Interprocedural analysis: Sharir-Pnueli's functional approach

Deepak D'Souza

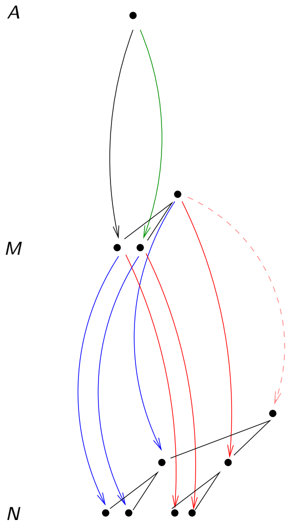
Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

18 September 2013

Outline

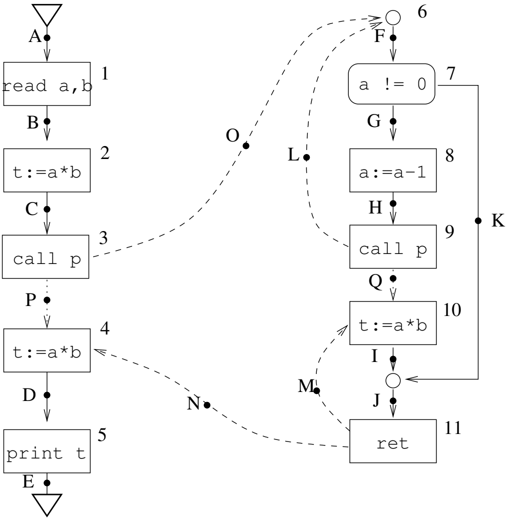
Equations to capture JOP: why it works

- We want JOP at N.
- If transfer functions are distributive, then we can take join over paths at an any intermediate point M, and then join over paths from M to N.



Equation solving: Problems with naive approach

- In non-procedural case, we setup equations to capture JOP assuming distributivity. Least solution to these equations gave us exact/over-approx JOP depending on distributive/monotonic framework.
- Try to set up similar equations for x_N (JVP at program point N).
- How do we describe x_N in terms of x_J?

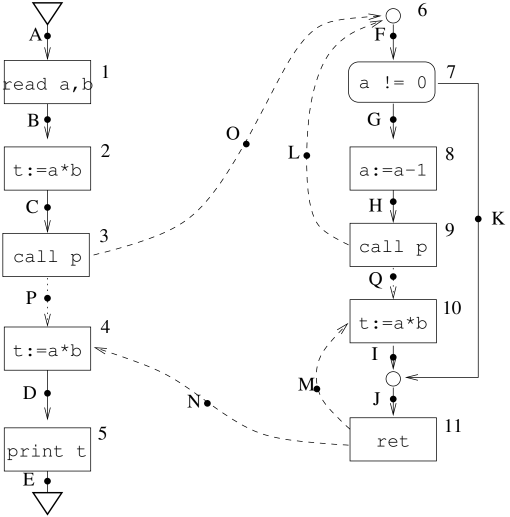


Instead try to capture join over complete paths first

- Set up equations to capture join over complete paths.
- Now set up equations to capture JVP using join over complete path values.

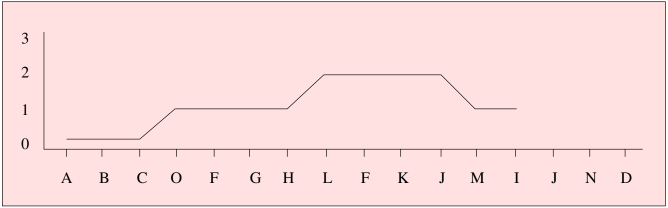
Notation

- Root of procedure p is denoted r_p.
- Exit (return) of procedure p is denoted e_p.
- Sometimes use r_1 for r_{main}.
- Assume WLOG that `main` is not called.

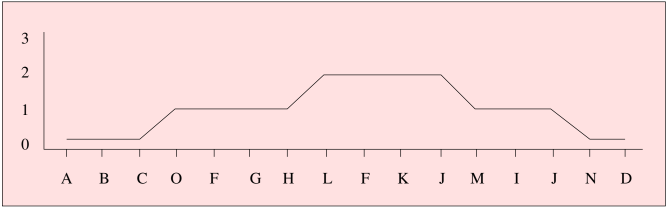


Example paths

An example valid path in $IVP(r_1, I)$.



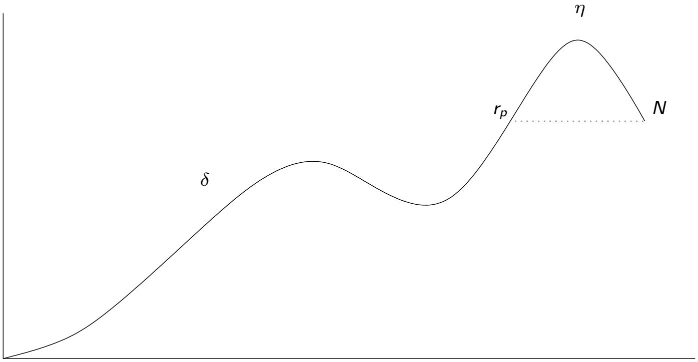
An example valid and complete path in $IVP_0(r_1, D)$.



Path "FGHLFKJMIJ" is valid and complete and is in $IVP_0(r_p, J)$.

Basic idea: Why join over complete paths help

An IVP path ρ from r_1 to N in procedure p can be written as $\delta \cdot \eta$ where δ is in $\mathrm{IVP}(r_1, r_p)$, and η is in $\mathrm{IVP}_0(r_p, N)$.



Path η is suffix after last pending call to procedure p was made.

Valid and complete paths from r_p to N

For a procedure p and node N in p, define:

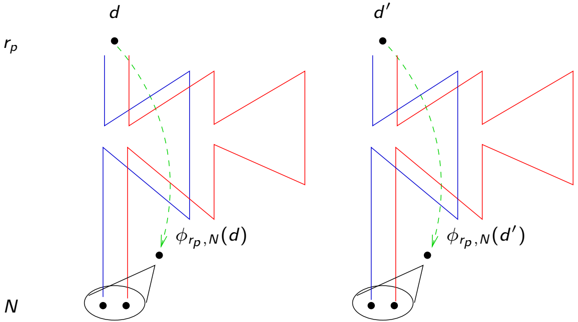
$$\phi_{r_p,N} : D \to D$$

given by

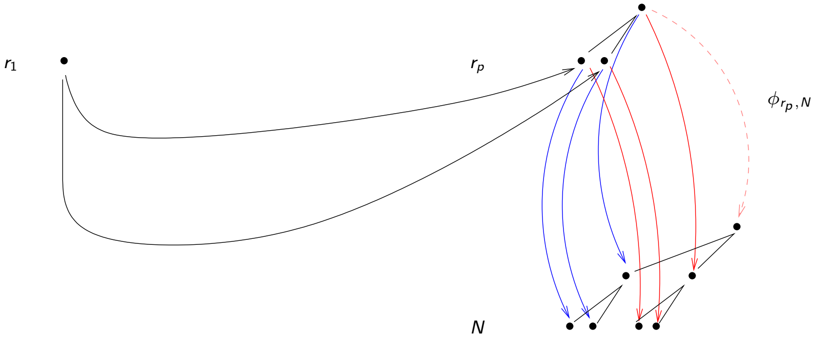
$$\phi_{r_p,N}(d) = \bigsqcup_{\text{paths } \rho \in \mathrm{IVP}_0(r_p,N)} f_\rho(d).$$

$\phi_{r_p,N}$ is thus the join of all functions f_ρ where ρ is an interprocedurally valid and complete path from r_p to N.

Visualizing $\phi_{r_p,N}$



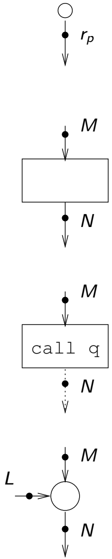
Using $\phi_{r_p,N}$'s to get JVP values



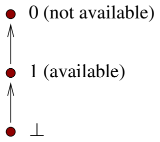
Assuming distributivity of underlying transfer functions, JVP value at N equals $\phi_{r_p,N}$ applied to JVP value at r_p.

Equations (1) to capture $\phi_{r_p,N}$

$$
\begin{aligned}
y_{r_p,r_p} &= id_D & (root)\\
y_{r_p,N} &= f_{MN} \circ y_{r_p,M} & (stmt)\\
y_{r_p,N} &= y_{r_q,e_q} \circ y_{r_p,M} & (call)\\
y_{r_p,N} &= y_{r_p,L} \sqcup y_{r_p,M}. & (join)
\end{aligned}
$$

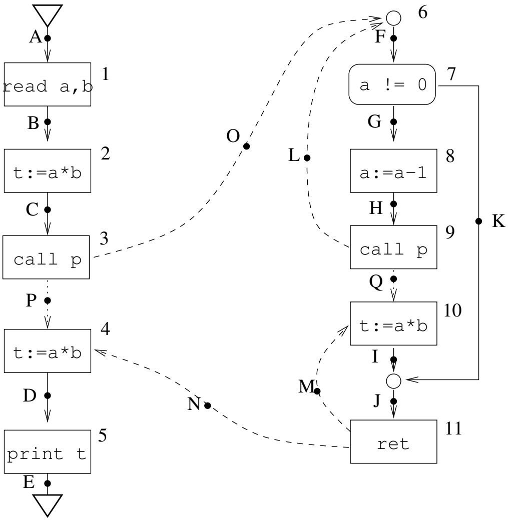


Example: Available expressions analysis

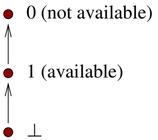


Lattice for Av-Exp analysis.

- Is a*b available at program point N?

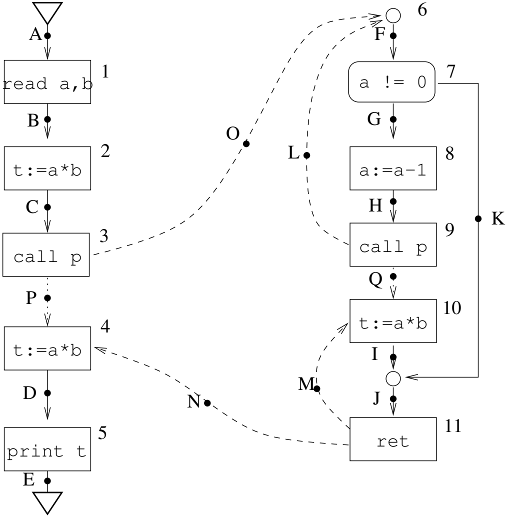


Example: Available expressions analysis

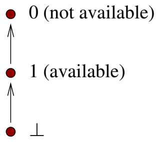


Lattice for Av-Exp analysis.

- Is a*b available at program point N?
- No if we consider all paths.

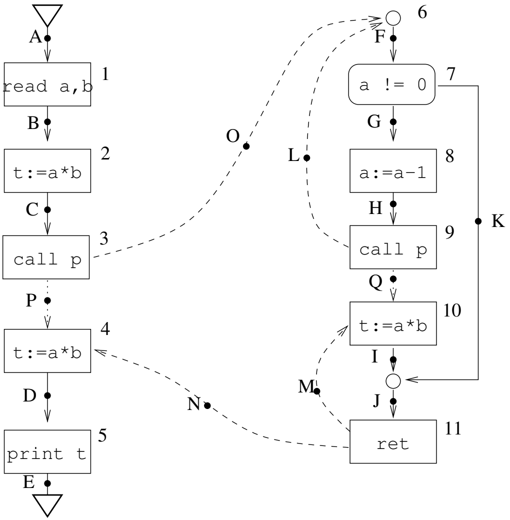


Example: Available expressions analysis



Lattice for Av-Exp analysis.

- Is a*b available at program point *N*?
- No if we consider all paths.
- Yes if we consider interprocedurally valid paths only.



Functions we will use for example analysis

- $D = \{\bot, 1, 0\}$.
- $\mathbf{0} : D \to D$ given by

$$\begin{aligned} \bot &\mapsto \bot \\ 0 &\mapsto 0 \\ 1 &\mapsto 0 \end{aligned}$$

- $\mathbf{1} : D \to D$ given by

$$\begin{aligned} \bot &\mapsto \bot \\ 0 &\mapsto 1 \\ 1 &\mapsto 1 \end{aligned}$$

- $\mathbf{id} : D \to D$ given by

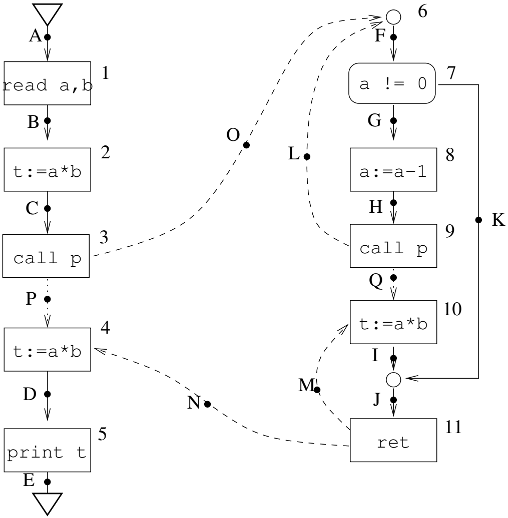
$$\begin{aligned} \bot &\mapsto \bot \\ 0 &\mapsto 0 \\ 1 &\mapsto 1 \end{aligned}$$

- Ordering: $\mathbf{1} \leq \mathbf{id} \leq \mathbf{0}$.

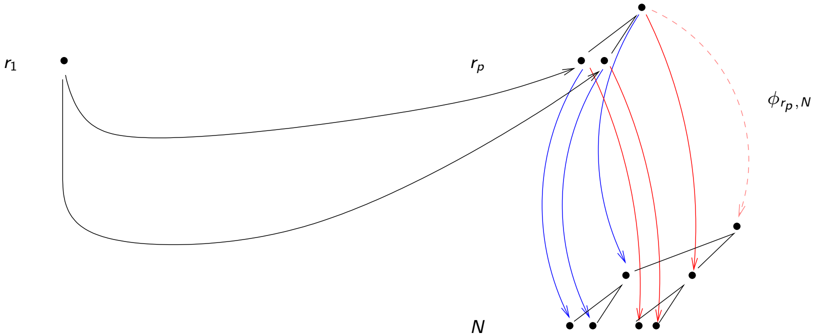
Example: Equations for ϕ's

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Using $\phi_{r_p,N}$'s to get JVP values



Assuming distributivity of underlying transfer functions, JVP value at N equals $\phi_{r_p,N}$ applied to JVP value at r_p.

Equations (2) to capture JVP

$$
\begin{array}{lcl}
x_1 & = & d_0 \\
x_{r_p} & = & \bigsqcup_{\text{calls } C \text{ to } p} x_C \\
x_N & = & \phi_{r_p,N}(x_{r_p}) \qquad \text{for } N \in \text{ProgPts}(p) - \{r_p\}.
\end{array}
$$

Example: Equations for x_N's (JVP)

$$
\begin{aligned}
x_A &= 0\\
x_B &= \mathbf{0}(x_A)\\
x_C &= \mathbf{1}(x_A)\\
x_P &= \mathbf{1}(x_A)\\
x_D &= \mathbf{1}(x_A)\\
x_E &= \mathbf{1}(x_A)
\end{aligned}
$$

$$
\begin{aligned}
x_F &= x_C \sqcup x_H\\
x_G &= id(x_F)\\
x_K &= id(x_F)\\
x_H &= \mathbf{0}(x_F)\\
x_Q &= \mathbf{0}(x_F)\\
x_I &= \mathbf{1}(x_F)\\
x_J &= id(x_F).
\end{aligned}
$$

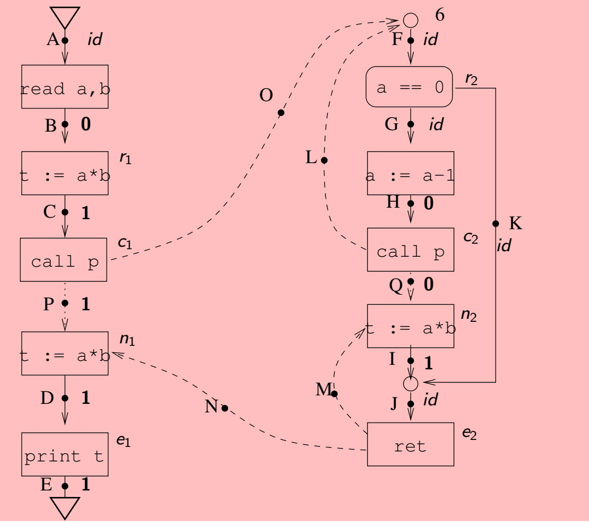


Fig. shows values of $\phi_{r_p,N}$'s in bold.

Correctness claims

- Consider lattice $(F, \leq)$ of functions from D to D, obtained by closing the transfer functions, identity, and $f_\perp : d \mapsto \perp$ under composition and join. (Alternatively we can take F to be all monotone functions on D.)
- Ordering is $f \leq g$ iff $f(d) \leq g(d)$ for each $d \in D$.
- $(F, \leq)$ is also a complete lattice.
- $\overline{f}$ induced by Eq (1) is monotone on complete lattice $(\overline{F}, \overline{\leq})$.
  - Sufficient to argue that function composition $\circ$ is monotone when applied to monotone functions.
  - Join operation $\bigsqcup$ is monotone.
- LFP / least solution (say $y^*_{r_p,N}$'s) exists by Knaster-Tarski.
- Each $y^*_{r_p,N}$ is necessarily monotonic.

Claim

$\phi_{r_p,N}$'s are the least solution to Eq (1) (i.e. $\phi_{r_p,N} = y^*_{r_p,N}$) when f_{MN}'s are distributive. Otherwise each $\phi_{r_p,N} \leq y^*_{r_p,N}$.

Using Kildall to compute LFP

- We can use Kildall's algo to compute the LFP of these equations as follows.
  - Initialize the value at program points with RHS of the constant equations (in this case *id* at entry of procedures), and the bottom value (in this case $f_{\perp}$) everywhere else.
  - Mark all values
  - Pick a marked value at point say N, and "propagate" it (i.e. for any node M in the LHS of an equation in which N occurs in the RHS, evaluate M and join it with the existing value at M). Mark as before in Kildall's algo.
  - Stop when no more marked values to propagate.
- Kildall's algo will compute $y^*_{r_p,N}$ if D is finite. Note that finite height of $(D, \leq)$ is not sufficient for termination.

Correctness and algo - II

Consider Eq (2)':

$$
\begin{array}{lcl}
x_1 & = & d_0 \\
x_{r_p} & = & \bigsqcup_{\text{calls } C \text{ to } p} x_C \\
x_N & = & y^*_{r_p,N}(x_{r_p}) \qquad \text{for } N \in N_p - \{r_p\}.
\end{array}
$$

(Recall that $y^*_{r_p,N}$'s are the least solution of Eq (1).)

- $\overline{f}$ induced by Eq (2)' is a monotone function on the complete lattice $(\overline{D}, \overline{\leq})$.
- LFP / least solution (say x^*_N's) exists by Knaster-Tarski.

Claim

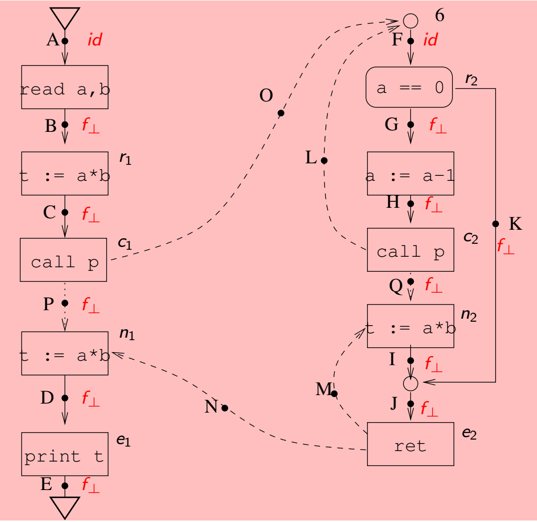
JVP values are the least solution to Eq (2)' (i.e. $\mathrm{JVP}_N = x^*_N$) when f_{MN}'s are distributive. Otherwise $\mathrm{JVP}_N \leq x^*_N$ for each N.

Kleene/Kildall's algo will compute x^*_N's (assuming D finite).

Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

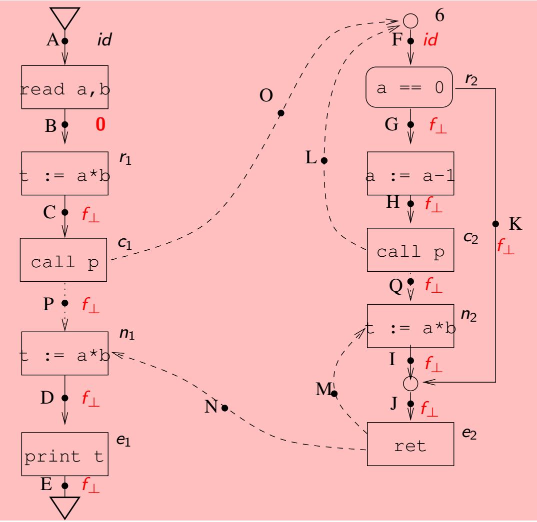
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

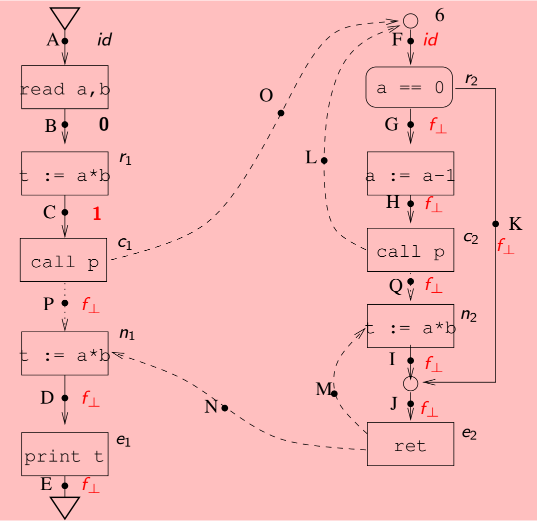
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

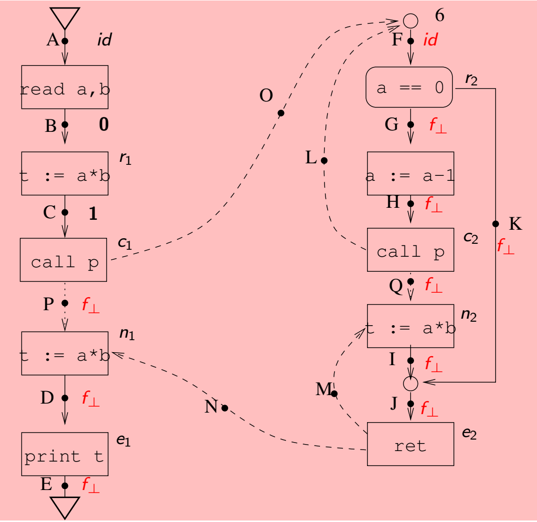
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id\\
y_{A,B} &= \mathbf{0} \circ y_{A,A}\\
y_{A,C} &= \mathbf{1} \circ y_{A,B}\\
y_{A,P} &= y_{F,J} \circ y_{A,C}\\
y_{A,D} &= \mathbf{1} \circ y_{A,P}\\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

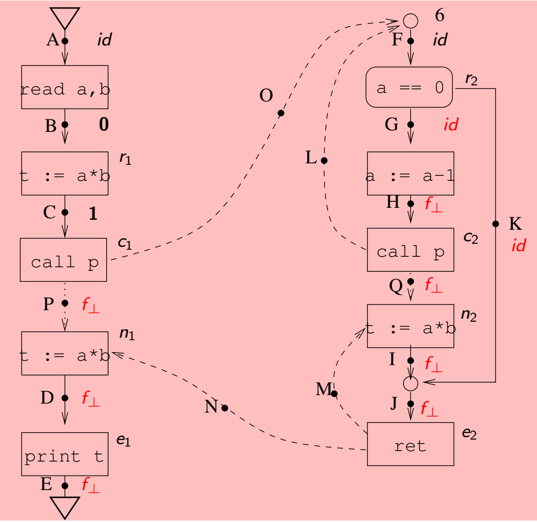
$$
\begin{aligned}
y_{F,F} &= id\\
y_{F,G} &= id \circ y_{F,F}\\
y_{F,K} &= id \circ y_{F,F}\\
y_{F,H} &= \mathbf{0} \circ y_{F,G}\\
y_{F,Q} &= y_{F,J} \circ y_{F,H}\\
y_{F,I} &= \mathbf{1} \circ y_{F,Q}\\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

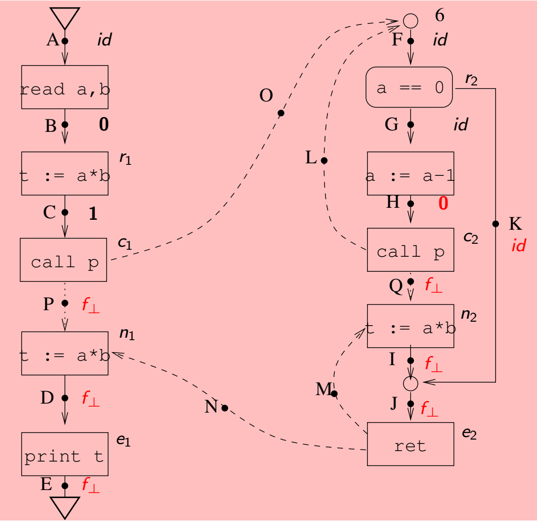
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

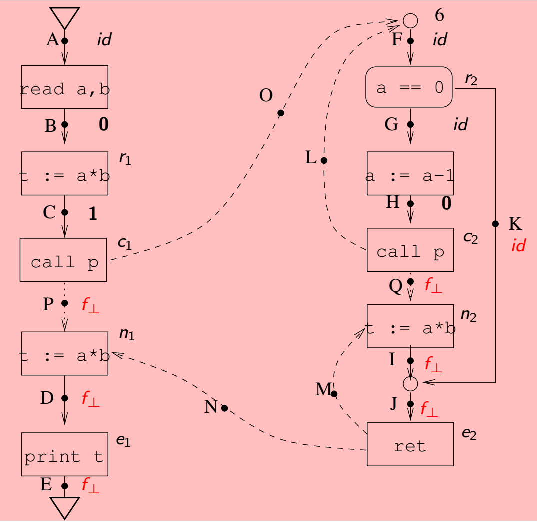
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

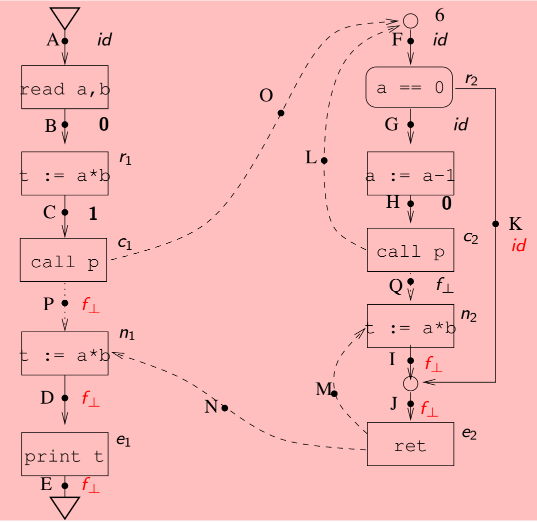
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

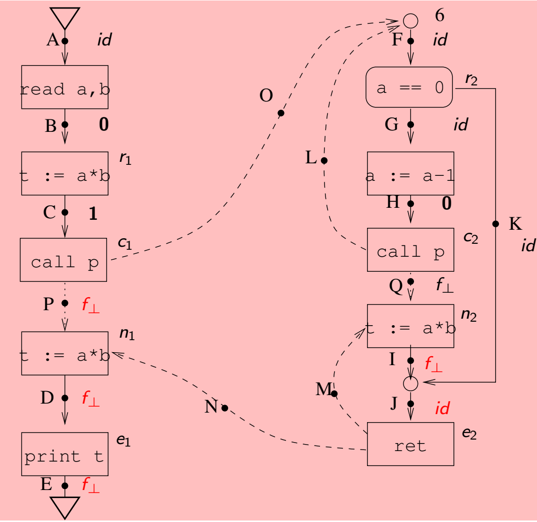
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

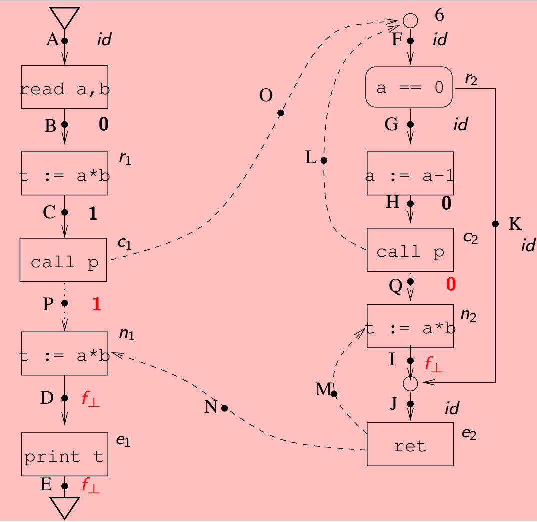
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id\\
y_{A,B} &= \mathbf{0} \circ y_{A,A}\\
y_{A,C} &= \mathbf{1} \circ y_{A,B}\\
y_{A,P} &= y_{F,J} \circ y_{A,C}\\
y_{A,D} &= \mathbf{1} \circ y_{A,P}\\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

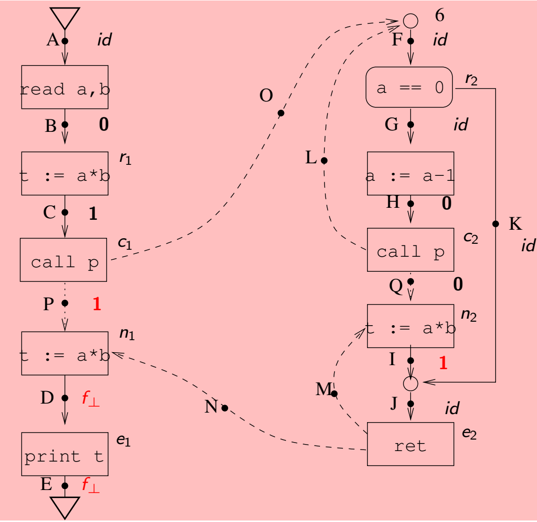
$$
\begin{aligned}
y_{F,F} &= id\\
y_{F,G} &= id \circ y_{F,F}\\
y_{F,K} &= id \circ y_{F,F}\\
y_{F,H} &= \mathbf{0} \circ y_{F,G}\\
y_{F,Q} &= y_{F,J} \circ y_{F,H}\\
y_{F,I} &= \mathbf{1} \circ y_{F,Q}\\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

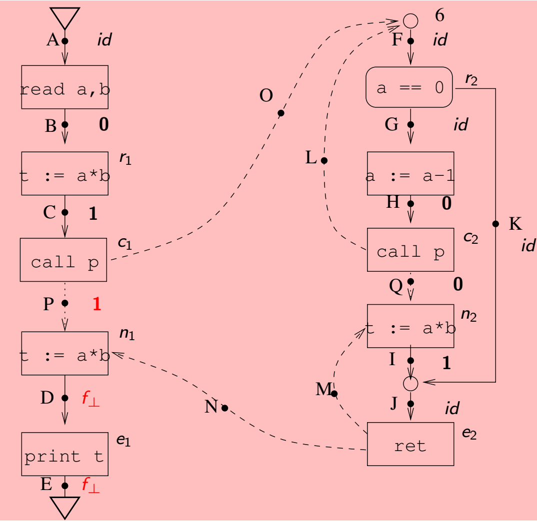
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

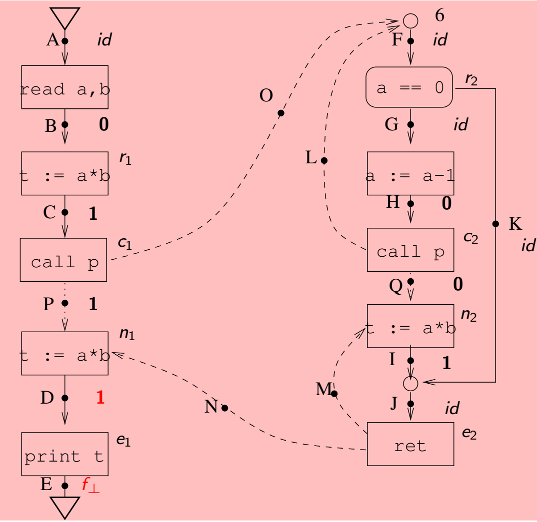
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

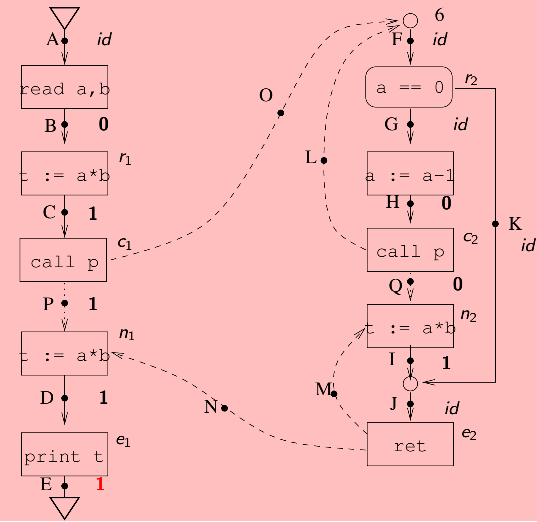
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

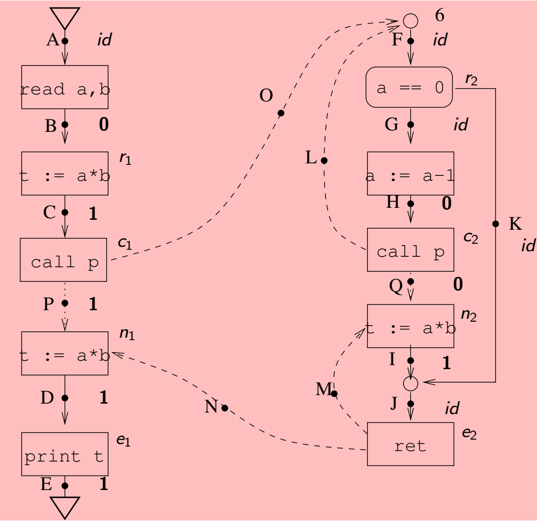
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing $\phi_{r_p,N}$'s ($y^*_{r_p,N}$ to be precise) using Kildall's algo

$$
\begin{aligned}
y_{A,A} &= id \\
y_{A,B} &= \mathbf{0} \circ y_{A,A} \\
y_{A,C} &= \mathbf{1} \circ y_{A,B} \\
y_{A,P} &= y_{F,J} \circ y_{A,C} \\
y_{A,D} &= \mathbf{1} \circ y_{A,P} \\
y_{A,E} &= id \circ y_{A,D}
\end{aligned}
$$

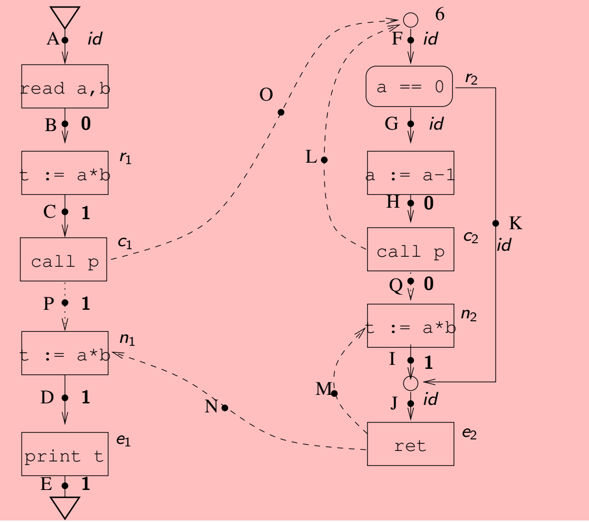
$$
\begin{aligned}
y_{F,F} &= id \\
y_{F,G} &= id \circ y_{F,F} \\
y_{F,K} &= id \circ y_{F,F} \\
y_{F,H} &= \mathbf{0} \circ y_{F,G} \\
y_{F,Q} &= y_{F,J} \circ y_{F,H} \\
y_{F,I} &= \mathbf{1} \circ y_{F,Q} \\
y_{F,J} &= y_{F,I} \sqcup y_{F,K}
\end{aligned}
$$



Example: Computing JVP values (x_N^*'s to be precise)

$$
\begin{aligned}
x_A &= 0 \\
x_B &= \mathbf{0}(x_A) \\
x_C &= \mathbf{1}(x_A) \\
x_P &= \mathbf{1}(x_A) \\
x_D &= \mathbf{1}(x_A) \\
x_E &= \mathbf{1}(x_A)
\end{aligned}
$$

$$
\begin{aligned}
x_F &= x_C \sqcup x_H \\
x_G &= id(x_F) \\
x_K &= id(x_F) \\
x_H &= \mathbf{0}(x_F) \\
x_Q &= \mathbf{0}(x_F) \\
x_I &= \mathbf{1}(x_F) \\
x_J &= id(x_F).
\end{aligned}
$$



Example: Computing JVP values (x_N^*'s to be precise)

$$
\begin{aligned}
x_A &= 0\\
x_B &= \mathbf{0}(x_A)\\
x_C &= \mathbf{1}(x_A)\\
x_P &= \mathbf{1}(x_A)\\
x_D &= \mathbf{1}(x_A)\\
x_E &= \mathbf{1}(x_A)
\end{aligned}
$$

$$
\begin{aligned}
x_F &= x_C \sqcup x_H\\
x_G &= id(x_F)\\
x_K &= id(x_F)\\
x_H &= \mathbf{0}(x_F)\\
x_Q &= \mathbf{0}(x_F)\\
x_I &= \mathbf{1}(x_F)\\
x_J &= id(x_F).
\end{aligned}
$$

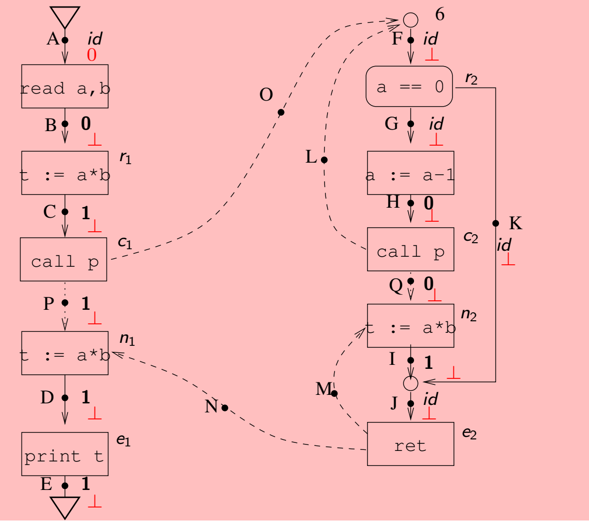


Fig shows initial (red) and final (blue) values.

Example: Computing JVP values (x_N^*'s to be precise)

$$
\begin{aligned}
x_A &= 0 \\
x_B &= \mathbf{0}(x_A) \\
x_C &= \mathbf{1}(x_A) \\
x_P &= \mathbf{1}(x_A) \\
x_D &= \mathbf{1}(x_A) \\
x_E &= \mathbf{1}(x_A)
\end{aligned}
$$

$$
\begin{aligned}
x_F &= x_C \sqcup x_H \\
x_G &= id(x_F) \\
x_K &= id(x_F) \\
x_H &= \mathbf{0}(x_F) \\
x_Q &= \mathbf{0}(x_F) \\
x_I &= \mathbf{1}(x_F) \\
x_J &= id(x_F).
\end{aligned}
$$

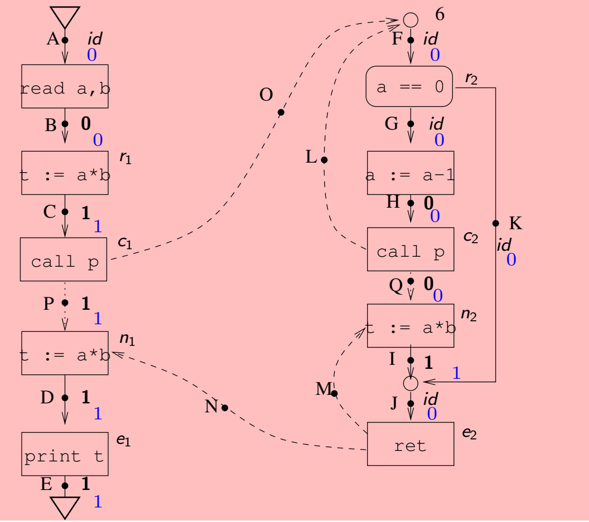


Fig shows initial (red) and final (blue) values.

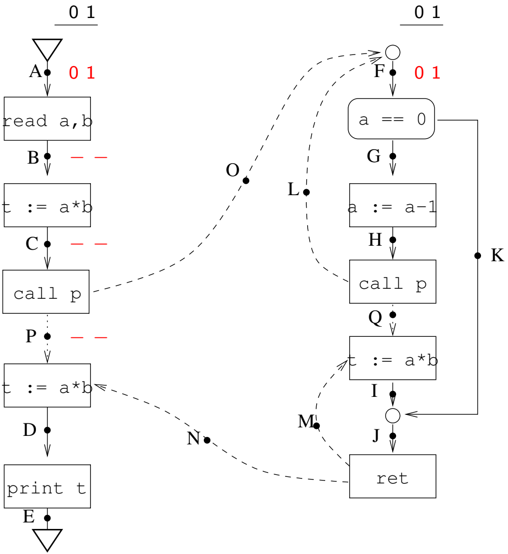
Summary of functional approach

- Uses a two step approach
  1. Compute $\phi_{r_p,N}$'s.
  2. Compute x_n's (JVP's) at each point.

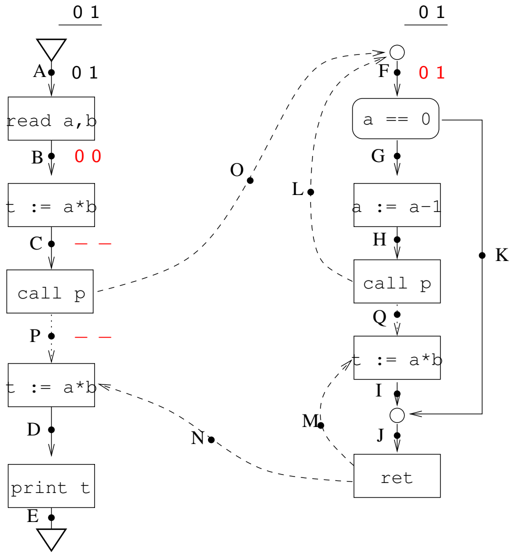
Summary of conditions: For each property (column heading), the conjunction of the ticked conditions (row headings) are sufficient to ensure the property.

	Termination	Least Sol of Eq(2) $\geq$ JVP	Least Sol of Eq(2)$=$ JVP
f_{MN}'s monotonic	$\surd$	$\surd$	
Finite underlying lattice	$\surd$		
f_{MN}'s distributive			$\surd$

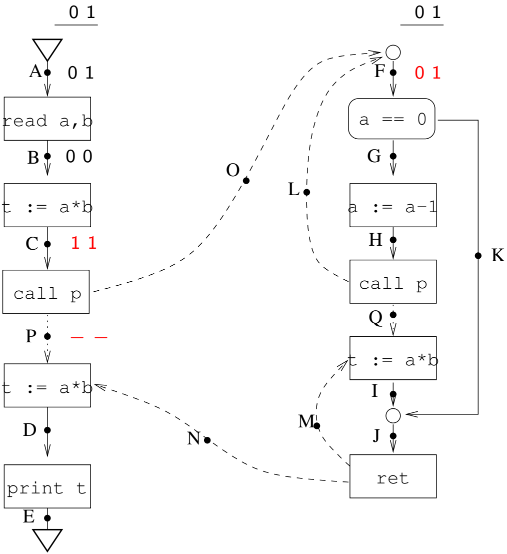
Viewing ϕ computation as a table



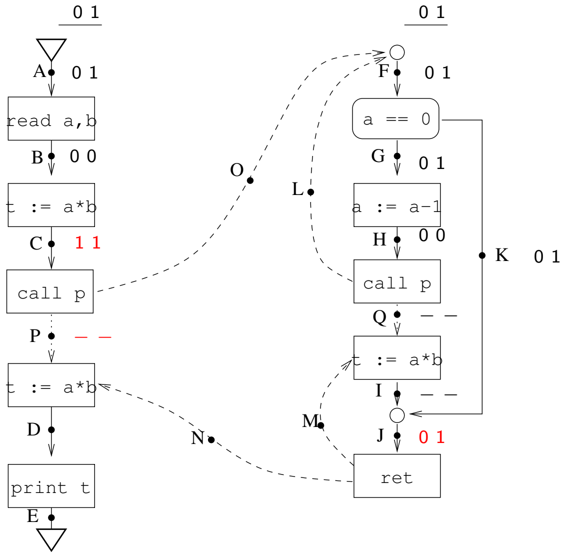
Viewing ϕ computation as a table



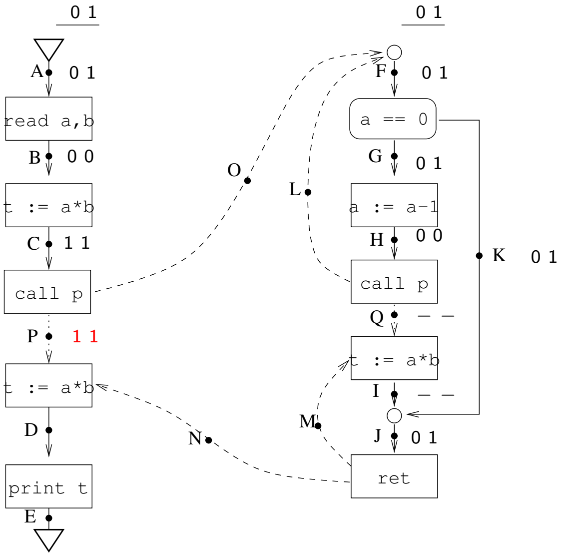
Viewing ϕ computation as a table



Viewing ϕ computation as a table



Viewing ϕ computation as a table



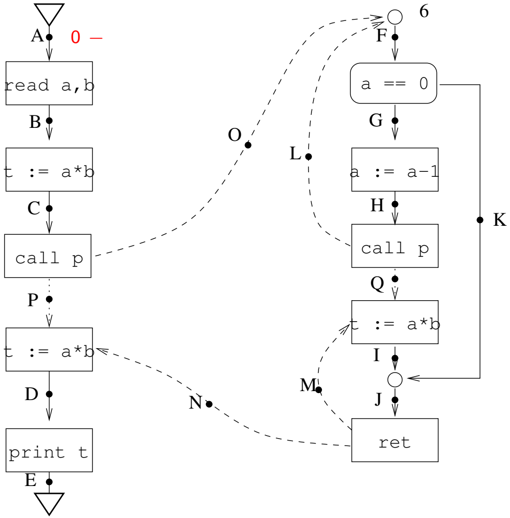
Iterative/Tabulation Approach

- Main idea: de-couple the propagation of function rows.
- Maintain a table of values representing the current value of $\phi_{r_p,N}$ for each program point N in procedure p.
- Expand column for data value d in procedure p only if d is reachable at r_p.
- Informally, at N in procedure p, the table has an entry $d \mapsto d'$ if we have seen
  1. valid paths ρ from r_1 to r_p with $\bigsqcup_\rho f_\rho(d_0) = d$, and
  2. valid and complete paths δ from r_p to N with $\bigsqcup_\delta f_\delta(d) = d'$.

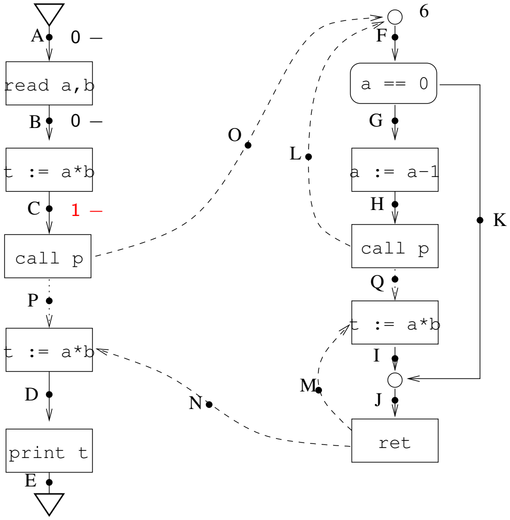
Iterative/Tabulation Approach

- Apply Kildall's algo with initial value of $d_0 \mapsto d_0$ at r_1.
- Propagating across a call to procedure p: value d is propagated to the column for d at root of p.
- Propagating across return nodes from procedure p: value d' in column for d is propagated to each column at a return site of a call to procedure p that has the value d in the preceding entry.

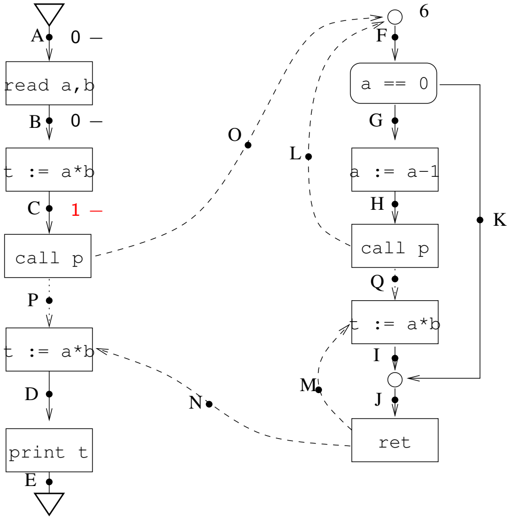
Example: Computing ϕ's iteratively: 1



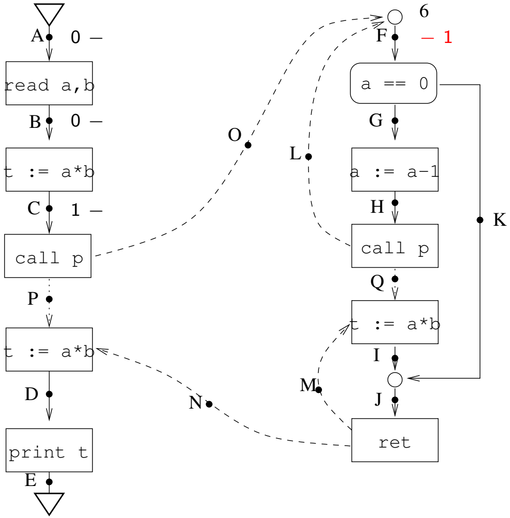
Example: Computing ϕ's iteratively: 2



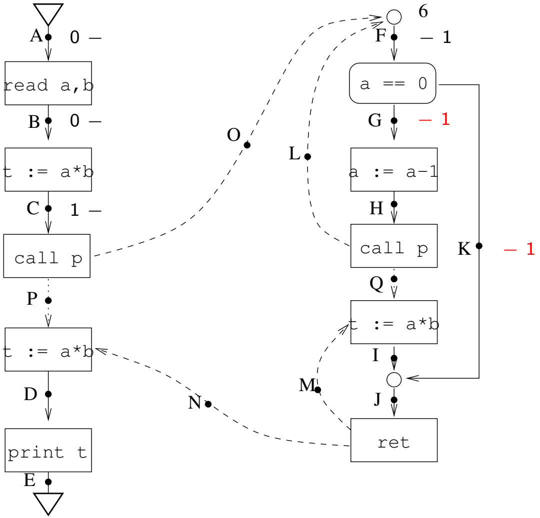
Example: Computing ϕ's iteratively: 3



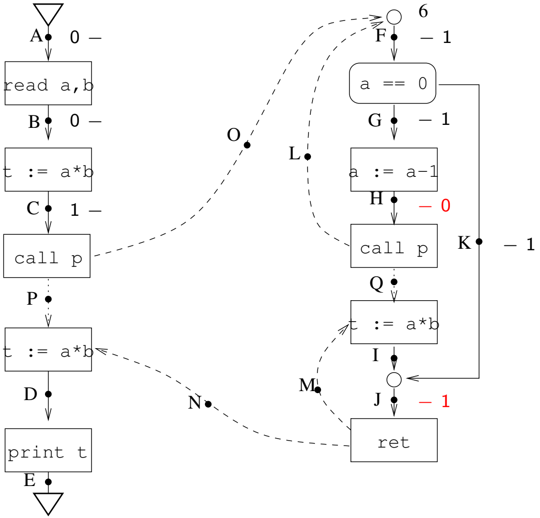
Example: Computing ϕ's iteratively: 4



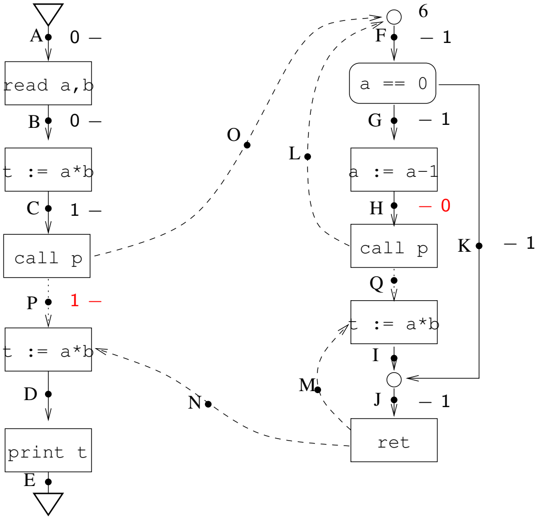
Example: Computing ϕ's iteratively: 5



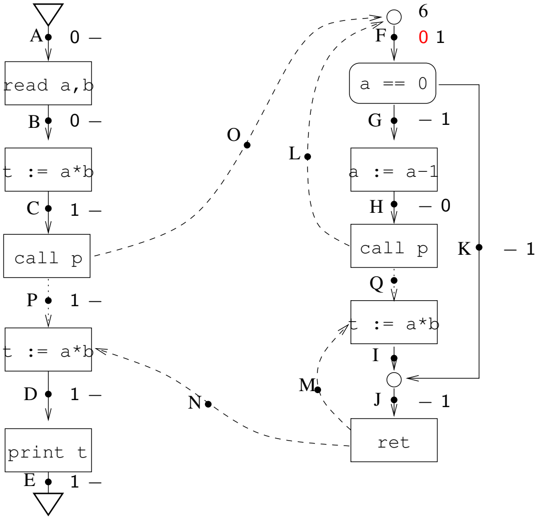
Example: Computing ϕ's iteratively: 6



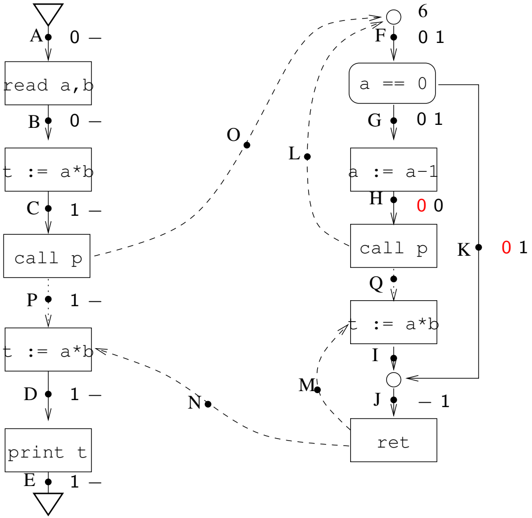
Example: Computing ϕ's iteratively: 7



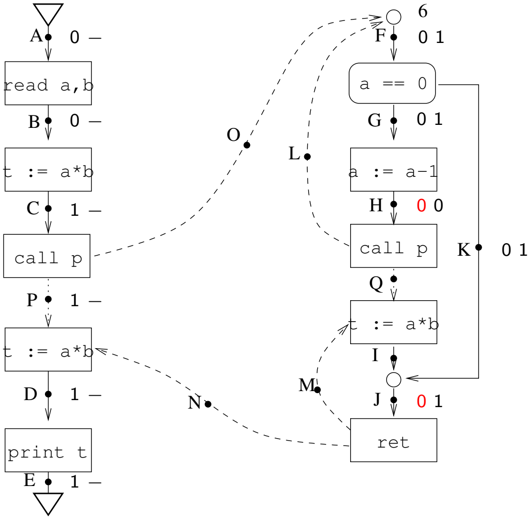
Example: Computing ϕ's iteratively: 8



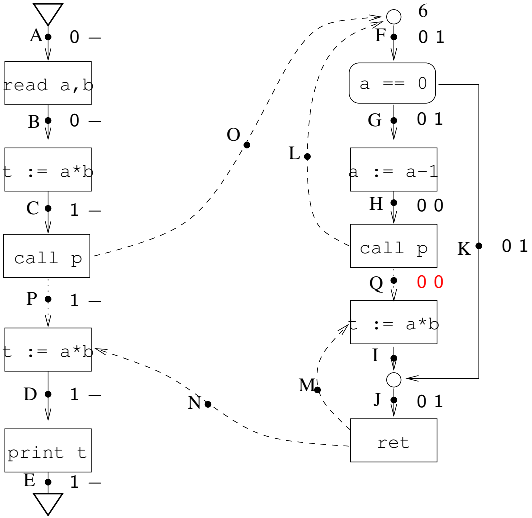
Example: Computing ϕ's iteratively: 9



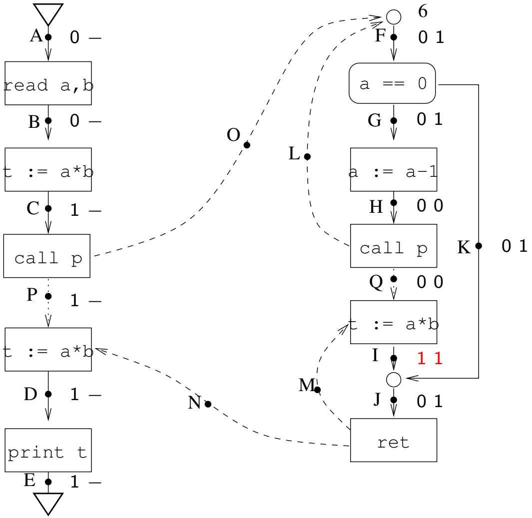
Example: Computing ϕ's iteratively: 10



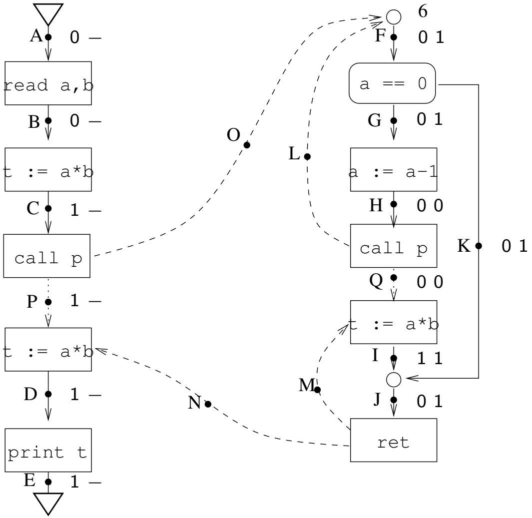
Example: Computing ϕ's iteratively: 11



Example: Computing ϕ's iteratively: 12

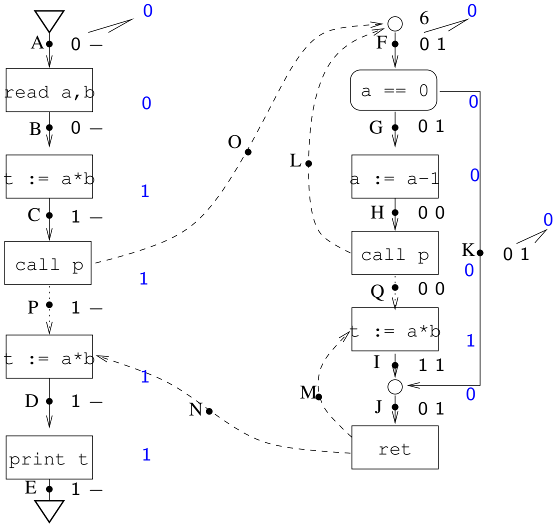


Example: Computing ϕ's iteratively: 13



Example: Finally compute x_N's from ϕ values

At each point N take join of reachable $\phi_{r_p,N}$ values.



Correctness of iterative algo

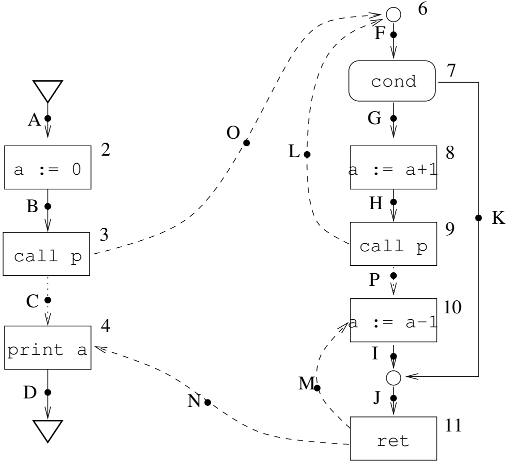
- Iterative algo terminates provided underlying lattice is finite.
- It computes the $y^*_{r_p,N}$'s (where $y^*_{r_p,N}$'s are the least solution to Eq (1)) "partially": If it maps d to $d' \neq \bot$ then $y^*_{r_p,N}(d) = d'$.
- The JVP values it gives (say z_N's) are such that

  $$\mathrm{JVP}_N \leq z_N \leq x^*_N$$

  (where x^*_N's are the solution to Eq (2')).
- If underlying transfer functions are distributive it computes $\phi_{r_p,N}$'s correctly (though partially), and the JVP values correctly.
- It thus computes an overapproximation of JVP for monotonic transfer functions, and exact JVP when transfer functions are distributive.

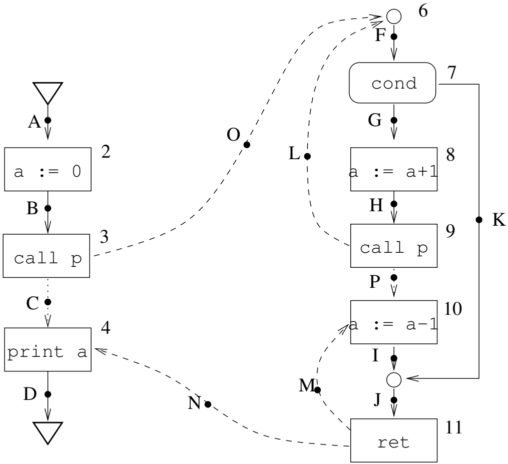
Exercise 1: Iterative algo

Run the iterative algo to do constant propagation analysis for the program below with initial value $\emptyset$. Assume here that "cond" is the condition "$a \leq 2$".



Exercise 2: Functional vs Iterative algo

Run the functional and iterative algos to do constant propagation analysis for the program below with initial value $\emptyset$:



Comparing functional vs iterative approach

- Functional algo can terminate even when underlying lattice is infinite, provided we can represent and compose/join functions "symbolically".
- Iterative is typically more efficient than functional since it only computes $\phi_{r_p,N}$'s for values reachable at start of procedure.