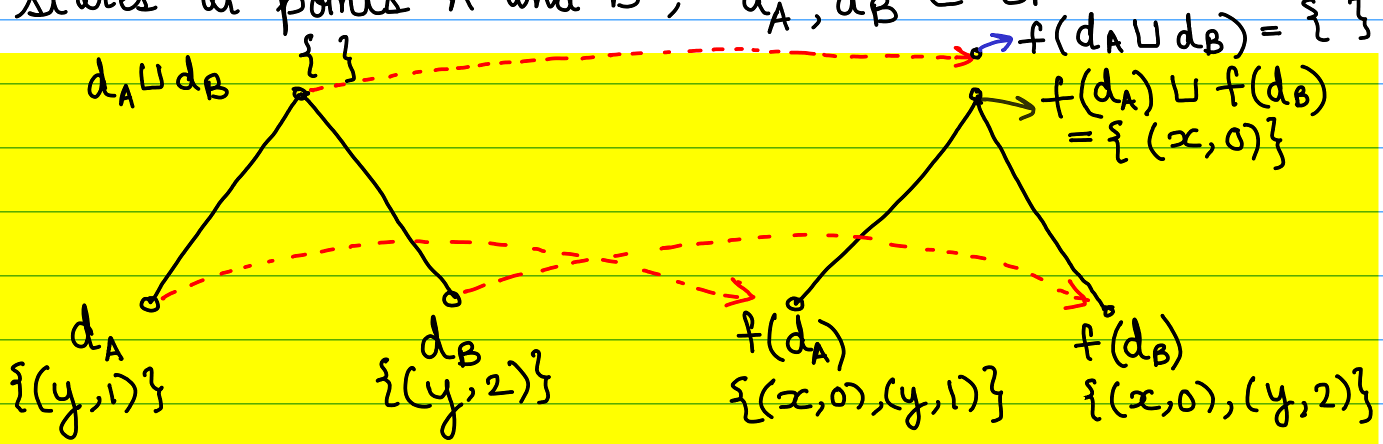
$$[0,0] \leq [0,1] \leq [0,2] \leq \ldots$$

This chain is an infinite sequence, and hence the height of IA lattice is not finite.

5

i) Let, $\boxed{x = y * 0}$ be an assignment statement

Let, $d_A = \{(y,1)\}$ and $d_B = \{(y,2)\}$ be the abstract states at points A and B, $d_A, d_B \in CP$

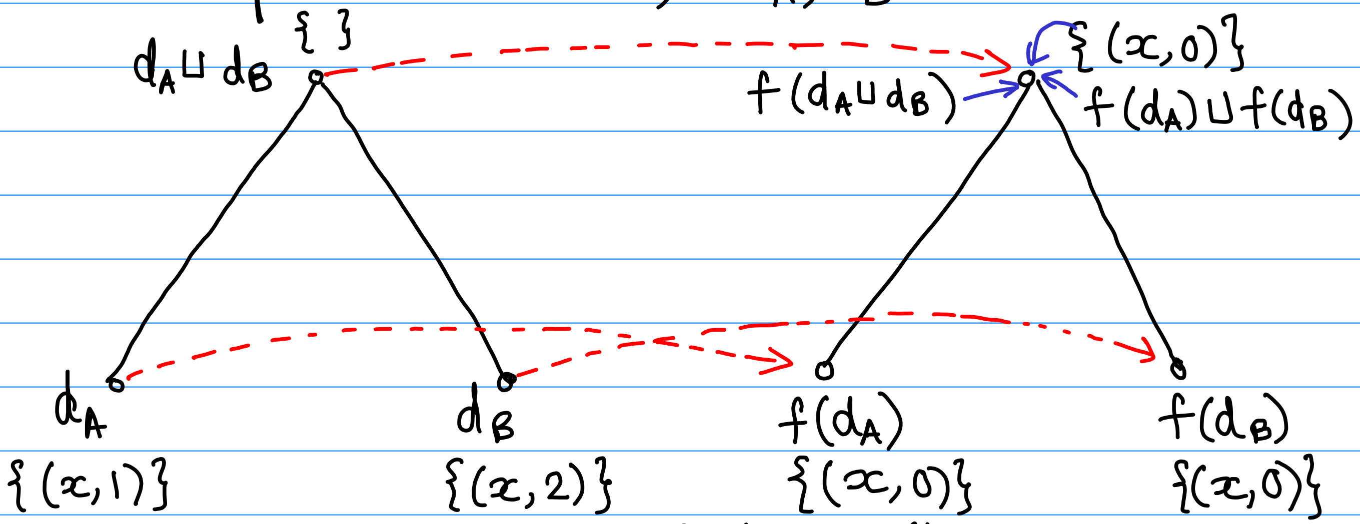


So, $f(d_A \sqcup d_B) \neq f(d_A) \sqcup f(d_B)$

Hence, f is not distributive in this case.

ii) Let, $\boxed{x = 0}$ be an assignment statement.

Let, $d_A = \{(x,1)\}$ and $d_B = \{(x,2)\}$ be the abstract states at points A and B, $d_A, d_B \in CP$



So, $f(d_A \sqcup d_B) = f(d_A) \sqcup f(d_B)$

Hence, f is distributive in this case.

iii)