Aman Choudhary 17920

1. Let us consider the available expression analysis. We denote the set of all functions as $D_S = \{ f \mid f : D \to D \}$. We use $\leqslant$ to define a lattice over D_S i.e. $D_L = (D_S, \leqslant)$.
We consider the function 'in' which inverts the availability of expression i.e. $in \in D_L$

$$\begin{aligned} \perp &\to \perp \\ 0 &\to 1 \\ 1 &\to 0 \end{aligned}$$

Let us assume we have the following data flow equations for an arbitrary program of two points,

$$y_A = 0$$
$$y_B = in \circ y_A$$

The function $\bar{f}$ induced by above eq^n^

$$\bar{f}(\langle y_A, y_B \rangle) = \langle 0, in \circ y_A \rangle$$

We consider $x = \langle 1,1 \rangle$, $y = \langle 0,0 \rangle$ where $x, y \in D_L$ such that $x \leqslant y$. Now,

$$\bar{f}(x) = \langle 0, in \circ 1 \rangle = \langle 0,0 \rangle$$
$$\bar{f}(y) = \langle 0, in \circ 0 \rangle = \langle 0,1 \rangle$$

Here, $\bar{f}(x) \geqslant \bar{f}(y)$

Since, $x \leqslant y$ and $f(x) \geqslant f(y)$, the function $\bar{f}$ is not monotonic.

(2) We first write eq^n for ϕ's,

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= o \circ y_{A,A} \\
y_{A,C} &= y_{F,J} \circ y_{A,B} \\
y_{A,E} &= id \circ y_{A,C}
\end{aligned}
$$

$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,L} &= in \circ y_{F,K} \\
y_{F,H} &= in \circ y_{F,G} \\
y_{F,P} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= in \circ y_{F,P} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,L}
\end{aligned}
$$

Here, $in : D \to D$ is a function for inverting the parity.

$$
\begin{aligned}
\perp &\to \perp \\
o &\to e \\
e &\to o \\
oe &\to oe
\end{aligned}
$$

$id \to$ identity func^c
$o \to$ odd func^c
$e \to$ even func^c

Running Kildall's algorithm to approximate ϕ's,

A	B	C	E	F	G	K	L	H	P	I	J
id	_I_	_I_	_I_	id	_I_	_I_	_I_	_I_	_I_	_I_	_I_
id	o										
	o	_I_									
				id	id	id					
					id			in			
						id	in				
							in				in
								in	id		
									id	in	
										in	in
		e							id		in
		e	e								
			e								
id	**o**	**e**	**e**	**id**	**id**	**id**	**in**	**in**	**id**	**in**	**in**

Next, we write eq^n 2 to capture JVP,

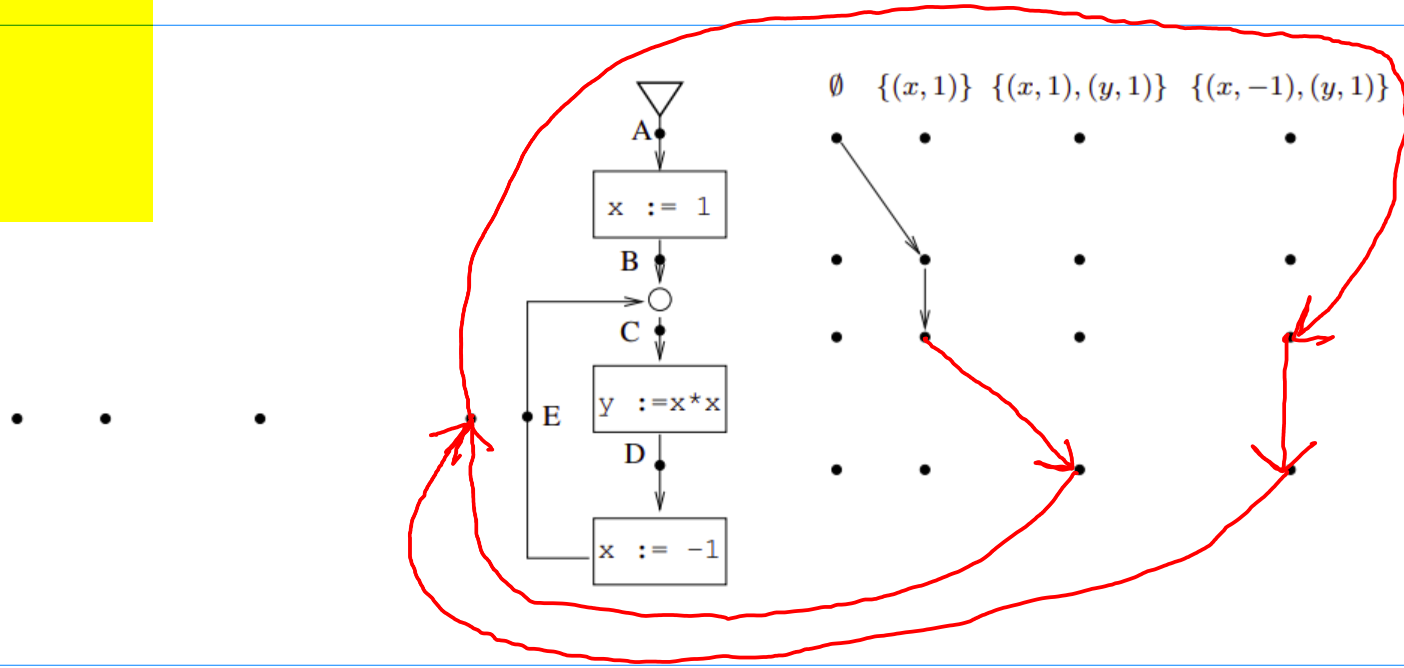
$$x_A = oe$$
$$x_B = o(x_A)$$
$$x_C = e(x_A)$$
$$x_E = e(x_A)$$

$$x_F = x_B \sqcup x_H$$
$$x_G = id(x_F)$$
$$x_K = id(x_F)$$
$$x_L = in(x_F)$$
$$x_H = in(x_F)$$
$$x_P = id(x_F)$$
$$x_I = in(x_F)$$
$$x_J = in(x_F)$$

A	B	C	E	F	G	K	L	H	P	I	J
oe	_|_	_|_	_|_	id	_|_	_|_	_|_	_|_	_|_	_|_	_|_
oe	o	e	e								
	o			o							
		e									
			e								
				o	o	o	e	e	o	e	e
					o						
						o					
							e				
				oe				e			
									o		
										e	
											e
				oe	oe	oe	oe	oe	oe	oe	oe
					oe						
						oe					
							oe				
								oe			
									oe		
										oe	
											oe
oe	o	e	e	oe	oe	oe	oe	oe	oe	oe	oe

③

(a) (i)



(ii) JOP values can be computed by taking join over abstract values that reach a point,

$A \rightarrow \phi$

$B \rightarrow \{(x,1)\}$

$C \rightarrow \{(x,1)\} \sqcup \{(x,-1),(y,1)\} = \phi$

$D \rightarrow \{(x,1),(y,1)\} \sqcup \{(x,-1),(y,1)\} = \{(y,1)\}$

$E \rightarrow \{(x,-1),(y,1)\}$

(c) Algorithm to compute the exact JOP for a given program P and abstract interpretation A.

Assumption: Lattice is of finite height.

Part 1: Compute exploded graph

1) Initial set of vertices and edges.

$V = \{(P, d_0) \mid P \in \text{Program points}\}$

$E = \{\ \}$

For any vertex v, let v.p denote the program point, and v.a denote the abstract state i.e.

$$v = (\underbrace{\_}_{v.p}, \underbrace{\_}_{v.a})$$

2) Let D′ denote the set of seen abstract states i.e. at any point in time, V = Program points × D′. Initially, $D' = \{d_0\}$

3) Initialize a queue, Q to store vertices. Push (I, d_0) to Q. Here, I is the initial program point.

4) while Q is not empty,
   (i) curr_vertex = Q.front()
   (ii) Q.pop()
   (iii) Apply transfer function to curr_vertex.a. If the resultant abstract state $d \notin D'$. Add new vertices to V i.e., $V = V \cup \{(P, d) \mid P \in \text{Program Points}\}$. Also, $D' = D' \cup \{d\}$
   (iv) Now, add edges to the concerned vertices after applying the transfer function. Let, the set of vertices to which we add new edges from curr_vertex be V′.
   (v) Push into Q all vertices from V′ which have exactly one incoming edge i.e they have been visited the first time.

Part 2: Compute JOP

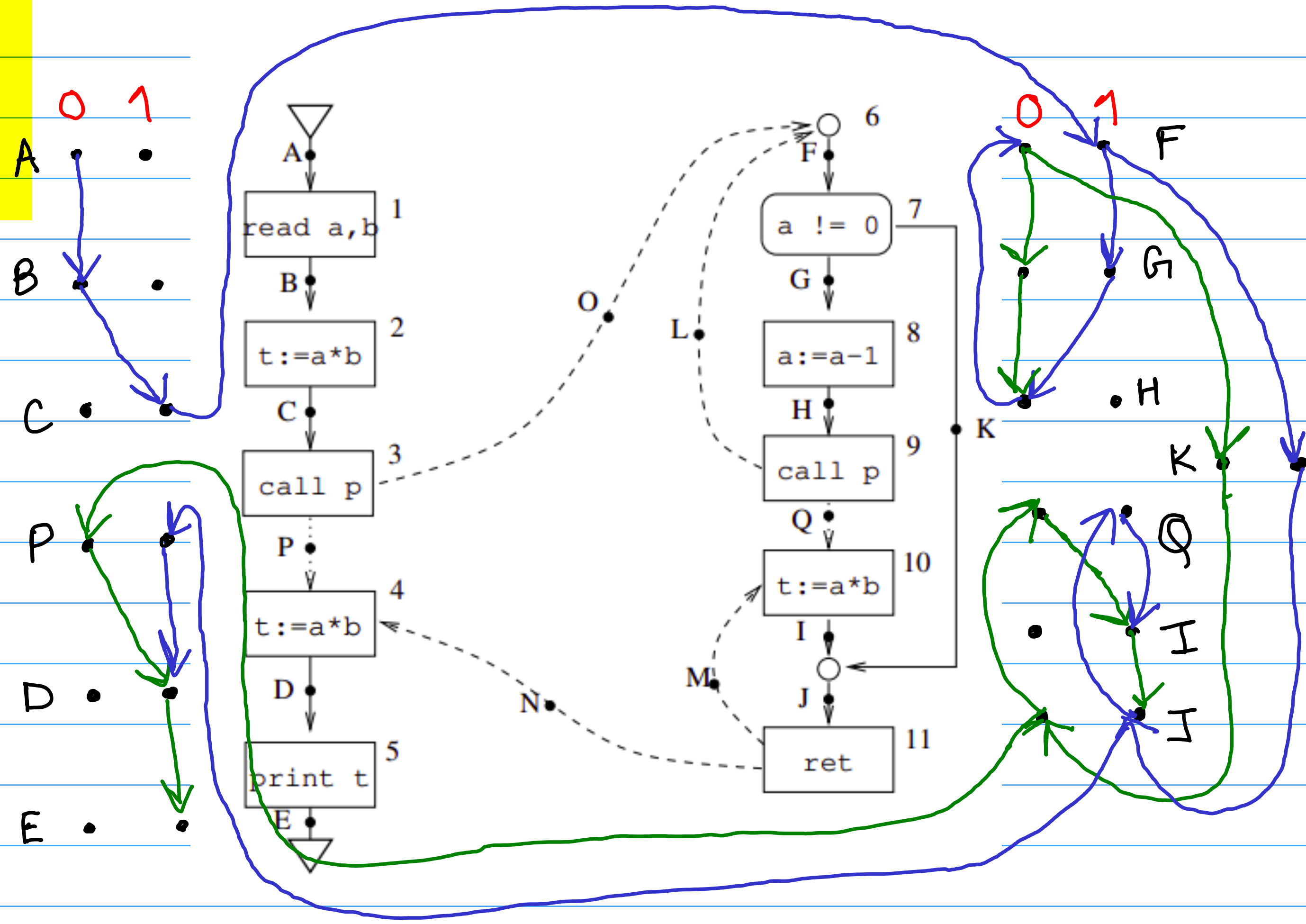
For any program point X, let

$$V_X = \{v \mid v.p = X \text{ and } v \text{ has an incoming edge}\}$$

$$JOP_X = \bigsqcup_{v \in V_X} v.a$$

4

(a)



(b) (D,0) is not reachable from (A,0)
(D,1) is reachable from (A,0).

JVP at D → 1

(c)