Interprocedural Analysis: Sharir-Pnueli's Call-strings Approach

Deepak D'Souza

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

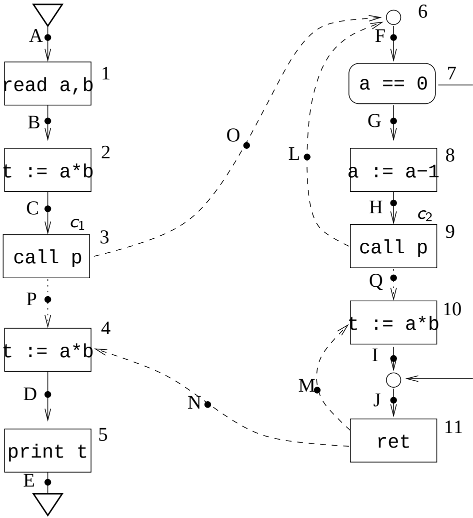
06 October 2010

Call strings approach

- For a given program P and analysis $((D, \leq), f_{MN}, d_0)$, the *join over all interprocedurally valid paths* (JVP) at point N is defined to be:

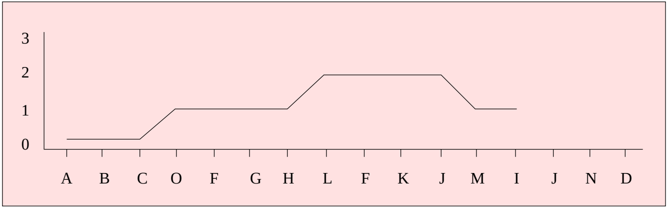
$$\bigsqcup_{\rho \in IVP(r_1, N)} f_\rho(d_0).$$

- Idea: collect data values that reach each point, tagged with call-string of associated path.
- This helps to say which values pass to a given return site.
- Now we can set up equations that capture JVP values.



Call-string along an interprocedurally valid path

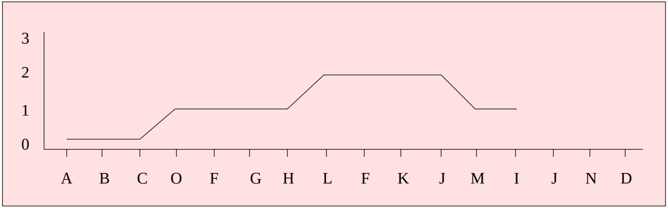
- Call-string associated with an *IVP* path ρ, denoted $CM(p)$, is the sequence of pending calls in ρ.
- A path ρ in $IVP(r_1, l)$ for example program:



- Associated call-string $CM(\rho)$ is c_1.

Call-string along an interprocedurally valid path

- Call-string associated with an IVP path ρ, denoted $CM(p)$, is the sequence of pending calls in ρ.
- A path ρ in $IVP(r_1, l)$ for example program:



- Associated call-string $CM(\rho)$ is c_1.
- For $\rho' = ABCOFGHLF$ $CM(\rho') = c_1 c_2$.
- Denote set of all call-strings for given program by Γ.

Tagging with call-strings

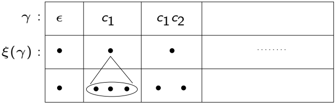
- Classify paths reaching N according to call-strings.
- For each call-string γ maintain data value

$$d = \bigsqcup_{\rho \in CM^{-1}(\gamma)} f_\rho(d_0).$$

- Thus elements of L^* are maps $\xi : \Gamma \to D$, and ordering $\xi_1 \leq \xi_2$ is pointwise extension of $\leq$ in D.
- Tagged JVP value: $\xi^*_N : \gamma \mapsto \bigsqcup_{\rho \in CM^{-1}(\gamma)} f_\rho(d_0)$.
- JVP value $d_N = \bigsqcup_{\gamma \in \Gamma} \xi^*_N(\gamma)$.

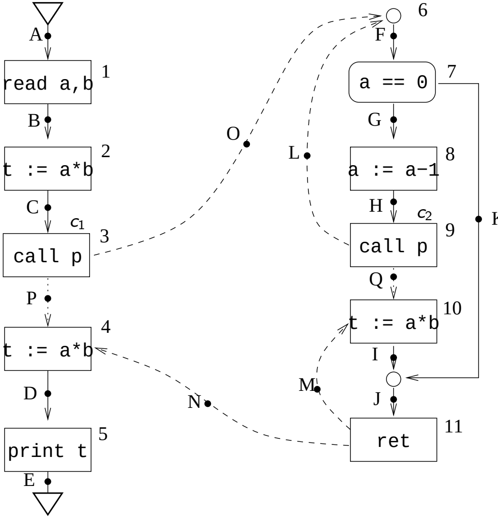
Example: Tagging

Eg: Path ABCOFGHLFKJ
has associated callstring $c_1 c_2$.



Tagged data values at J for
for availability of a*b analysis

γ :	ϵ	c_1	$c_1 c_2$	$c_1 c_2 c_2$	
$\xi(\gamma)$:	$\perp$	1	0	0	....



Data-flow analysis with tagged data values

- Let $D^* = \Gamma \to D$.
- Pointwise ordering on D^*
  - $\xi \leq' \xi'$ iff $\xi(\gamma) \leq \xi'(\gamma)$ for each call-string γ.
- $(D^*, \leq')$ is also a complete lattice.
- Initial value ξ_0 is given by

$$\xi_0(\gamma) = \begin{cases} d_0 & \text{if } \gamma = \epsilon \\ \perp & \text{otherwise.} \end{cases}$$

- Transfer functions for non call/ret nodes: $f^*_{MN} = \lambda\xi.f_{MN} \circ \xi$.
- Transfer functions f^*_{MN}'s are monotonic (distributive) if f_{MN}'s are monotonic (distributive).

Transfer functions f^*_{MN} by example

- (Non-call/ret node)

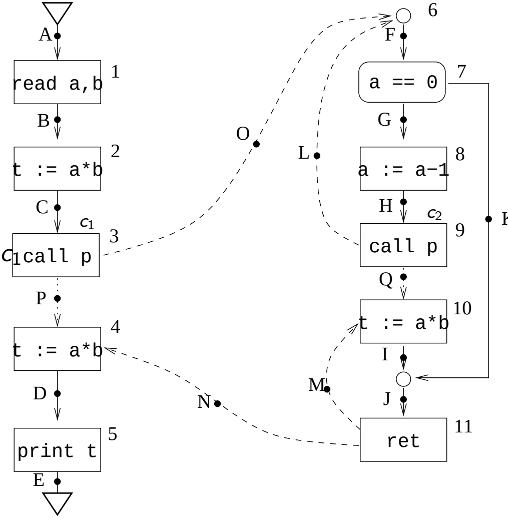
$$\xi_C = f_{BC} \circ \xi_B.$$

- (Call node)

$$\xi_F(\gamma) = \begin{cases} \xi_C(\gamma') & \text{if } \gamma = \gamma' \cdot c_1 \\ \perp & \text{otherwise} \end{cases}$$

- (Return site)

$$\xi_P(\gamma) = \xi_J(\gamma \cdot c_1).$$



Correctness claims

Claim

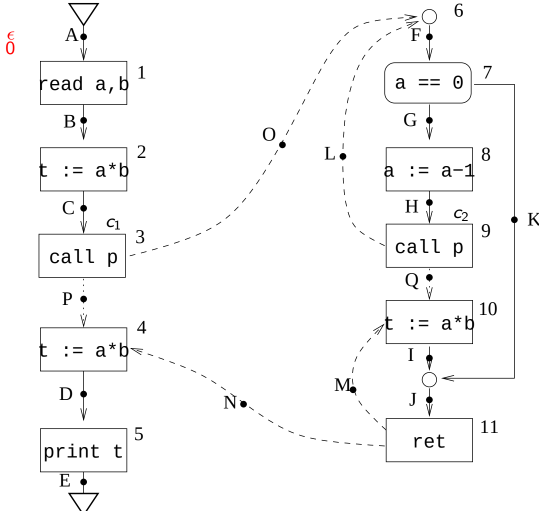
Let the LFP of the analysis $((D^*, \leq'), f^*_{MN}, \xi_0)$ be ξ^*. Then

$$x^*_N = \bigsqcup_{\gamma \in \Gamma} \xi^*_N(\gamma)$$

is an over-approximation of the JVP at N. When f_{MN}'s are distributive x^*_N coincides with JVN at N.

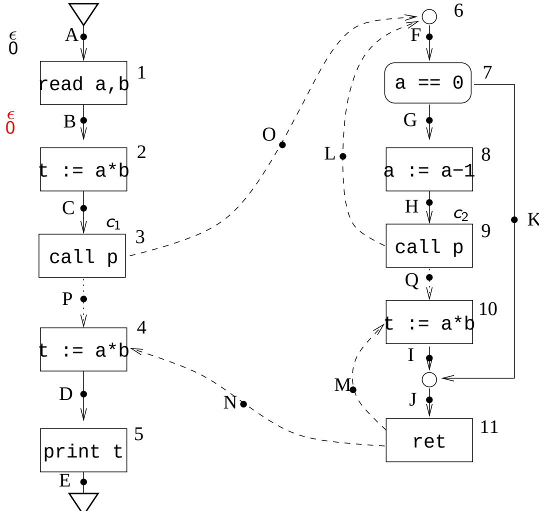
Exercise

Use Kildall's algo to compute the ξ table values for the example program, for $|\gamma| \leq 4$. Start with initial value $d_0 = 0$.



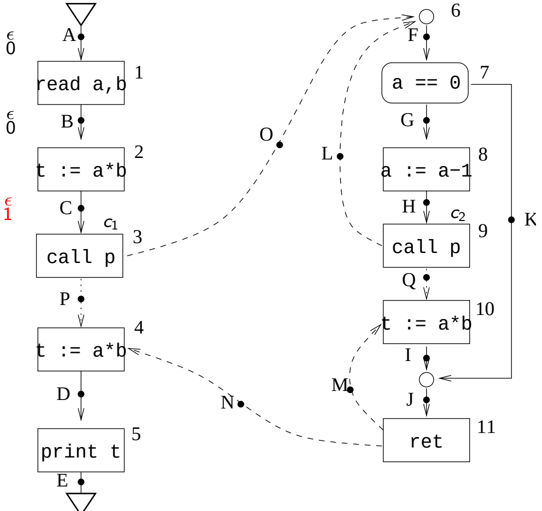
Exercise

Use Kildall's algo to compute the ξ table values for the example program, for $|\gamma| \leq 4$. Start with initial value $d_0 = 0$.



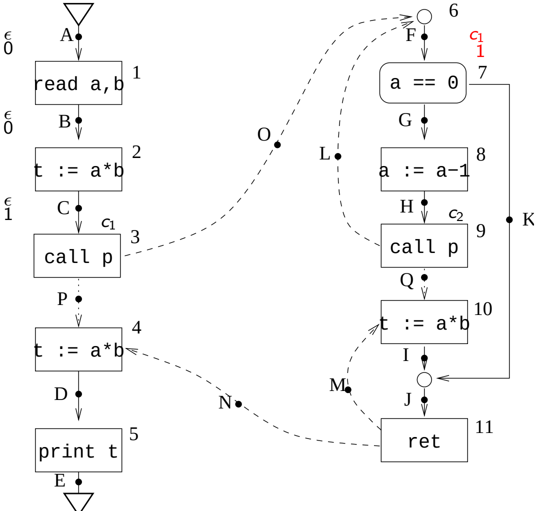
Exercise

Use Kildall's algo to compute the ξ table values for the example program, for $|\gamma| \leq 4$. Start with initial value $d_0 = 0$.



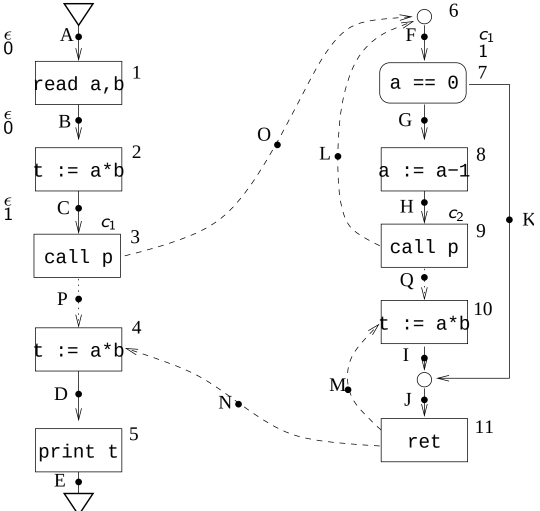
Exercise

Use Kildall's algo to compute the ξ table values for the example program, for $|\gamma| \leq 4$. Start with initial value $d_0 = 0$.



Exercise

Use Kildall's algo to compute the ξ table values for the example program, for $|\gamma| \leq 4$. Start with initial value $d_0 = 0$.



Convergence of iteration

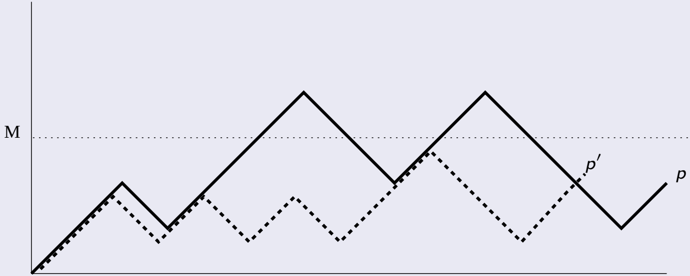
- Lattice $(D^*, \leq')$ is infinite for recursive programs.
- It is possible to bound the size of call strings Γ we need to consider.
- Let k be the number of call sites in P.

Convergence of iteration

Claim

For any path p with a prefix q such that $|CM(q)| > k|D|^2 = M$ there is a path p' with $|CM(q')| \leq M$ for each prefix q' of p', and $f_p(d_0) = f_{p'}(d_0)$.

Paths with bounded call-strings



Proof follows shortly.

Ensuring convergence

- Go over to a finite lattice.
- Consider only call strings of length $\leq M$ (Call this Γ_M).
- $IVP_{\Gamma_M}(r_1, N) =$ paths from r_1 to N such that for each prefix q, $CM(q) \leq M$.

Data-flow analysis for JVP over IVP_{Γ_M}

- (Non-call/ret node)

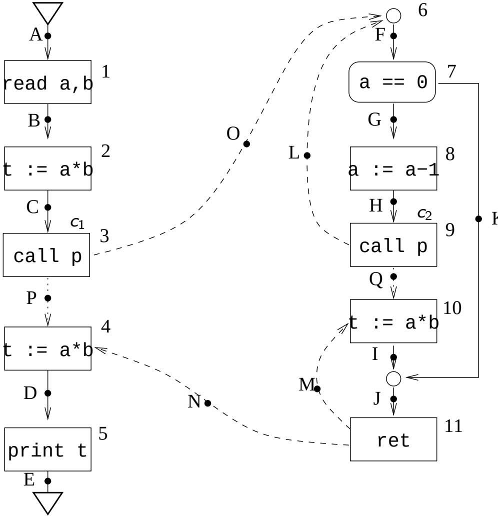
$$\xi_C = f_{BC} \circ \xi_B.$$

- (Call node)

$$\xi_F(\gamma) = \begin{cases} \xi_C(\gamma') & \text{if } \gamma = \gamma' \cdot c_1 \\ & \text{ and } \gamma \in \Gamma_M \\ \perp & \text{otherwise} \end{cases}$$

- (Return site)

$$\xi_P(\gamma) = \xi_J(\gamma \cdot c_1).$$



Bounding call-string size

Claim

For any path p in $IVP(r_1, N)$ such that $|CM(q)| > M = k|D|^2$ for some prefix q of p, there is a path p' in $IVP_{\Gamma_M}(r_1, N)$ with $f_{p'}(d_0) = f_p(d_0)$.

- Sufficient to prove:

Subclaim

For any path p in $IVP(r_1, N)$ with a prefix q such that $|CM(q)| > M$, we can produce a smaller path p' in $IVP(r_1, N)$ with $f_{p'}(d_0) = f_p(d_0)$.

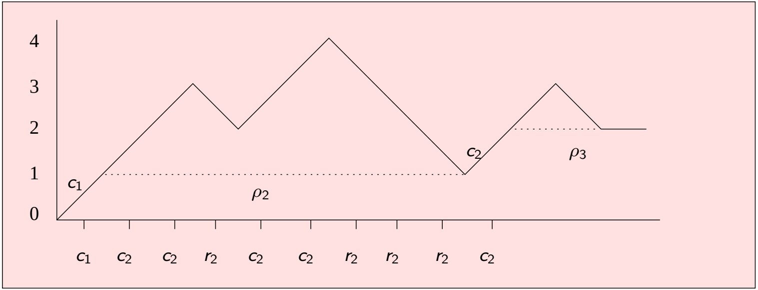
- ...since if $|p| \leq M$ then $p \in IVP_{\Gamma_M}$.

Proving subclaim: Path decomposition

A path ρ in $IVP(r_1, n)$ can be decomposed as

$$\rho_1 \| (c_1, r_{p_2}) \| \rho_2 \| (c_2, r_{p_3}) \| \sigma_3 \| \cdots \| (c_{j-1}, r_{p_j}) \| \rho_j.$$

where each ρ_i $(i < j)$is a valid and complete path from r_{p_i} to c_i, and ρ_j is a valid and complete path from r_{p_j} to n. Thus $c_1, \ldots, c_j$ are the unfinished calls at the end of ρ.



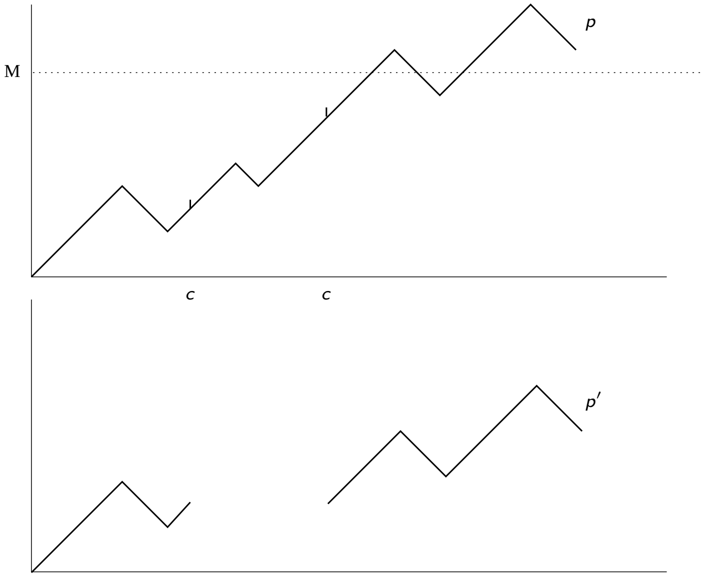
Proving subclaim

- Let p_0 be the first prefix of p where $|CM| > M$.
- Let decomposition of p_0 be

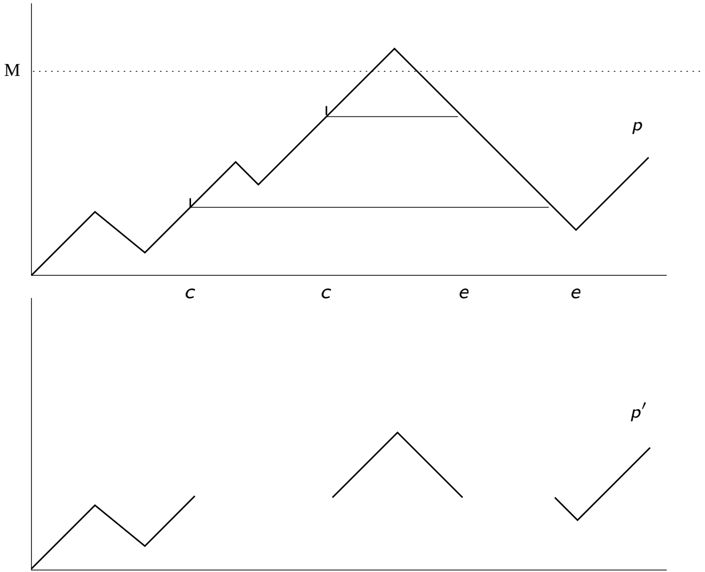
$$\rho_1 \| (c_1, r_{p_2}) \| \rho_2 \| (c_2, r_{p_3}) \| \sigma_3 \| \cdots \| (c_{j-1}, r_{p_j}) \| \rho_j .$$

- Tag each unfinished-call c_i in p_0 by $(c_i, f_{q \cdot c_i}(d_0), f_{q \cdot c_i q' e_{i+1}})$ where e_{i+1} is corresponding return of c_i in p.
- If no return for c_i in p tag with $(c, f_{q \cdot c_i}(d_0), \perp)$.
- Number of distinct such tags is $k \cdot |D|^2$.
- So there are two calls qc and $qcq'c$ with same tag values.

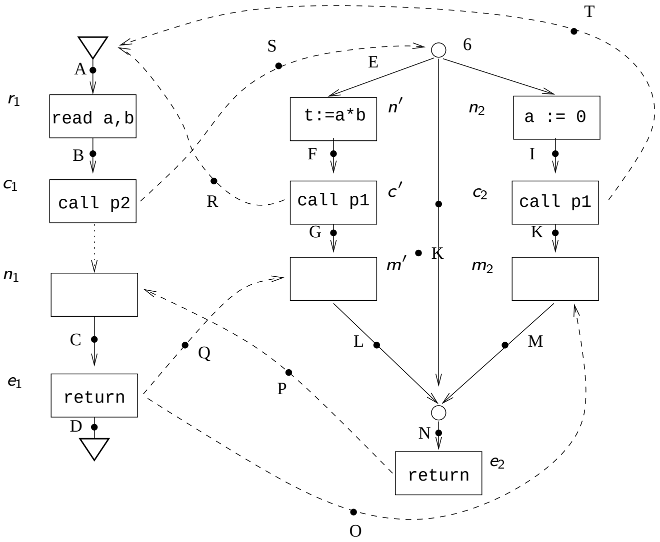
Proving subclaim – tag values are ⊥



Proving subclaim – tag values are not $\perp$



Example



T
S
6
E
A
r_1
read a,b
t:=a*b
n'
n_2
a := 0
B
F
I
c_1
call p2
R
call p1
c'
c_2
call p1
G
K
K
m'
m_2
n_1
C
L
M
Q
e_1
return
P
D
N
return
e_2
O