

# Two Approaches to Interprocedural Data Flow Analysis

Micha Sharir      Amir Pnueli

## Part one: The Functional Approach

# Intraprocedural analysis

procedure *main*

read a, b  
t := a \* b

call *p*

t := a \* b  
print t

stop

a\*b available?

procedure *p*

if a=0

T

a := a - 1

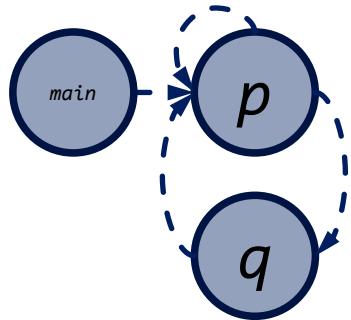
call *p*

t := a \* b

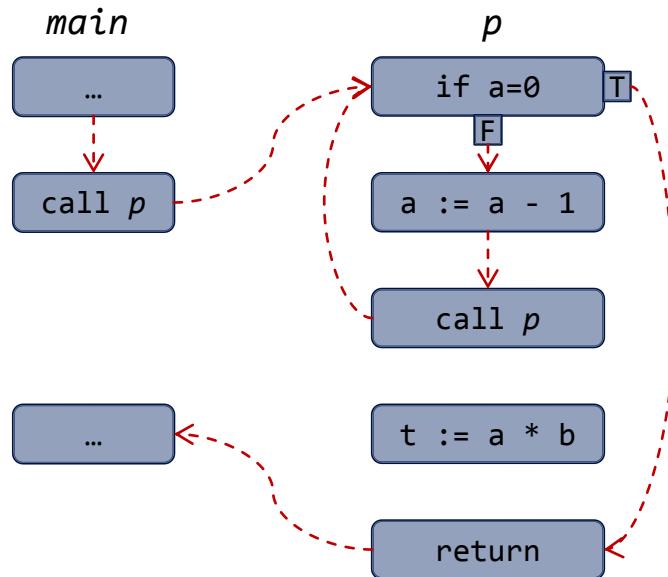
return

# Interprocedural challenges

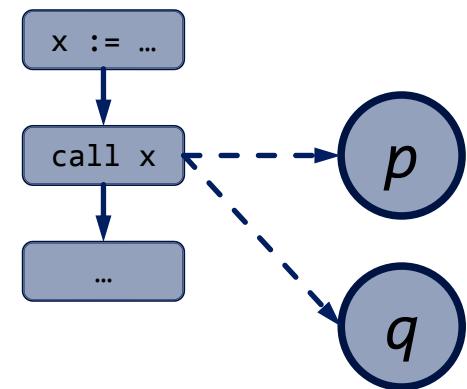
## Recursion



## Infeasible paths



## Function variables & Virtual functions



- Infinite paths
- Efficiency vs. Precision

- Filter invalid paths
- Precision and Efficiency

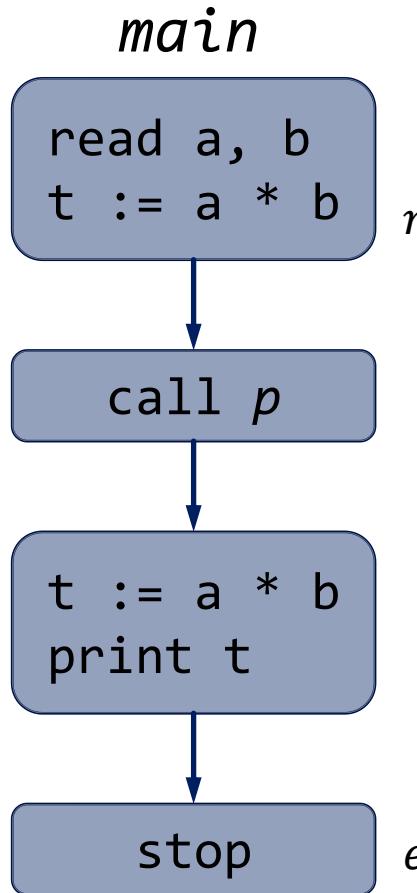
- No static call graph

# Outline

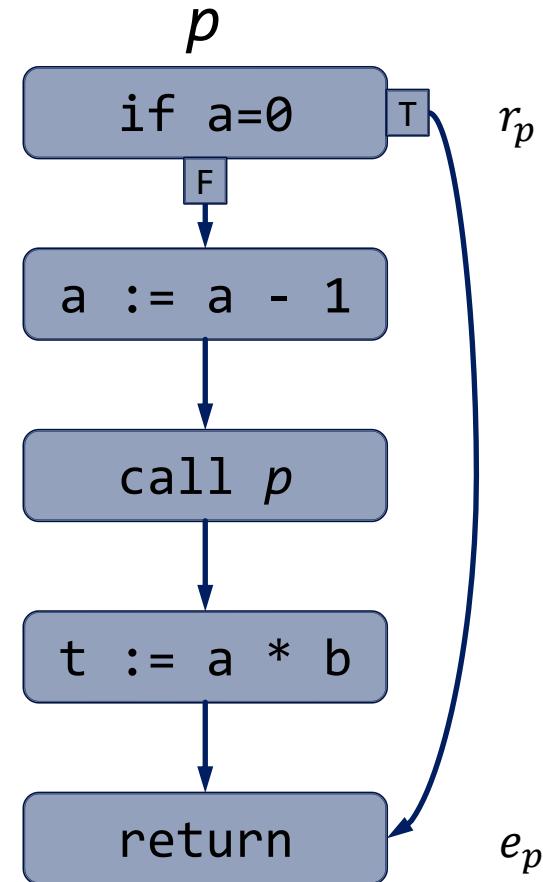
- Notation and Review
- Functional Approach
- Interprocedural MOP
- Pragmatic Considerations

# Notations

## Control Flow Graphs

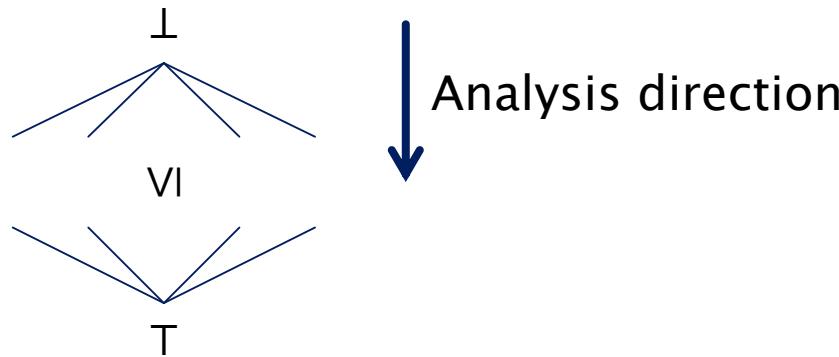


$e_{main}$



# Data Flow Frameworks

- $(L, F)$  is a data flow framework:  
 $L$  is a meet-semilattice
  - $\wedge$  = greatest lower bound
  - $T$  = smallest element (no information)
  - $\perp$  = largest element ("undefined")
  - *bounded* – No infinite descending chain



# Data Flow Frameworks

- $(L, F)$  is a data flow framework:
  - $F$  is a monotone space of transfer functions
    1. Closed under composition and meet  $(f \wedge g)(x) = f(x) \wedge g(x)$
    2. Contains  $\text{id}_L(x) = x$  and  $f_{\perp}(x) = \perp$
  - $F$  is *distributive* iff  $\forall f, x, y: f(x) \wedge f(y) = f(x \wedge y)$
- Restrict  $F$  to graph  $G = (N, E)$ :  
Smallest  $S \subseteq F$  such that  $\{f_{(m,n)} | (m, n) \in E\} \subseteq S$  and 1. and 2. hold

# Intraprocedural example

- Available expression framework for the single expression  $a * b$ :

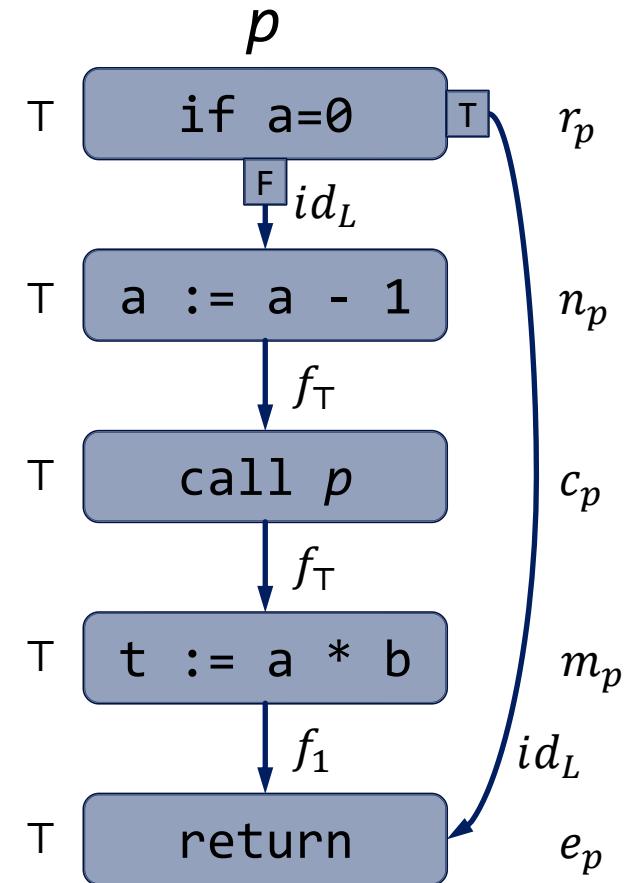
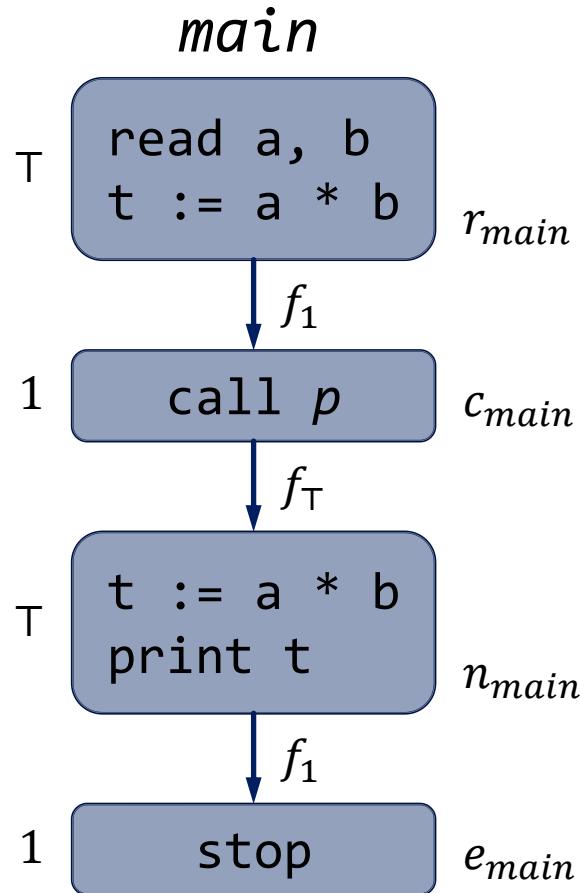
$$L = \{\top, 1, \perp\} \quad F = \{f_{\top}, f_1, id_L, f_{\perp}\}$$

$\top$ :  $a * b$  not available

$1$ :  $a * b$  available

$$f_{\top}(x) = \top, \quad f_1(x) = 1$$

# Intraprocedural example



# Intraprocedural equations

The data flow equations

$$x_r = \top$$

$$x_n = \bigwedge_{(m,n) \in E} f_{(m,n)}(x_m) \quad n \in N - \{r\}$$

approximate the meet-over-all paths (MOP) solution

$$y_n = \bigwedge \{f_p(\top) | p \in \text{path}_G(r, n)\} \quad n \in N$$

where  $f_{p=(n_1, \dots, n_k)} = f_{(n_{k-1}, n_k)} \circ \dots \circ f_{(n_1, n_2)}$

# Intraprocedural solutions

$F$  is *distributive*  $\Rightarrow$

The maximum fixed point solution  $x_n^* = y_n$

$F$  is *monotone*  $\Rightarrow x_n^* \leq y_n$

# Outline

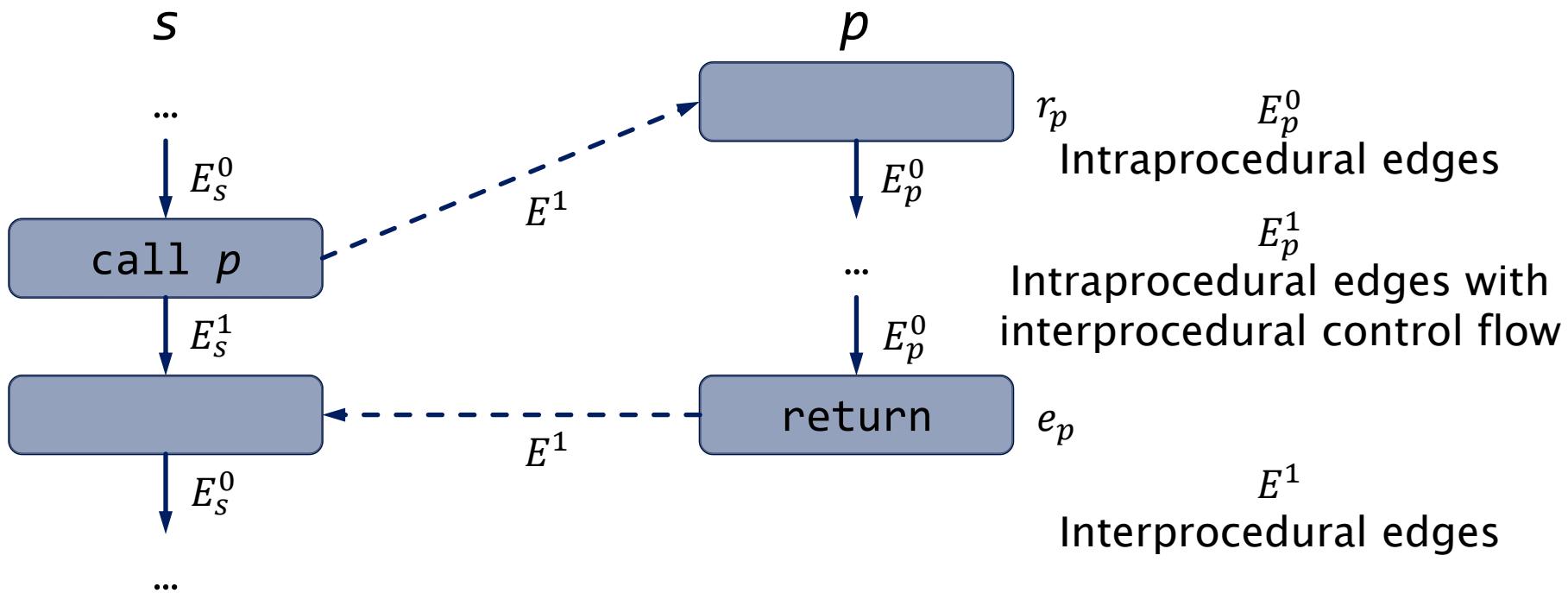
- Notation and Review
- Functional Approach
- Interprocedural MOP
- Pragmatic Considerations

# Interprocedural Graphs

Two representations:

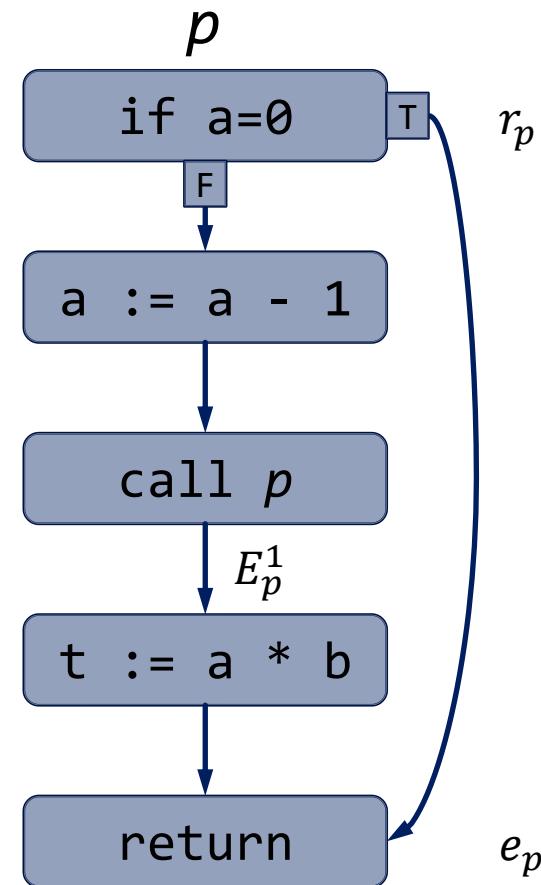
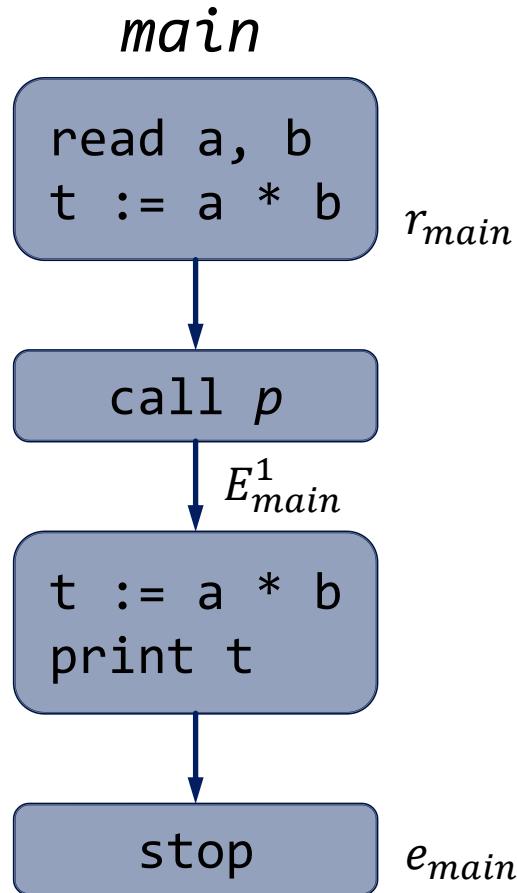
1.  $G = (\bigcup_p N_p, \bigcup_p E_p)$
2.  $G^* = (\bigcup_p N_p, E^*)$

$$E_p = E_p^0 \cup E_p^1 \quad E^0 = \bigcup_p E_p^0$$
$$E^* = E^0 \cup E^1$$



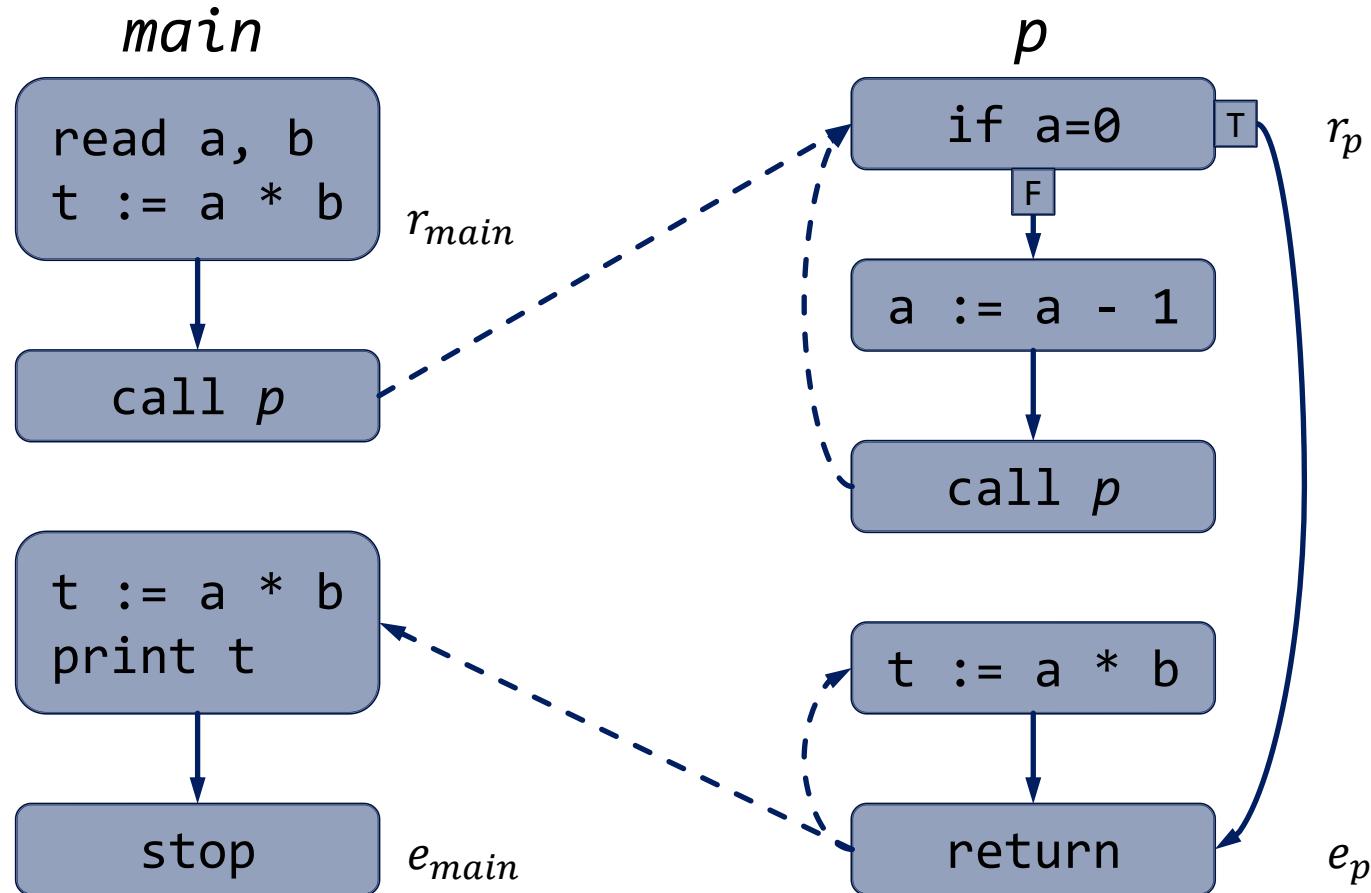
# Example

$G$



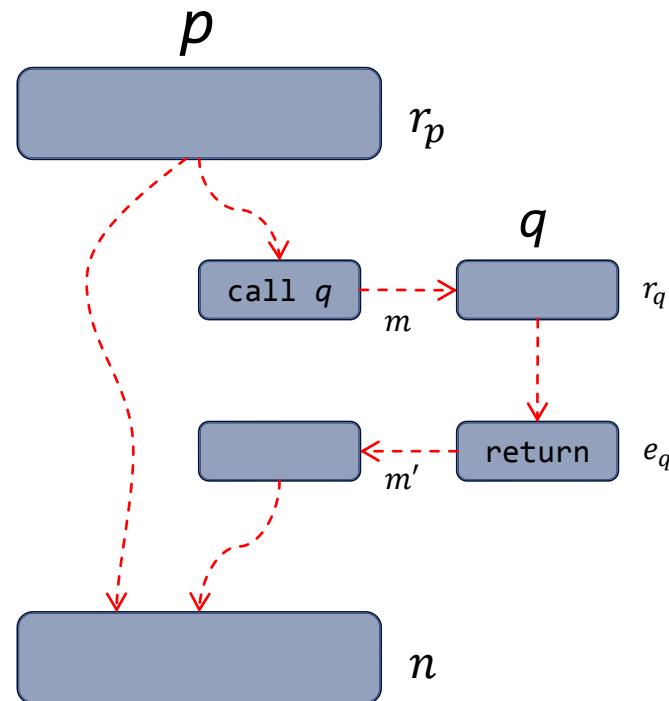
# Example

$G^*$

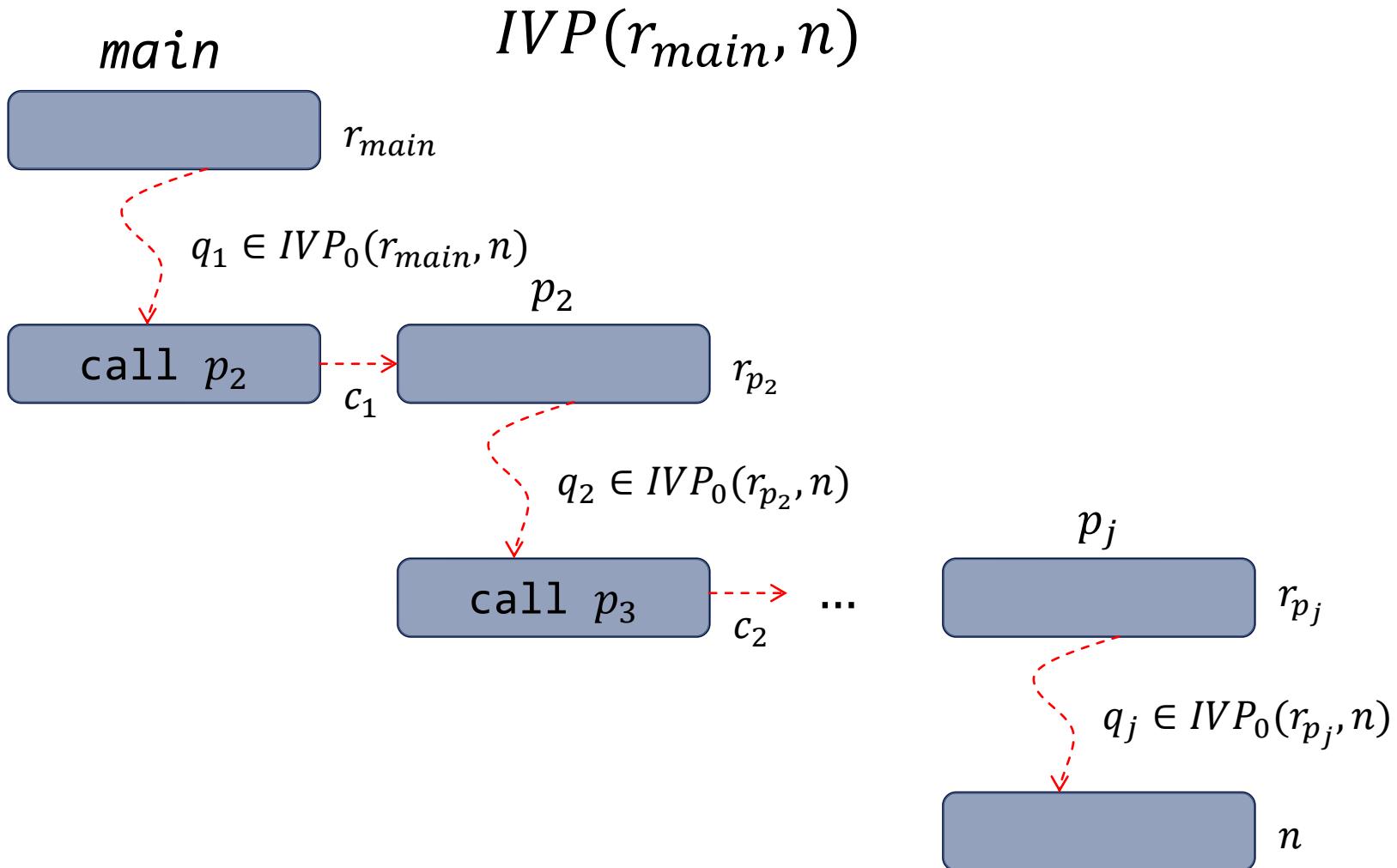


# Interprocedurally Valid paths

$$IVP_0(r_p, n)$$



# Interprocedurally Valid paths



# Path notations

- $p_1, p_2 \in path_{G^*}(r_q, n)$
- $p_1|_{E^1} :=$  Sequence of call & return edges in  $p_1$
- $p = p_1 \parallel p_2 :=$  Concatenation of  $p_1, p_2$

# Interprocedurally Valid paths

- $p \in path_{G^*}(r_q, n)$  is in  $IVP_0(r_q, n) \Leftrightarrow$

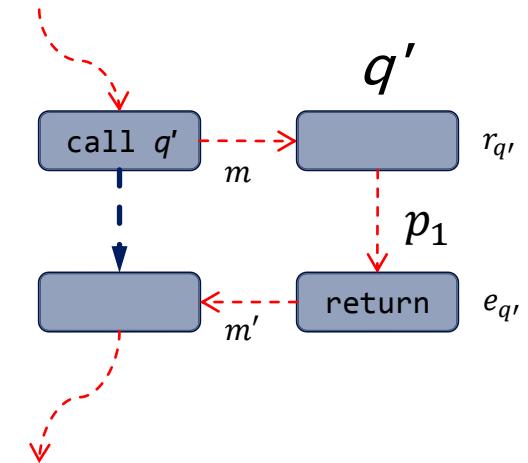
$p|_{E^1}$  is complete

defined as:

1.  $p|_{E^1} = \varepsilon$

2.  $p|_{E^1} = p_1 \parallel p_2$  and  $p_1, p_2$  are complete

3.  $p|_{E^1} = (m, r_{q'}) \parallel p_1 \parallel (e_{q'}, m')$  and  $p_1$  is complete



# Interprocedurally Valid paths

- $q \in IVP(r_{main}, n) :\Leftrightarrow$

$$q = q_1 \parallel (c_1, r_{p_2}) \parallel q_2 \parallel \cdots \parallel (c_{j-1}, r_{p_j}) \parallel q_j$$

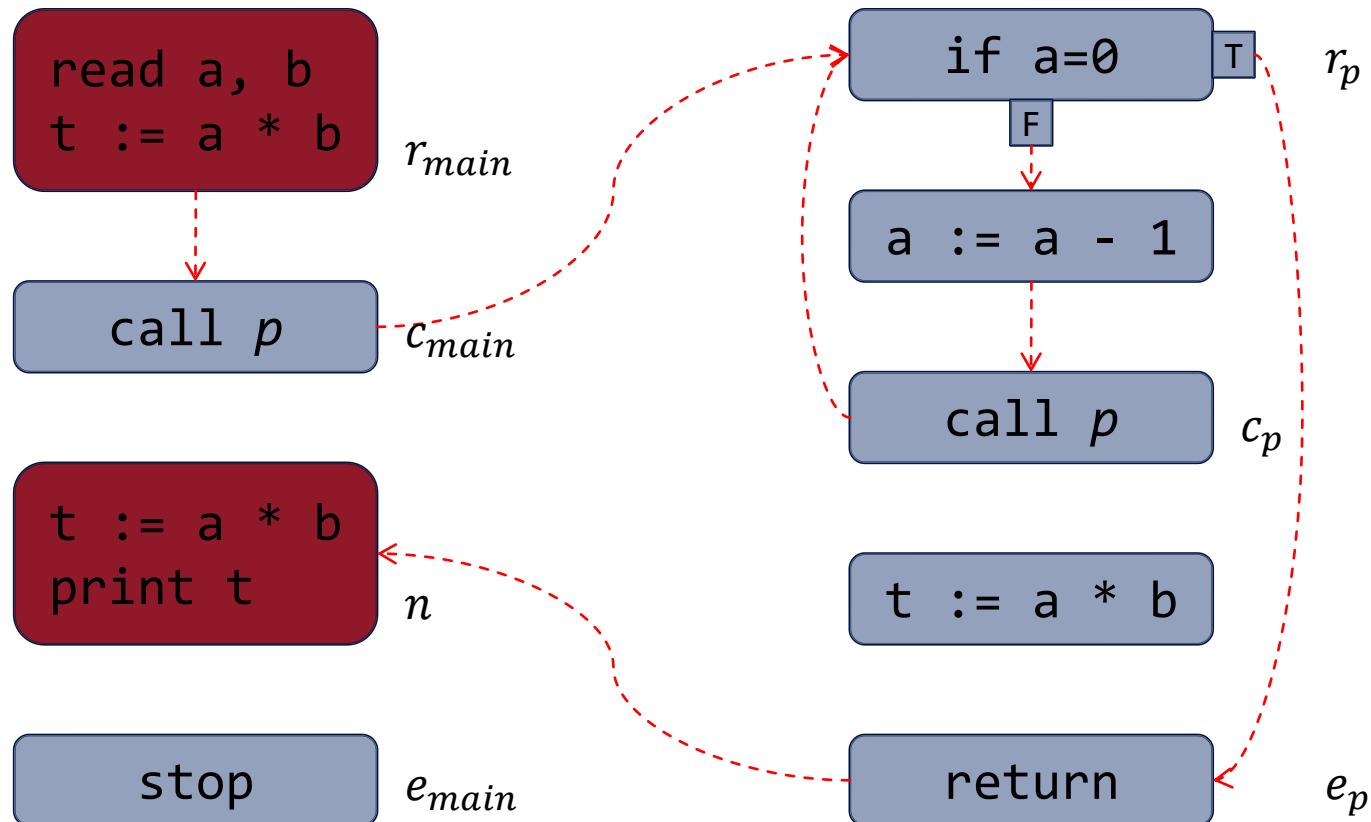
$$\forall i < j: q_i \in IVP_0(r_{p_i}, c_i) \text{ and } q_j \in IVP_0(r_{p_j}, n)$$

- Also called Path Decomposition

# Examples

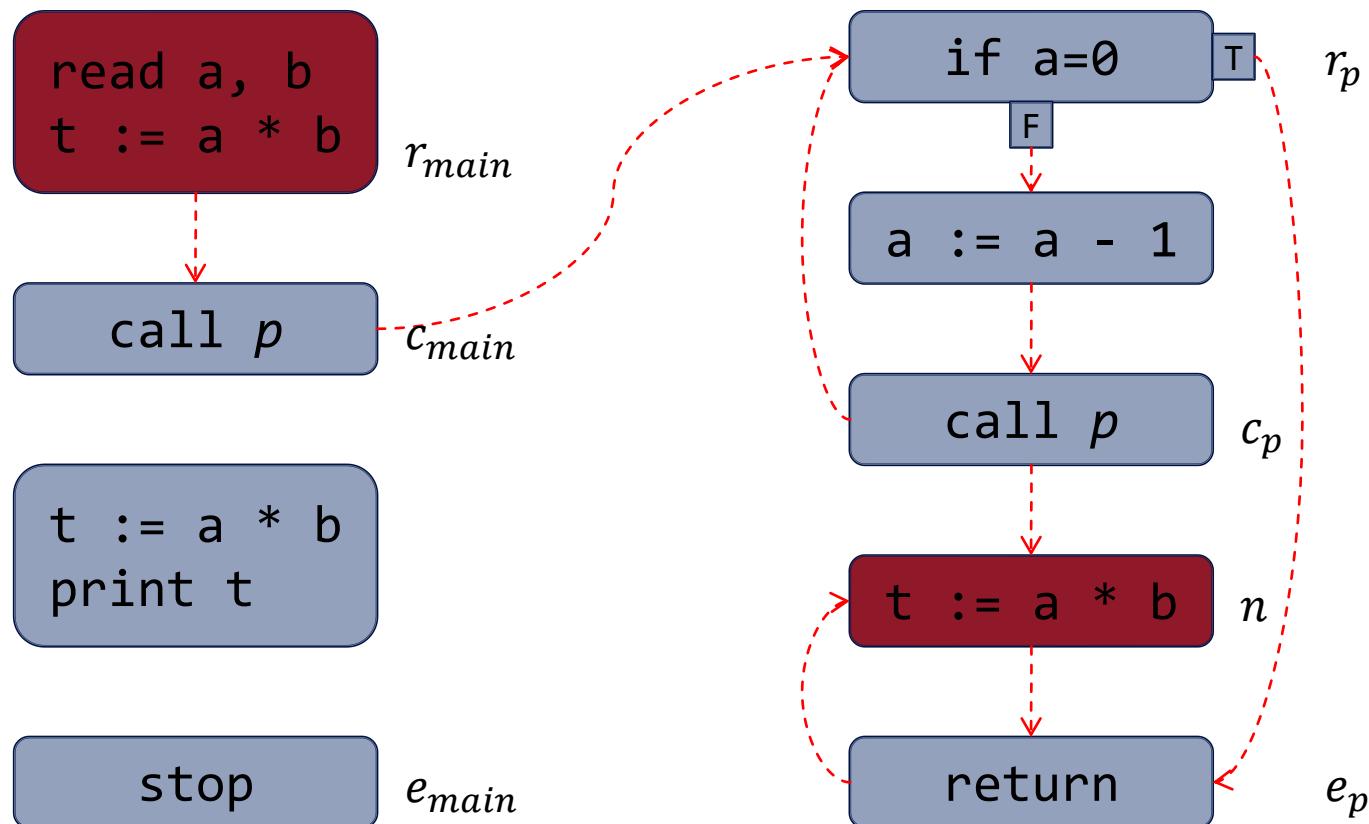
$G^*$

$$(c_{main}, r_p), (c_p, r_p), (e_p, n)$$

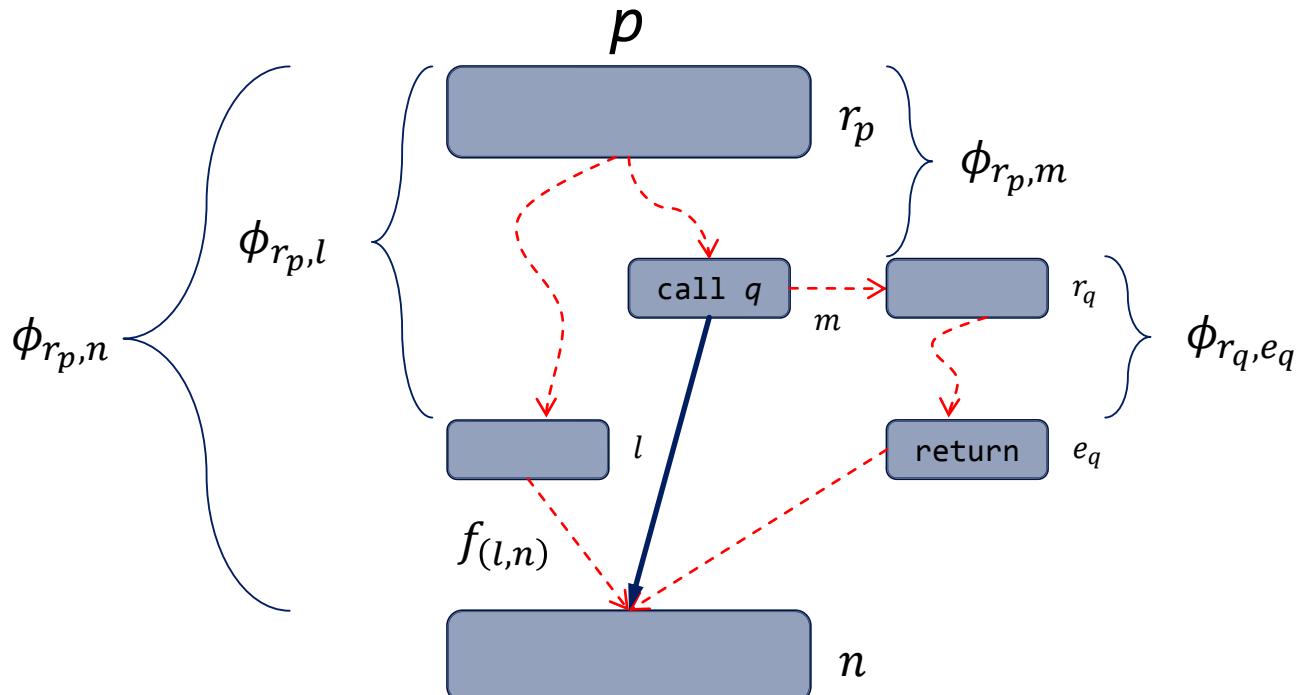


# Examples

$$G^* \quad (c_{main}, r_p), (c_p, r_p), (c_p, r_p), (e_p, n), (e_p, n)$$



# Functional approach



Information  $x$  at  $r_p$

is transformed to  $\phi_{r_p,n}(x)$  at  $n$

# Functional Approach equations

$$\phi_{r_p, r_p} = id_L$$

$$\phi_{r_p, n} = \bigwedge_{(m, n) \in E_p} (h_{(m, n)} \circ \phi_{r_p, m}) \quad n \in N_p - \{r_p\}$$

$$h_{(m, n)} = \begin{cases} f_{(m, n)} & (m, n) \in E_p^0 \\ \phi_{r_q, e_q} & (m, n) \in E_p^1, m \text{ calls } q \end{cases}$$

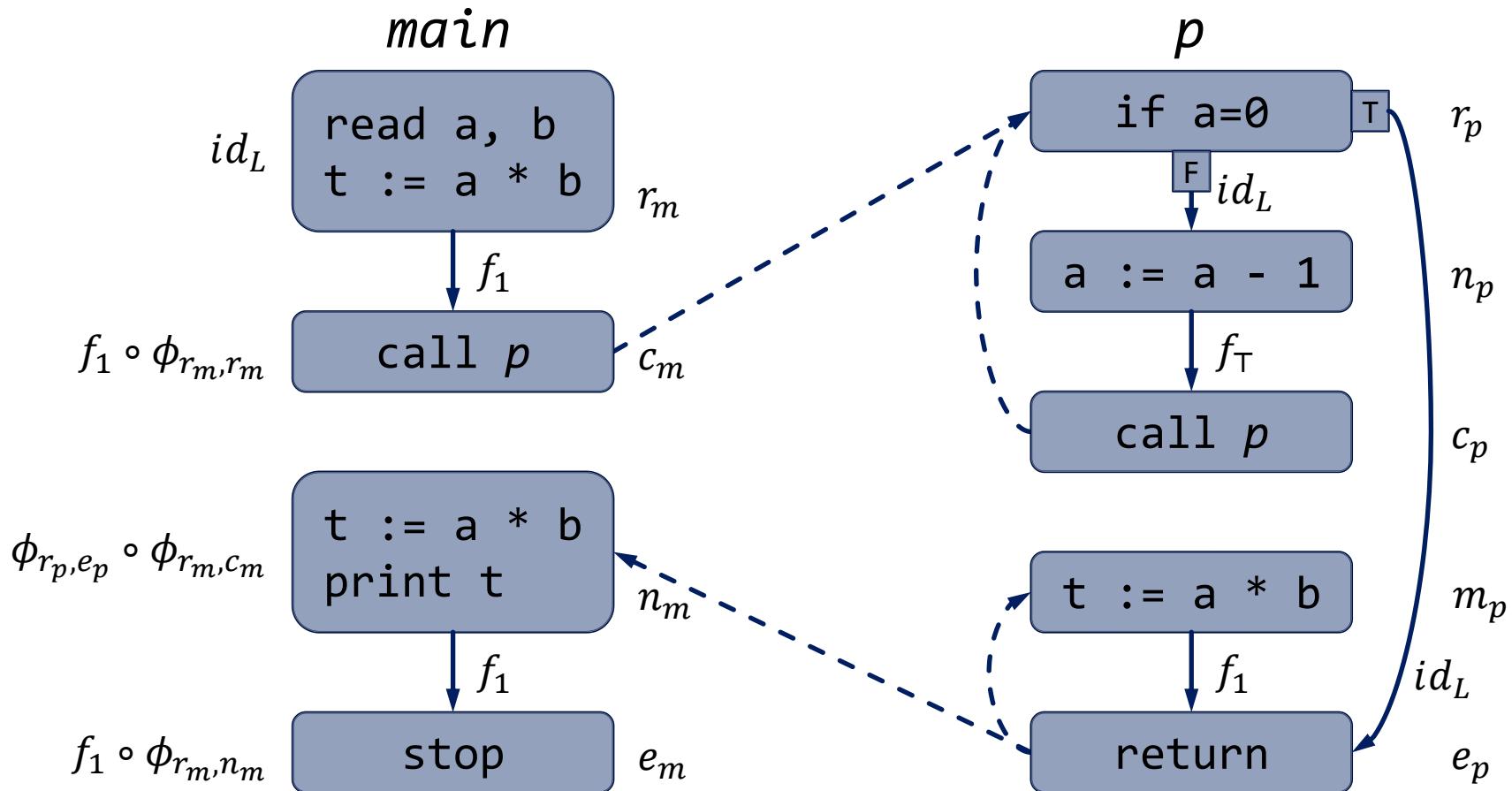
- Initialize the equations with Recursion implicitly encoded in equations

$$\phi_{r_p, r_p} = id_L$$

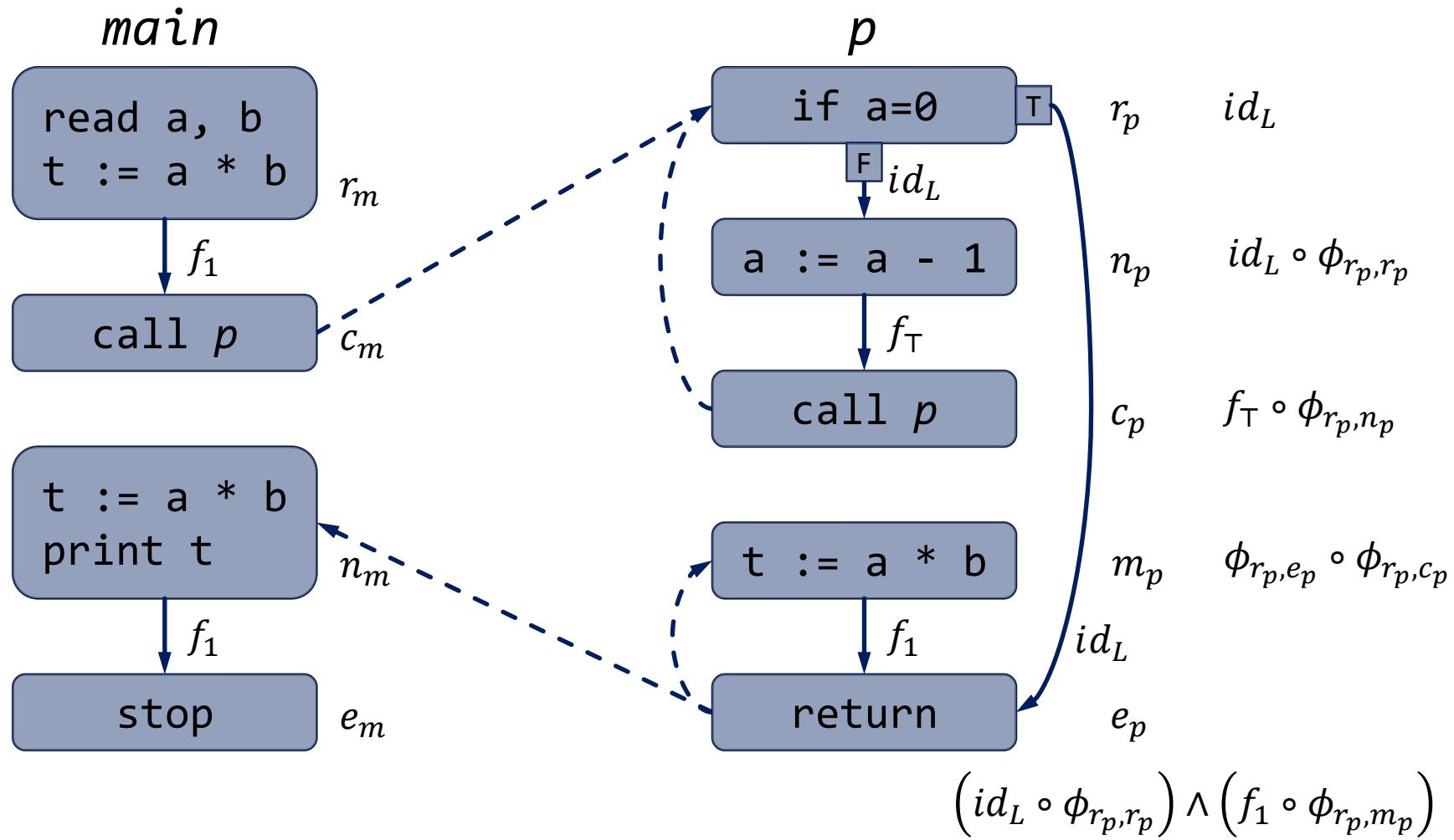
$$\phi_{r_p, n} = f_\perp \quad n \in N_p - \{r_p\}$$

- Compute the maximal fixed point

# Example



# Example



# Example

$$\phi_{r_m, r_m} = id_L$$

$$\phi_{r_m, c_m} = f_1 \circ \phi_{r_m, r_m}$$

$$\phi_{r_m, n_m} = \phi_{r_p, e_p} \circ \phi_{r_m, c_m}$$

$$\phi_{r_m, e_m} = f_1 \circ \phi_{r_m, n_m}$$

$$\phi_{r_p, r_p} = id_L$$

$$\phi_{r_p, n_p} = id_L \circ \phi_{r_p, r_p}$$

$$\phi_{r_p, c_p} = f_T \circ \phi_{r_p, n_p}$$

$$\phi_{r_p, m_p} = \phi_{r_p, e_p} \circ \phi_{r_p, c_p}$$

$$\phi_{r_p, e_p} = (id_L \circ \phi_{r_p, r_p}) \wedge (f_1 \circ \phi_{r_p, m_p})$$

Function	Initial value	Iteration 1	2	3
$\phi_{r_m, r_m}$	$id_L$	$id_L$	$id_L$	$id_L$
$\phi_{r_m, c_m}$	$f_\perp$	$f_1$	$f_1$	$f_1$
$\phi_{r_m, n_m}$	$f_\perp$	$f_\perp$	$f_1$	$f_1$
$\phi_{r_m, e_m}$	$f_\perp$	$f_1$	$f_1$	$f_1$
$\phi_{r_p, r_p}$	$id_L$	$id_L$	$id_L$	$id_L$
$\phi_{r_p, n_p}$	$f_\perp$	$id_L$	$id_L$	$id_L$
$\phi_{r_p, c_p}$	$f_\perp$	$f_T$	$f_T$	$f_T$
$\phi_{r_p, m_p}$	$f_\perp$	$f_\perp$	$f_T$	$f_T$
$\phi_{r_p, e_p}$	$f_\perp$	$id_L$	$id_L$	$id_L$

# Solution

$$x_{r_{main}} = \top$$

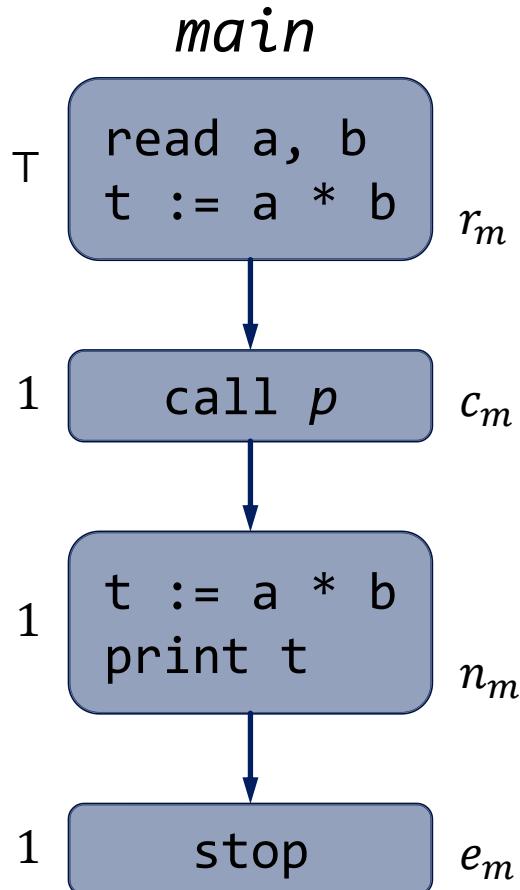
$$x_{r_p} = \bigwedge \left\{ \phi_{r_q, c} (x_{r_q}) \mid c \text{ calls } p \text{ in } q \right\}$$

- Compute the maximal fixed point iteratively

$$x_n = \phi_{r_p, n} (x_{r_p})$$

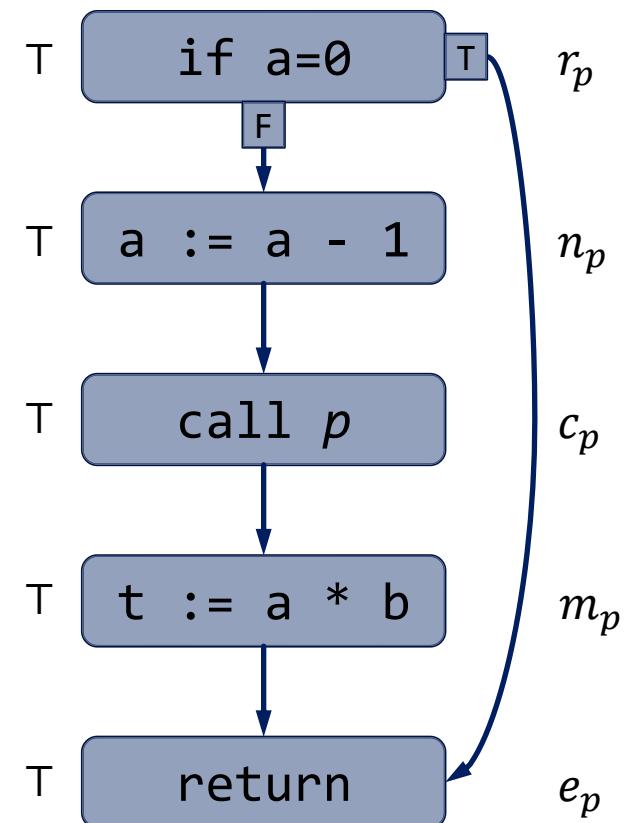
- Computes the solution for all other nodes

# Example (continued)



Functions
$\phi_{r_m,r_m} = id_L$
$\phi_{r_m,c_m} = f_1$
$\phi_{r_m,n_m} = f_1$
$\phi_{r_m,e_m} = f_1$
$\phi_{r_p,r_p} = id_L$
$\phi_{r_p,n_p} = id_L$
$\phi_{r_p,c_p} = f_T$
$\phi_{r_p,m_p} = f_T$
$\phi_{r_p,e_p} = id_L$

$$\begin{aligned}
 x_{r_p} &= \phi_{r_m,c_m}(x_{r_m}) \wedge \phi_{r_p,c_p}(x_{r_p}) \\
 &= f_1(x_{r_m}) \wedge f_T(x_{r_p})
 \end{aligned}$$



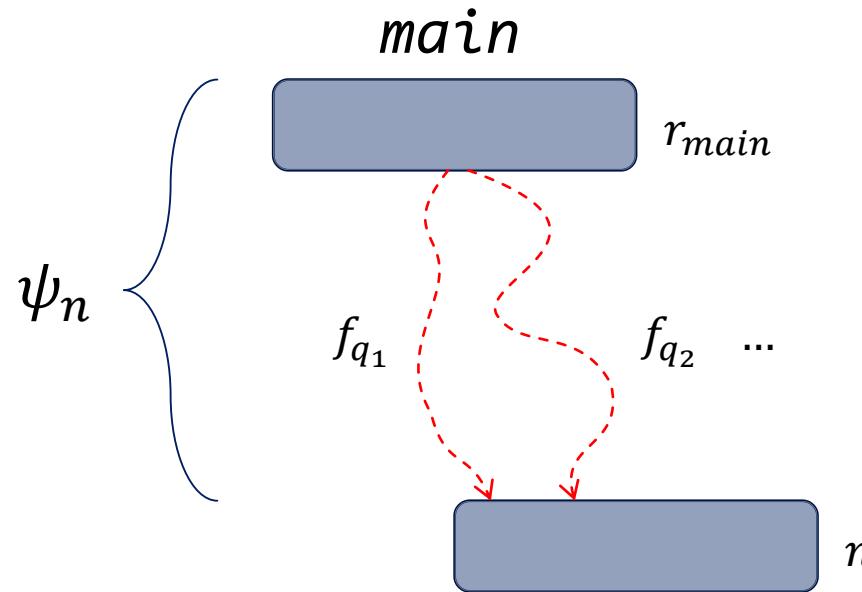
# Outline

- Notation and Review
- Functional Approach
- Interprocedural MOP
- Pragmatic Considerations

# Interprocedural MOP solution

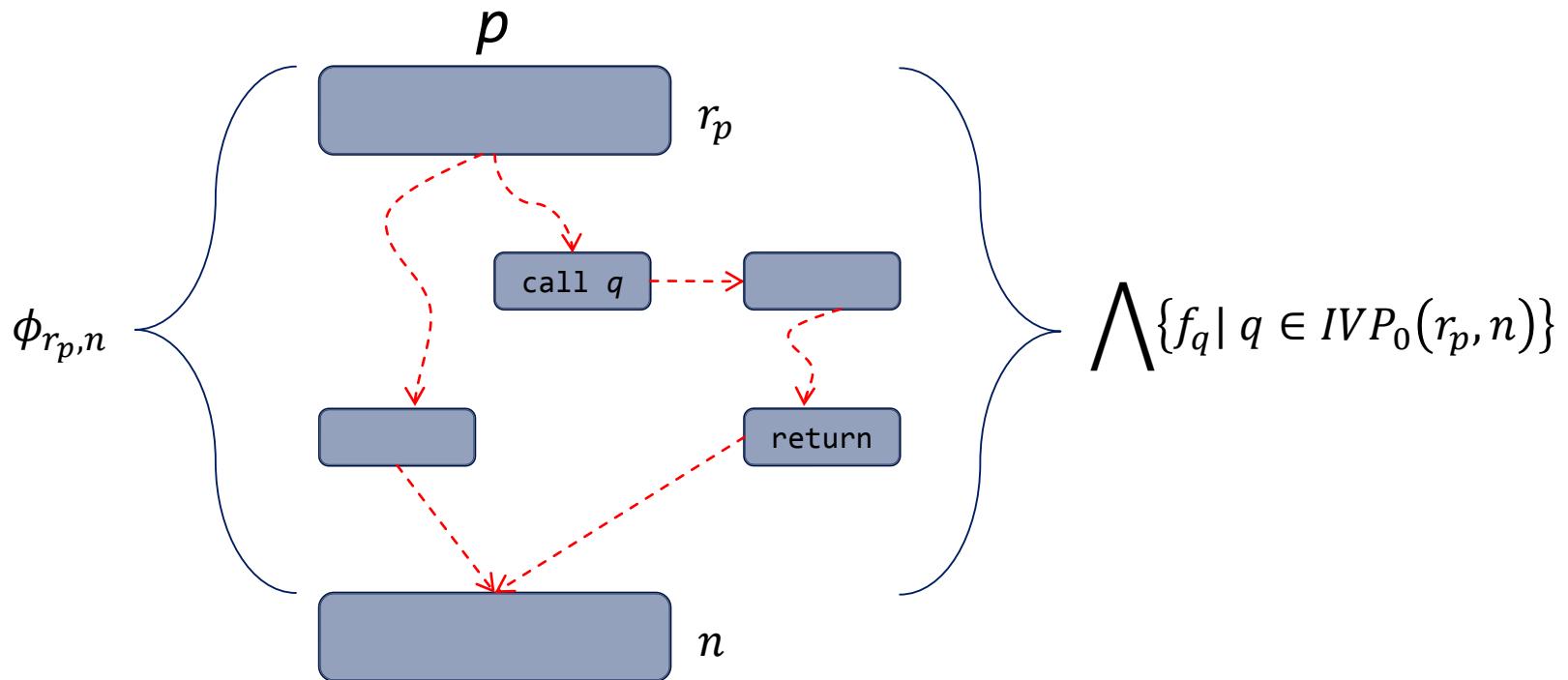
$$\psi_n = \bigwedge \{f_q \mid q \in IVP(r_{main}, n)\}$$

$$y_n = \psi_n(\top)$$



# $IVP_0$ Lemma

- $\phi_{r_p, n} = \bigwedge \{f_q \mid q \in IVP_0(r_p, n)\}$



# $IVP_0$ Lemma: Proof

- $\phi_{r_p,n} \geq \wedge\{f_q \mid q \in IVP_0(r_p, n)\}$
- By induction on  $\phi_{r_p,n}^i$ , the i-th approximation

For  $i = 0$ :

- If  $n = r_p$  then  $\phi_{r_p,n}^0 = id_L = f_\varepsilon \geq \wedge\{f_q \mid q \in IVP_0(r_p, r_p)\}$
- If  $n \neq r_p$  then  $\phi_{r_p,n}^0 = f_\perp \geq f \in F$

IH:  $\phi_{r_p,n}^i \geq \wedge\{f_q \mid q \in IVP_0(r_p, n)\}$

For  $i + 1$ : see blackboard

# $IVP_0$ Lemma: Proof

- $\Lambda\{f_q \mid q \in IVP_0(r_p, n)\} \geq \phi_{r_p, n}$
- Show:  $\forall q \in IVP_0(r_p, n)$ :  $f_q \geq \phi_{r_p, n}$  by induction over the length  $k$  of  $q$

For  $k = 0$ :

- Then  $f_q = \phi_{r_p, r_p} = id_L$

IH:  $\forall q \in IVP_0(r_p, n), |q| \leq k$ :  $f_q \geq \phi_{r_p, n}$

For  $k + 1$ : see blackboard

# MOP Rewritten

- $\psi_n = \bigwedge \{f_q \mid q \in IVP(r_{main}, n)\}$
- $\chi_n = \bigwedge \{\phi_{r_{p_j}, n} \circ \phi_{r_{p_{j-1}}, c_{j-1}} \circ \dots \circ \phi_{r_{main}, c_1} \mid c_i \text{ calls } p_{i+1}\}$
- Then  $\psi_n = \chi_n$  with  $IVP_0$  Lemma and Path decomposition
- Thus  $y_n = \psi_n(\top) = \chi_n(\top)$

# Equivalence to MOP

- $F$  distributive  $\Rightarrow y_n = x_n$
- Proof: Show that  $x_{r_p} = y_{r_p}$  then with

$$x_n = \phi_{r_p, n}(x_{r_p}) = \phi_{r_p, n}(y_{r_p}), y_n = \chi_n(\top) =$$

$$\wedge \left\{ \phi_{r_{p_j}, n} \circ \phi_{r_{p_{j-1}}, c_{j-1}} \circ \cdots \circ \phi_{r_{main}, c_1} \mid c_i \text{ calls } p_{i+1} \right\}(\top) =$$

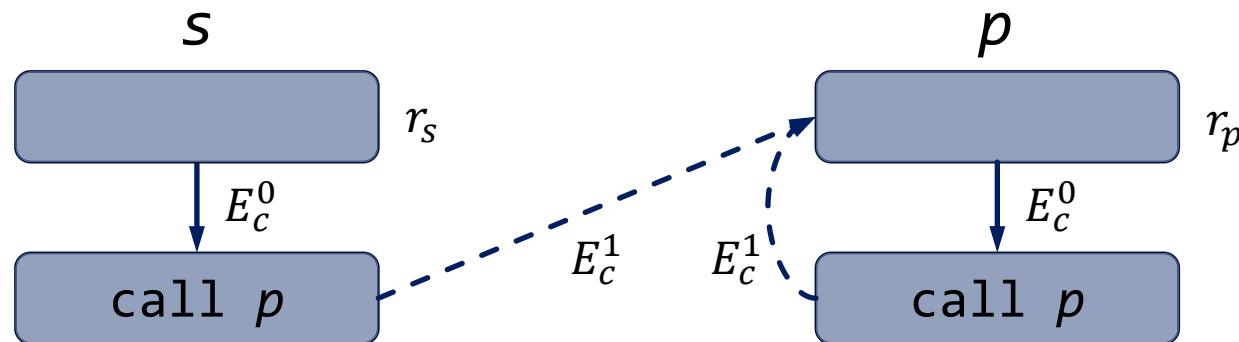
$$(\phi_{r_p, n} \circ \chi_{r_p})(\top) = \phi_{r_p, n}(y_{r_p})$$

it follows that  $x_n = y_n \forall n \in N^*$

# Equivalence to MOP

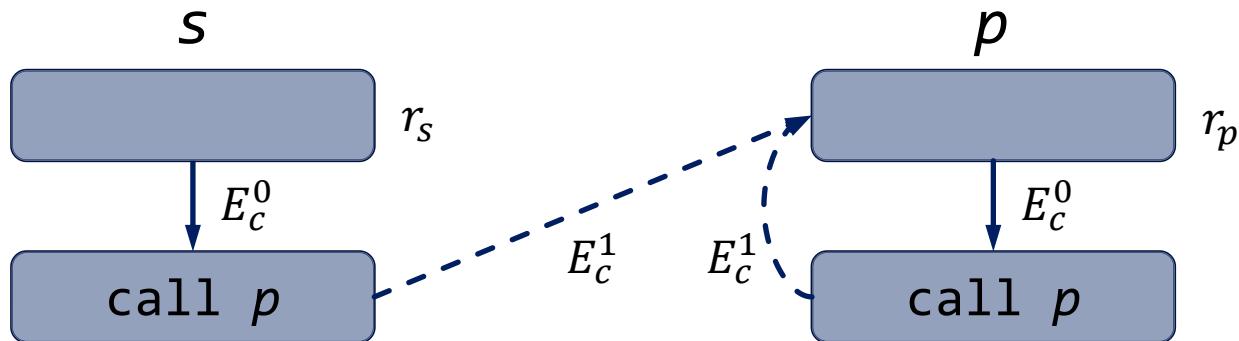
- New "Intraprocedural" Data Flow Problem:

- $G_c = (N_c, E_c) \quad E_c = E_c^0 \cup E_c^1$



- $g_{(m,n)} = \begin{cases} \phi_{m,n}, & (m,n) \in E_c^0 \\ id_L, & (m,n) \in E_c^1 \end{cases}$

# Equivalence to MOP



- $$g_{(m,n)} = \begin{cases} \phi_{m,n}, & (m,n) \in E_c^0 \\ id_L, & (m,n) \in E_c^1 \end{cases}$$

interprocedural

$$\begin{aligned} x_{r_{main}} &= \top \\ x_{r_p} &= \bigwedge \left\{ \phi_{r_q, c}(x_{r_q}) \mid c \text{ calls } p \text{ in } q \right\} \\ x_n &= \phi_{r_p, n}(x_{r_p}) \end{aligned}$$

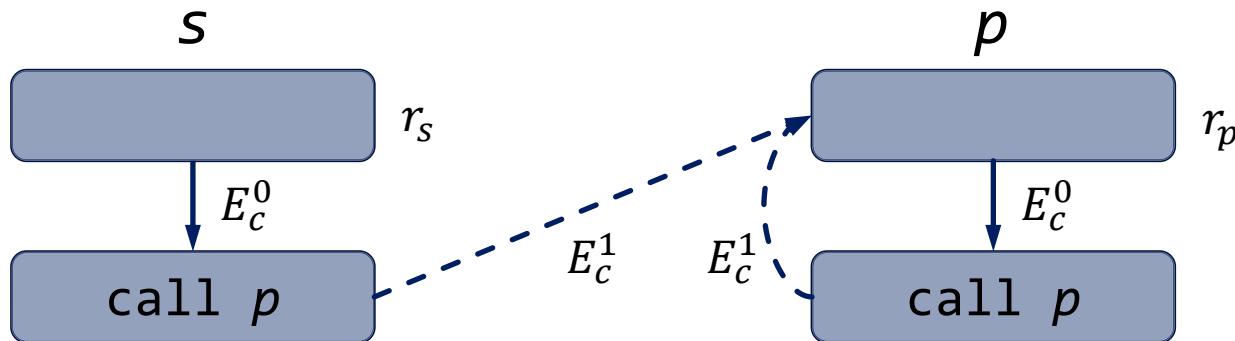
intraprocedural

$$\begin{aligned} x_r &= \top \\ x_n &= \bigwedge_{(m,n) \in E} g_{(m,n)}(x_m) \end{aligned}$$

# Equivalence to MOP

interprocedural	intraprocedural
$x_{r_{main}} = \top$ $x_{r_p} = \bigwedge \left\{ \phi_{r_q, c}(x_{r_q}) \mid c \text{ calls } p \text{ in } q \right\}$ $x_n = \phi_{r_p, n}(x_{r_p})$	$x_{r_{main}} = \top$ $x_{r_p} = \bigwedge_{(m, r_p) \in E_c^1} id_L(x_m)$ $= \bigwedge_{(m, r_p) \in E_c^1} x_m$ $= \bigwedge \left\{ \phi_{r_q, c}(x_{r_q}) \mid c \text{ calls } p \text{ in } q \right\}$  $x_c = \bigwedge_{(m, c) \in E_c^0} \phi_{m, c}(x_m)$ $= \phi_{r_p, c}(x_{r_p})$

# Equivalence to MOP



- $$g_{(m,n)} = \begin{cases} \phi_{m,n}, & (m,n) \in E_c^0 \\ id_L, & (m,n) \in E_c^1 \end{cases}$$

interprocedural

$$y_n = \bigwedge \{f_q \mid q \in IVP(r_{main}, n)\}(\top)$$

intraprocedural

$$y_n = \bigwedge \{f_p(\top) \mid p \in path_G(r, n)\}$$

- Follows from  $y_n = \bigwedge \{\phi_{r_{p_j}, n} \circ \dots \circ \phi_{r_{main}, c_1} \mid c_i \text{ calls } p_{i+1}\}(\top)$   
and by construction of  $G_c$

# Equivalence to MOP

$F$  is *distributive*  $\Rightarrow$

The maximum fixed point solution  $x_n^*$  of the functional approach = the interprocedural MOP solution  $y_n$

$F$  is *monotone*  $\Rightarrow x_n^* \leq y_n$

# Outline

- Notation and Review
- Functional Approach
- Interprocedural MOP
- Pragmatic Considerations

# Pragmatic problems

- Representation of  $\phi$ :

- Symbolic not always possible
- Explicit representation maybe not finite if  $L$  infinite:

main:	p:	$\phi_{r_p, e_p}(\{(A, 0)\})$
A := 0	if cond then	$\phi_{r_p, e_p}(\{(A, 1)\})$
call P	A := A + 1	$\phi_{r_p, e_p}(\{(A, 2)\})$
print A	call P	...
	A := A - 1	
	endif	
	return	

# Pragmatic problems

- Approach with symbolic representation may not converge when  $L$  infinite and  $F$  unbounded:  
 $\forall k \geq 0$ : Need  $k$  iterations to show  $\phi_{r_p, e_p}(\{(A, -k)\}) = \emptyset$

main:                    p:

```
A := 0                if cond then
call P                A := A + 2 + (A<0)?-1:1
print A              call P
                      A := A - 1
endif
return
```

# Practical result

- If  $(L, F)$  distributive and  $L$  finite then the

iterative solution of

$$\phi_{r_p, r_p} = id_L$$

$$\phi_{r_p, n} = \bigwedge_{(m, n) \in E_p} (h_{(m, n)} \circ \phi_{r_p, m})$$

$$x_{r_{main}} = \top$$

$$x_{r_p} = \bigwedge \left\{ \phi_{r_q, c} (x_{r_q}) \mid c \text{ calls } p \text{ in } q \right\}$$

$$x_n = \phi_{r_p, n} (x_{r_p})$$

converges and results in the interprocedural MOP solution

# Algorithm

- Assume  $L$  finite and no symbolic representation available
- Compute  $\phi_{r_p, n}$  only for necessary values

$W := \{(r_{main}, T)\}; \text{ PHI}(r_{main}, T) = T$

*while*  $W \neq \emptyset$

*remove some*  $(n, x)$  *from*  $W$

*update PHIs and*  $W$

$x_n = \Lambda_{a \in L} \text{ PHI}(n, a)$

# Algorithm: updating PHIs and $W$

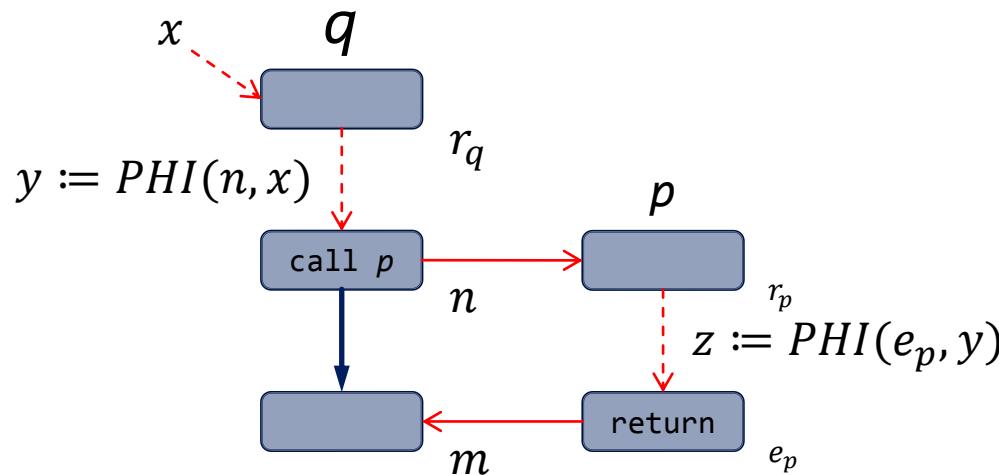
*propagate*( $x, z, m$ ):

$\text{PHI}(m, x) := \text{PHI}(m, x) \wedge z;$

*if PHI changed:*  $W := W \cup \{(m, x)\}$

# Algorithm: updating PHIs and $W$

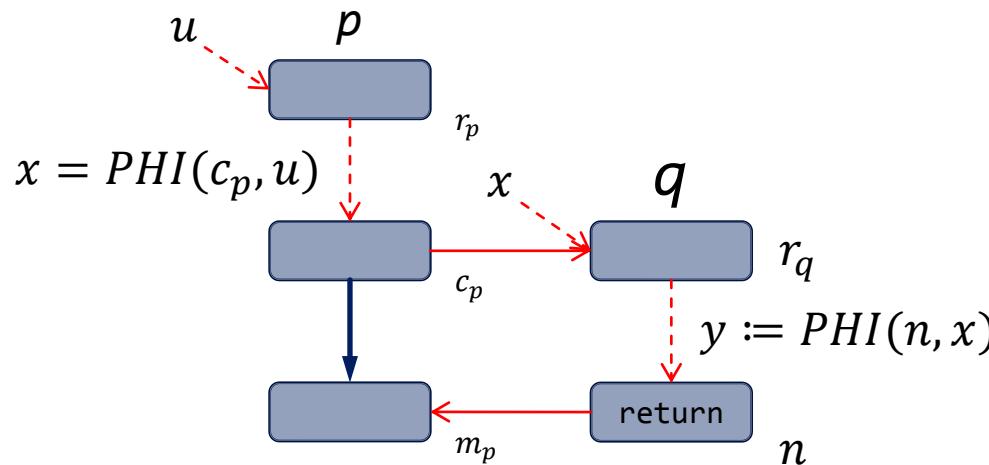
## Case 1



```
if z undefined:  
    propagate(y, y, rp)  
else:  
    propagate(x, z, m)
```

# Algorithm: updating PHIs and $W$

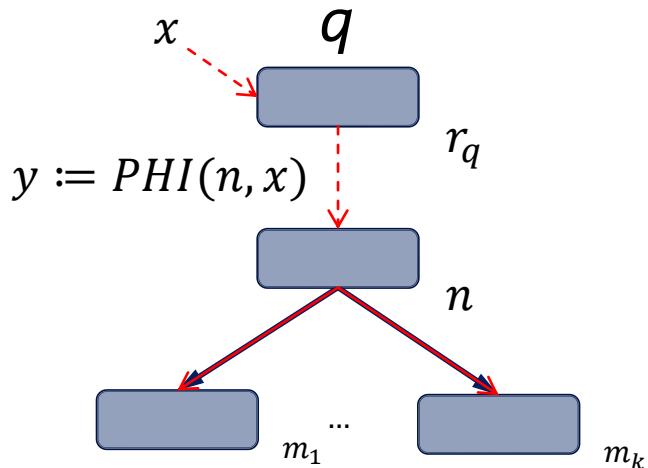
## Case 2



*for each call  $c_p$  to  $q$  and  
 $u \in L$  with  $x = PHI(c_p, u)$  :  
propagate( $u, y, m_p$ )*

# Algorithm: updating PHIs and $W$

## Case 3



*for each  $(n, m) \in E_q^0$ :*  
*propagate( $x, f_{(n,m)}(y), m$ )*

# Example

$$W := \{(r_m, \top) \quad (c_m, \top) \quad (r_p, 1) \quad \dots \quad (e_p, 1) \quad \dots\}$$

*main*

```
read a, b
t := a * b
```

$r_m$

$f_1$

```
call p
```

$c_m$

```
t := a * b
print t
```

$n_m$

```
stop
```

$e_m$

*p*

```
if a=0
```

F  
 $id_L$

```
a := a - 1
```

$f_T$

```
call p
```

```
t := a * b
```

$f_1$

```
return
```

$r_p$

$n_p$

$m_p$

$e_p$

$id_L$

$PHI(r_m, \top) = \top$

$propagate(\top, f_1(\top), c_m)$

$PHI(c_m, \top) = 1$

$propagate(1, 1, r_p)$

$PHI(r_p, 1) = 1$

$propagate(1, id_L(1), n_p)$

$c_p \quad propagate(1, id_L(1), e_p)$

$PHI(n_p, 1) = 1$

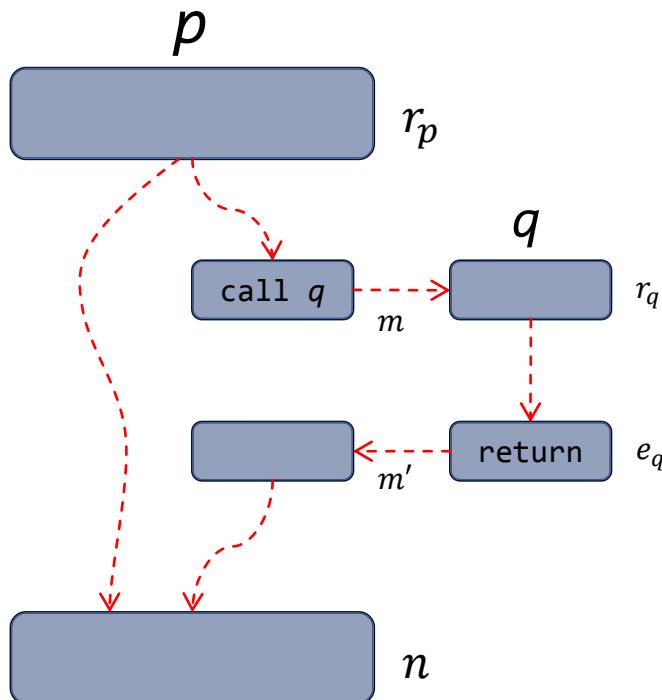
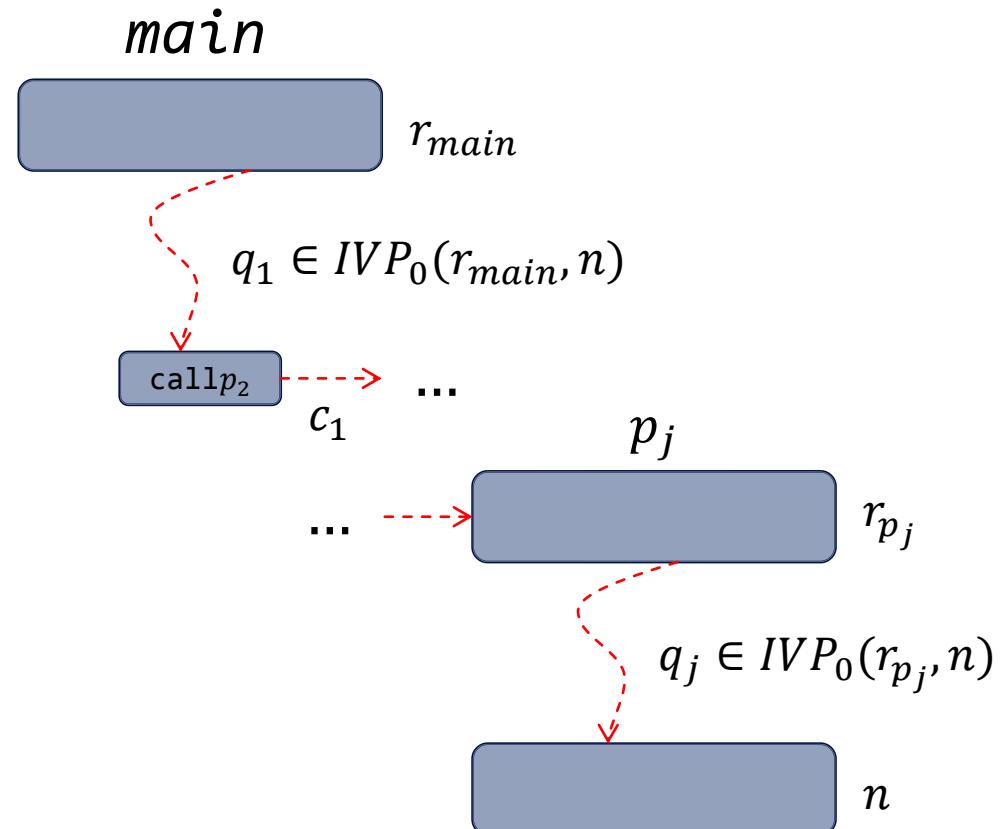
$PHI(e_p, 1) = 1$

...

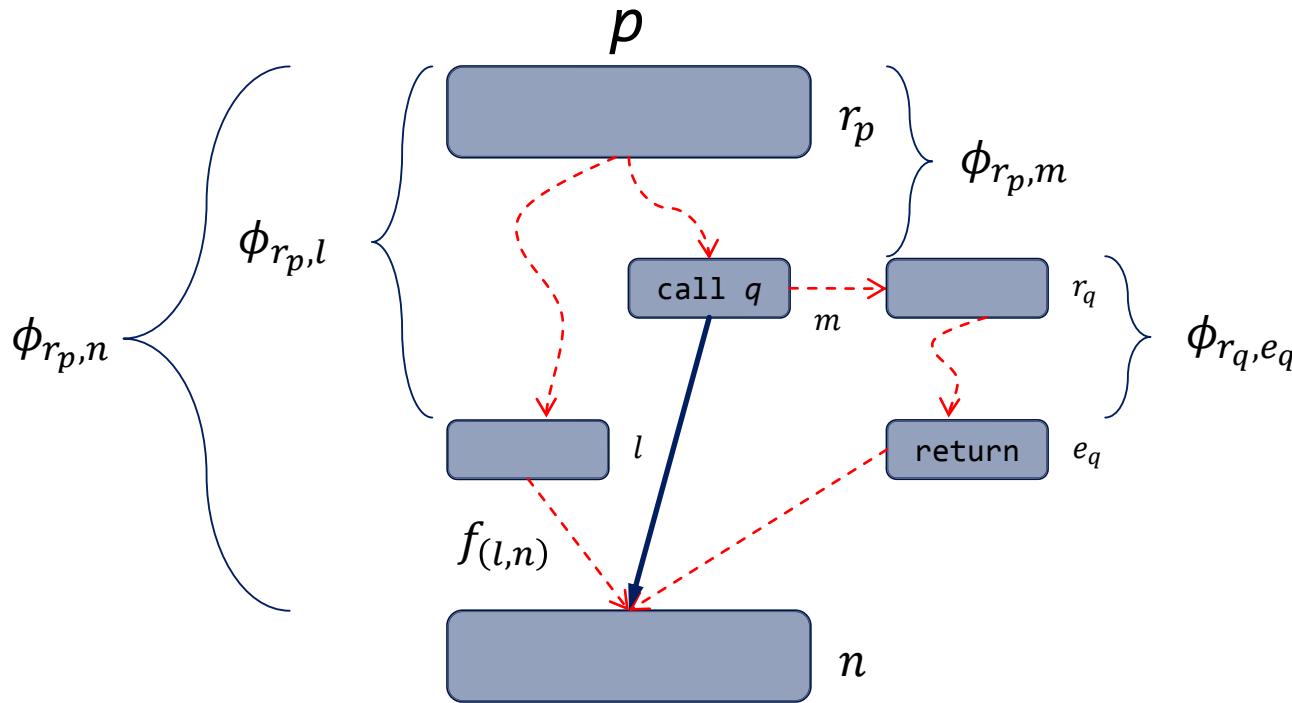
$propagate(\top, 1, n_m)$

$PHI(n_m, \top) = 1$

# Summary: Define valid paths

$$IVP_0(r_p, n)$$

$$IVP(r_{main}, n)$$


# Summary: Functional Approach



Information  $x$  at  $r_p$

is transformed to  $\phi_{r_p, n}(x)$  at  $n$

# Summary: Functional Result

$F$  is *distributive*  $\Rightarrow$

The maximum fixed point solution  $x_n^*$  of the functional approach = the interprocedural MOP solution  $y_n$

# Summary: Practical Aspects

- Represent  $\phi_{r_p,n}$  symbolically or by explicit relation
- Explicit approach may require much space
- $L \text{ infinite} \Rightarrow$  Explicit approach may diverge
- $L \text{ infinite}, F \text{ unbounded} \Rightarrow$  Symbolic approach may diverge