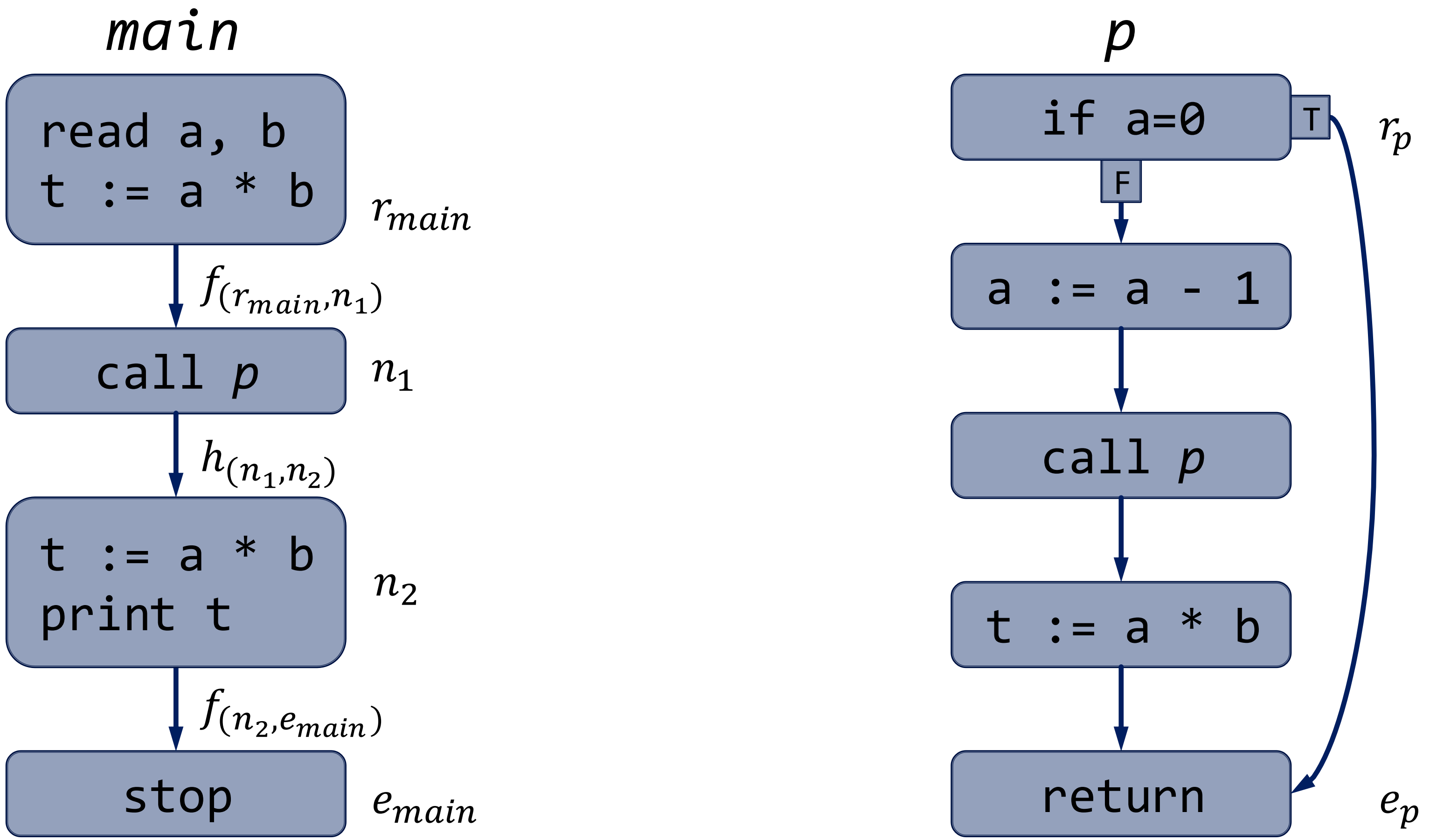
Two Approaches to Interprocedural Data Flow Analysis

Micha Sharir Amir Pnueli

Part Two: The Call String Approach

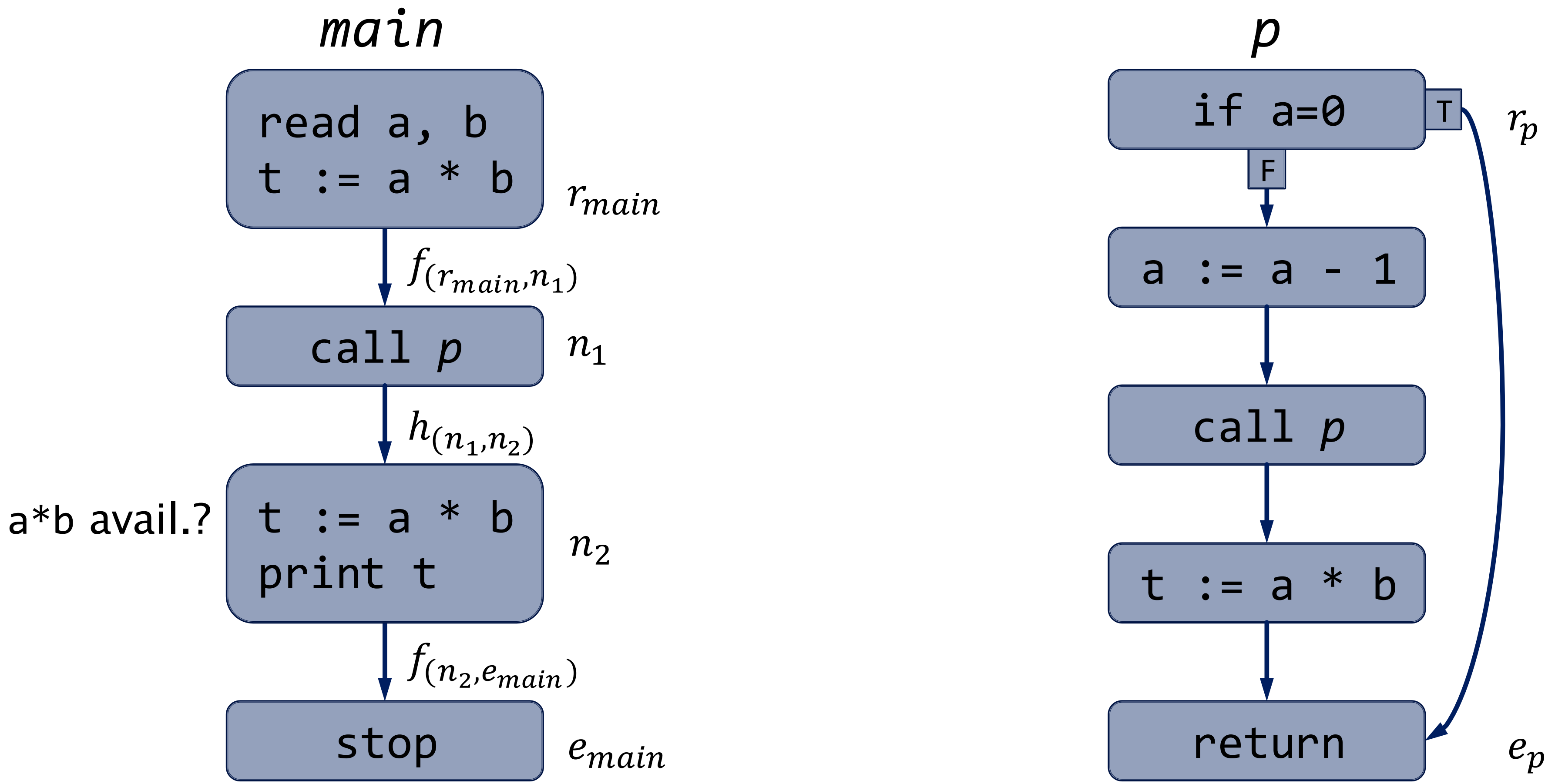
Recap

Functional approach



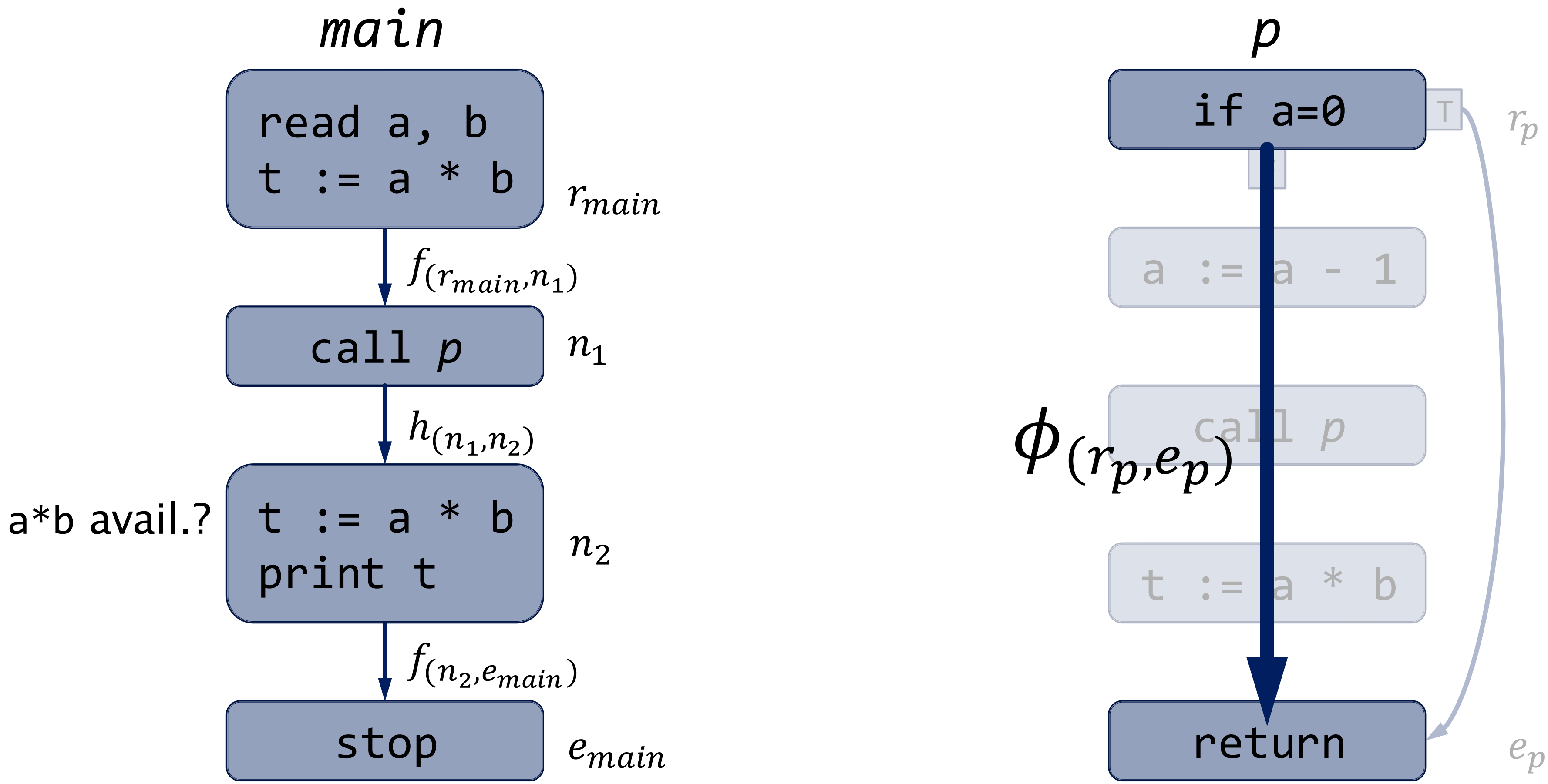
Recap

Functional approach



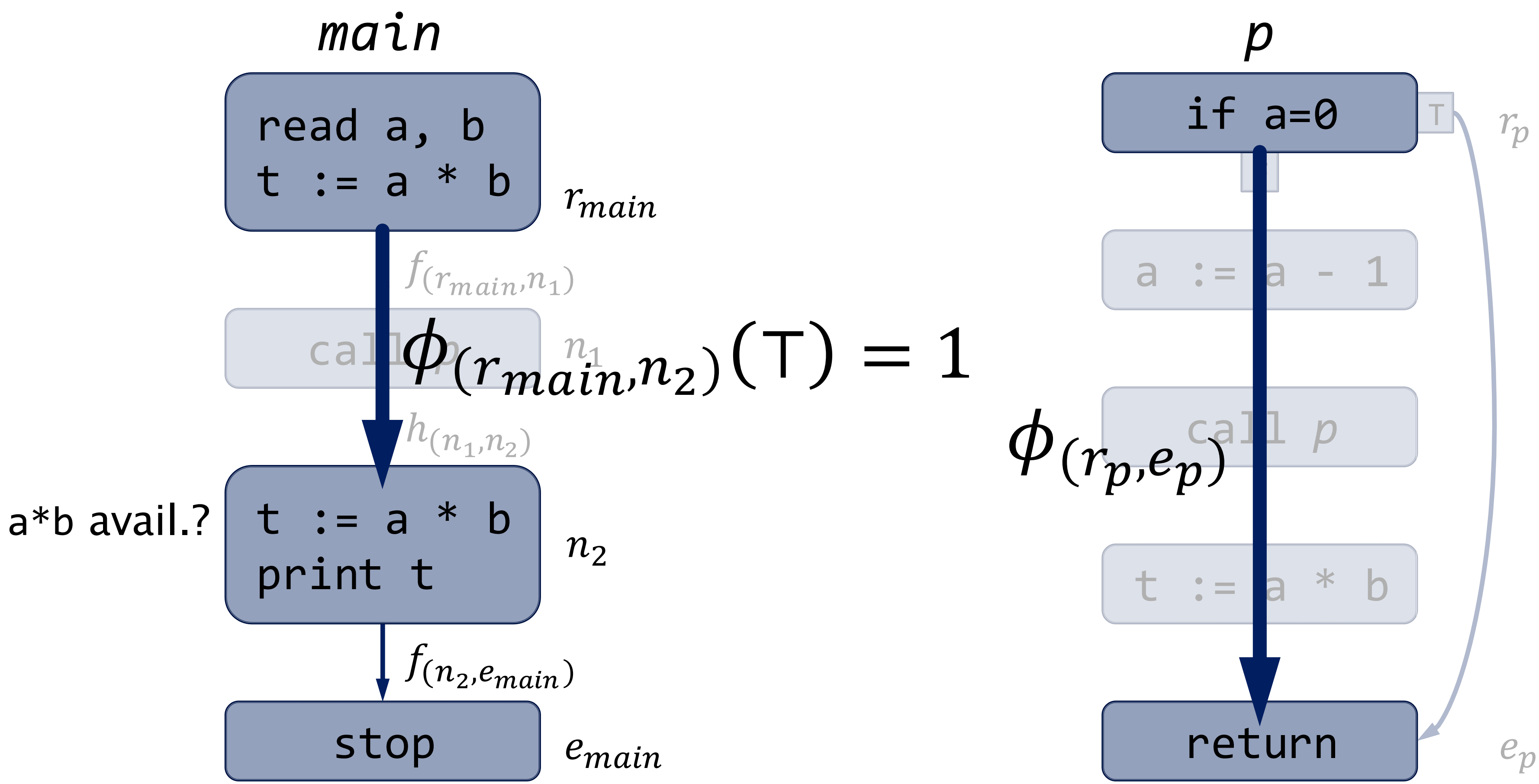
Recap

Functional approach



Recap

Functional approach



Motivation for a different approach

- Avoid expensive functional compositions

- Possibility to trade off precision vs. performance to reduce complexity

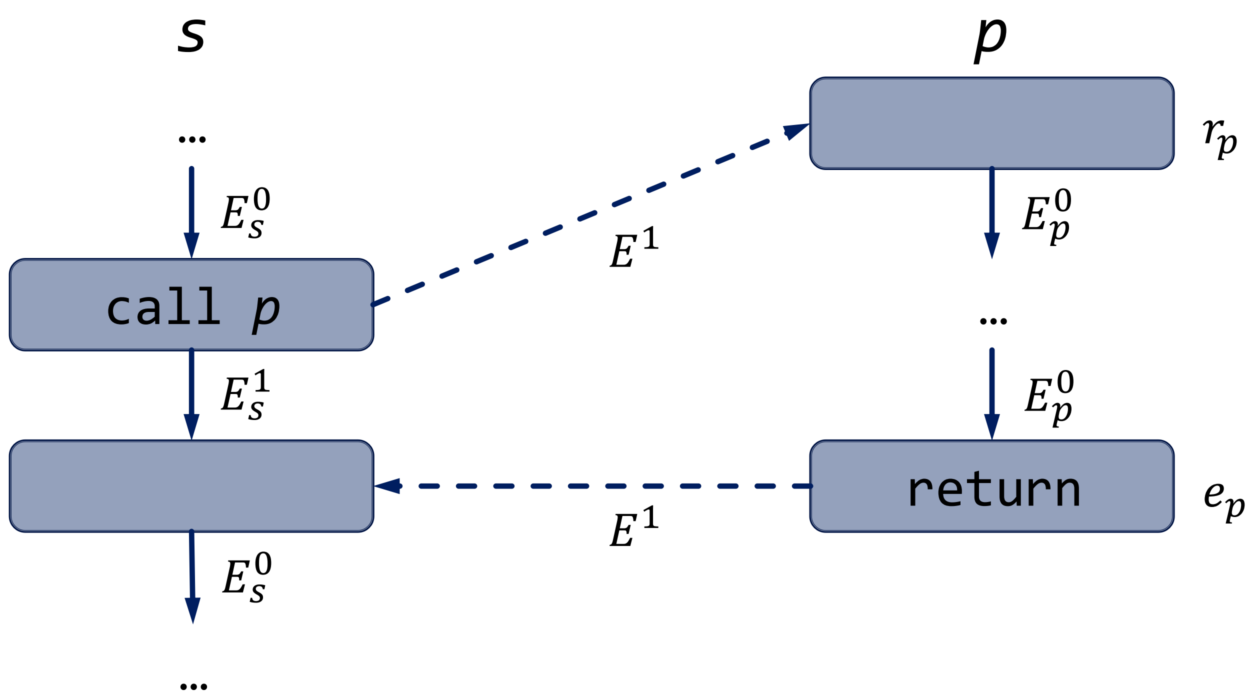
Structure

1. **Definition of a new DF problem $(\boldsymbol{L}^*, \boldsymbol{F}^*)$**
2. Proof: Solution to $(L^*, F^*) \equiv$ MOP solution
3. Feasibility and precise variants
4. Approximative solution

Recap: Interprocedural Graphs

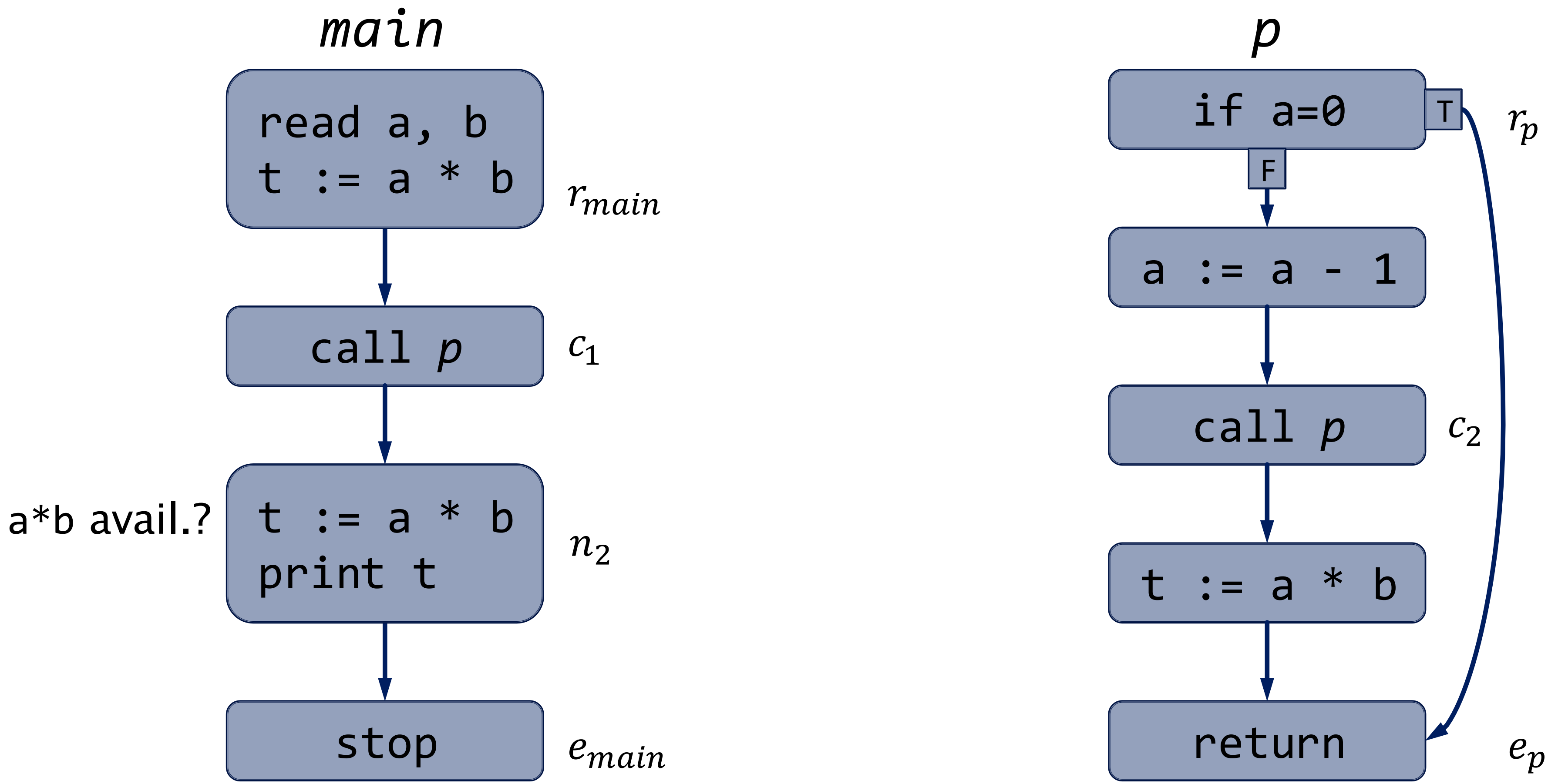
Two representations:

1. $G = \left(\bigcup_p N_p , \bigcup_p E_p\right)$ $\quad E_p = E_p^0 \cup E_p^1 \quad E^0 = \bigcup_p E_p^0$
2. $G^* = \left(\bigcup_p N_p , E^*\right)$ $\quad E^* = E^0 \cup E^1$



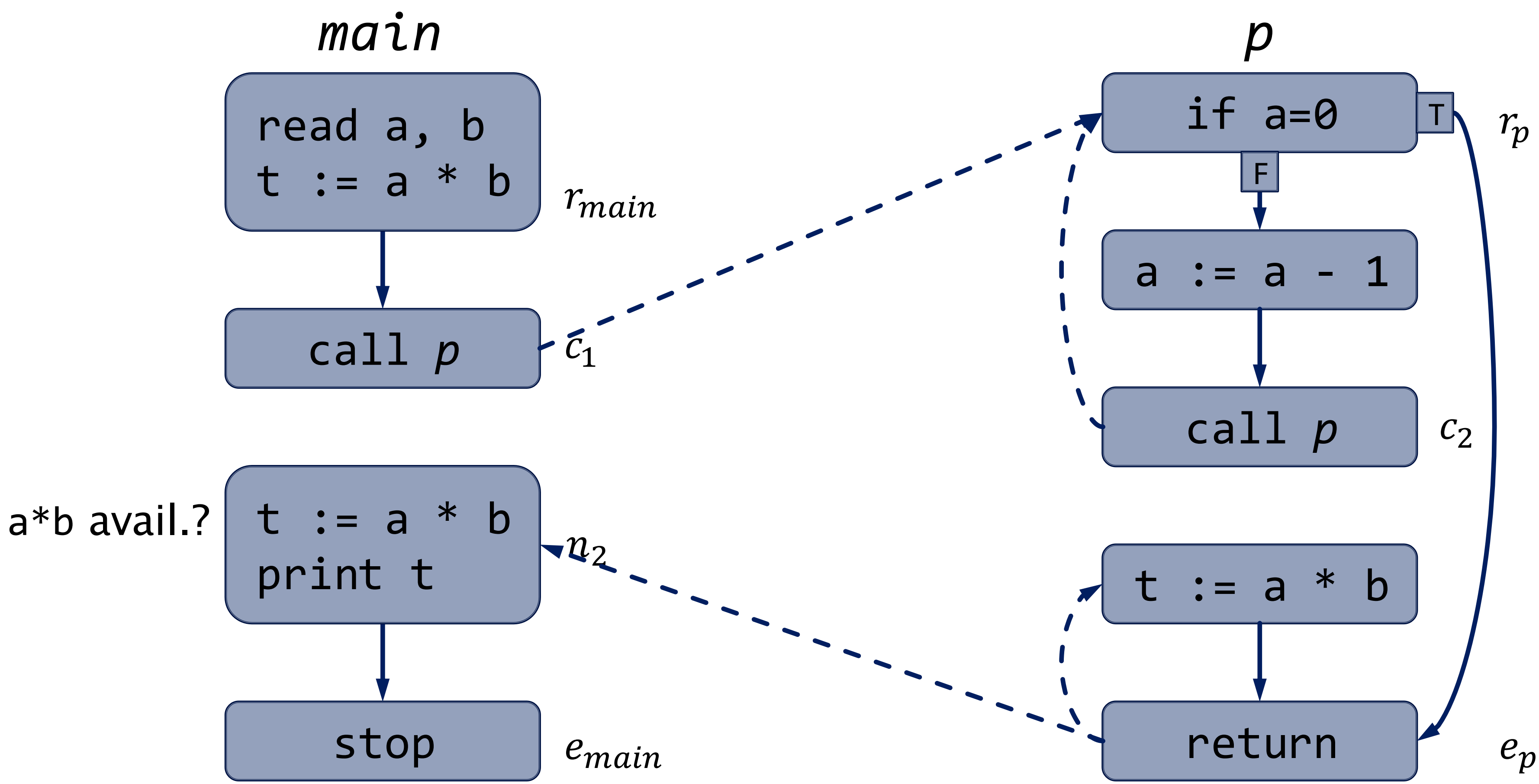
Basic idea

Call String (CS) approach



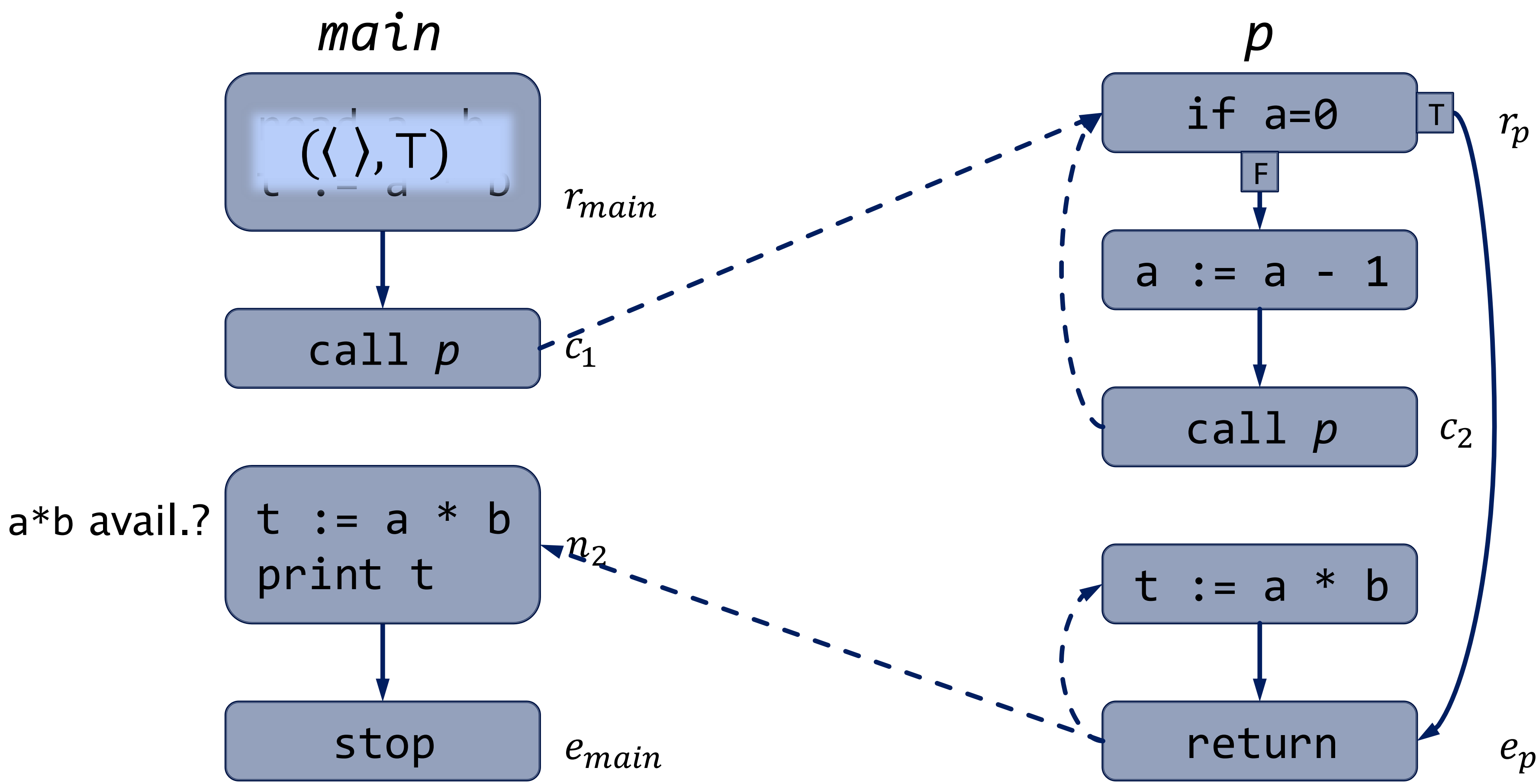
Basic idea

Call String (CS) approach



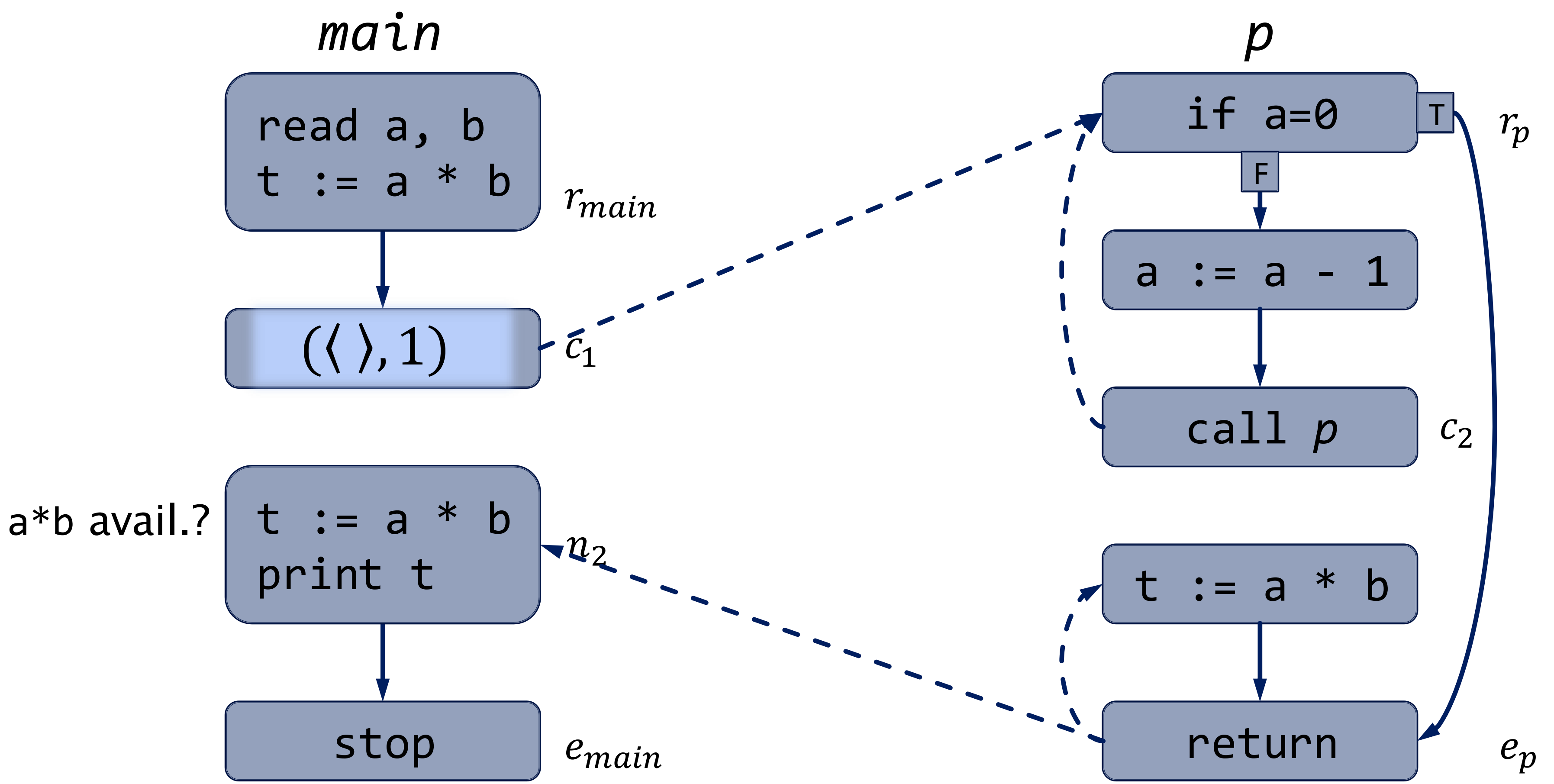
Basic idea

Call String (CS) approach



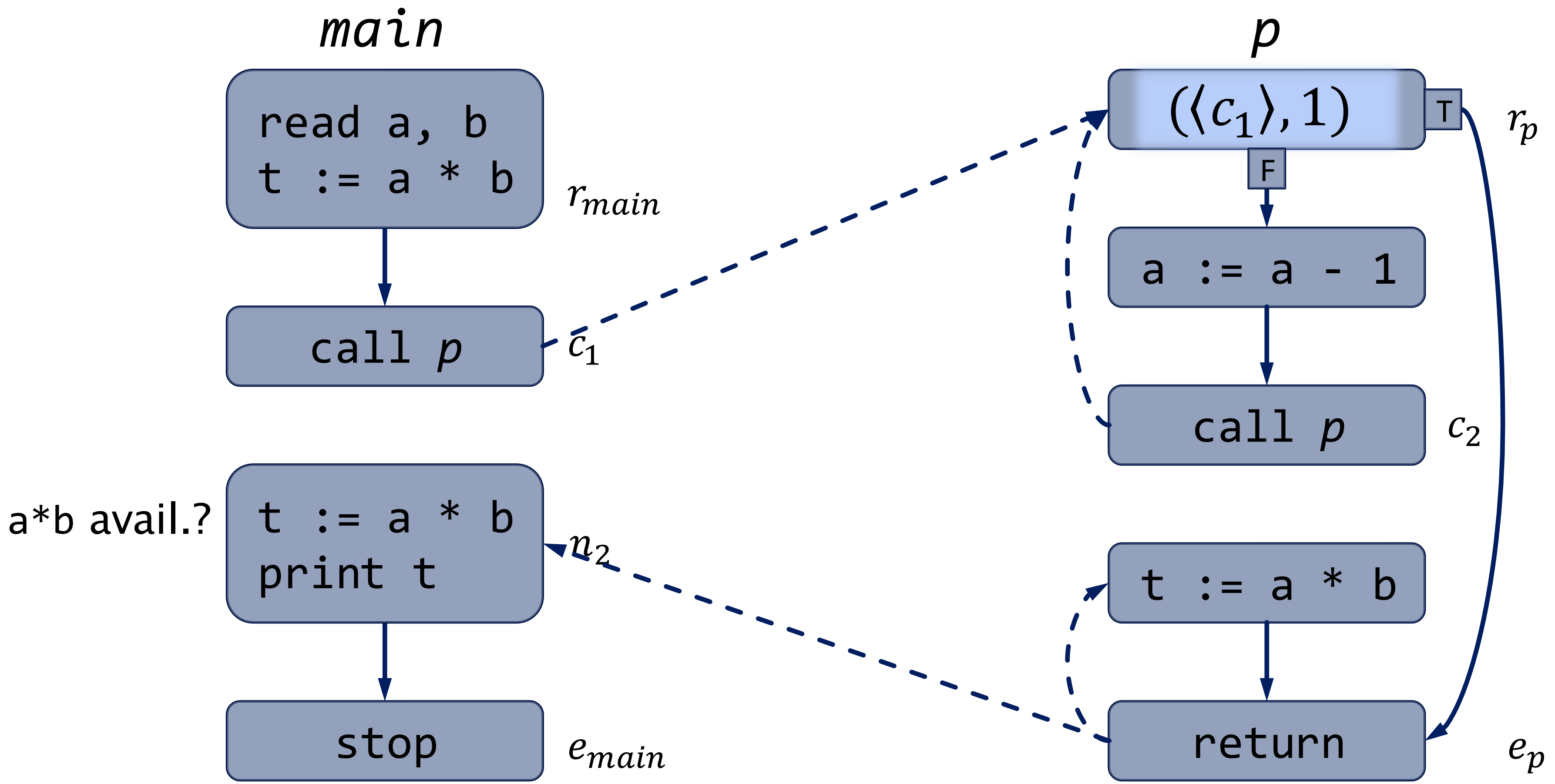
Basic idea

Call String (CS) approach



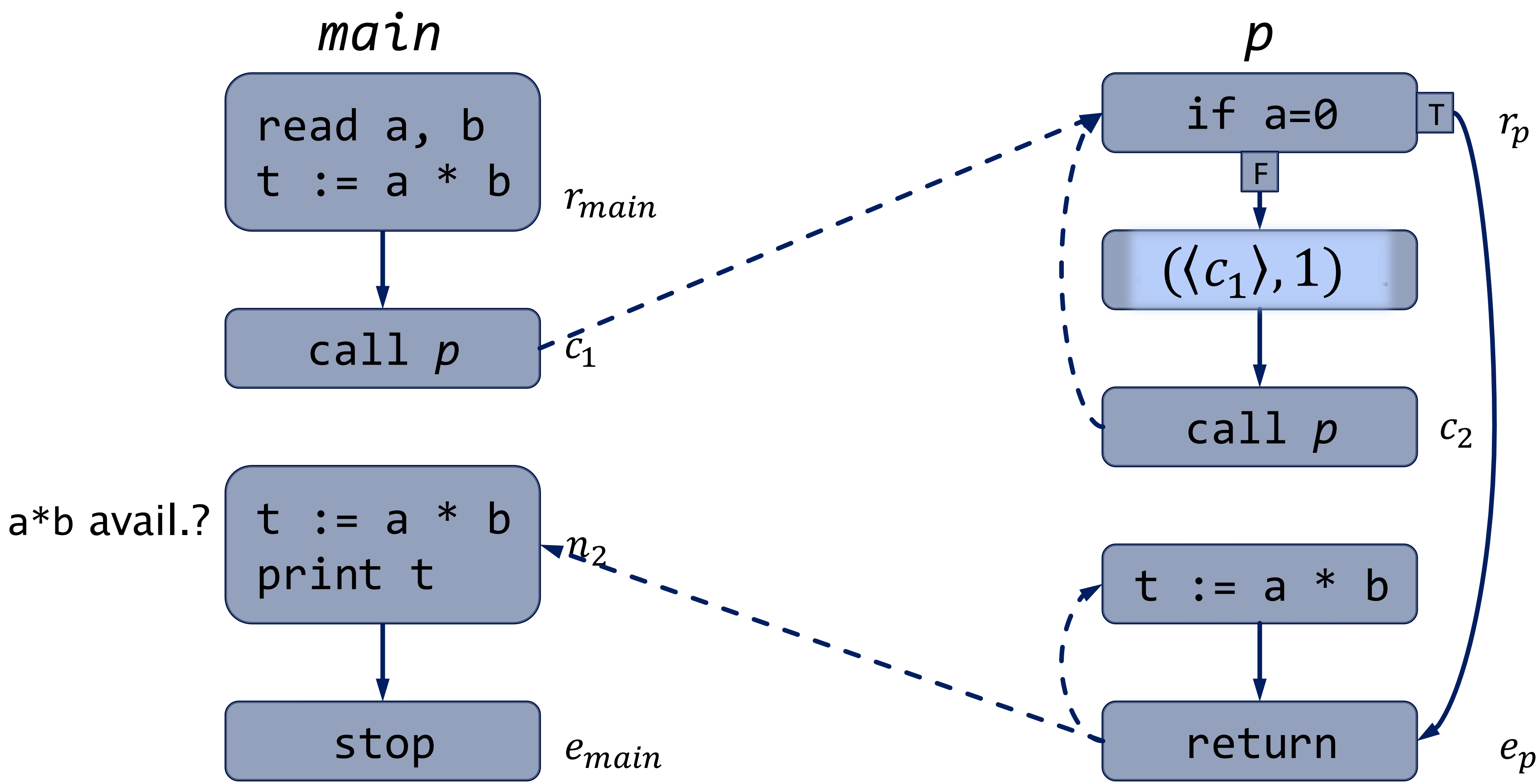
Basic idea

Call String (CS) approach



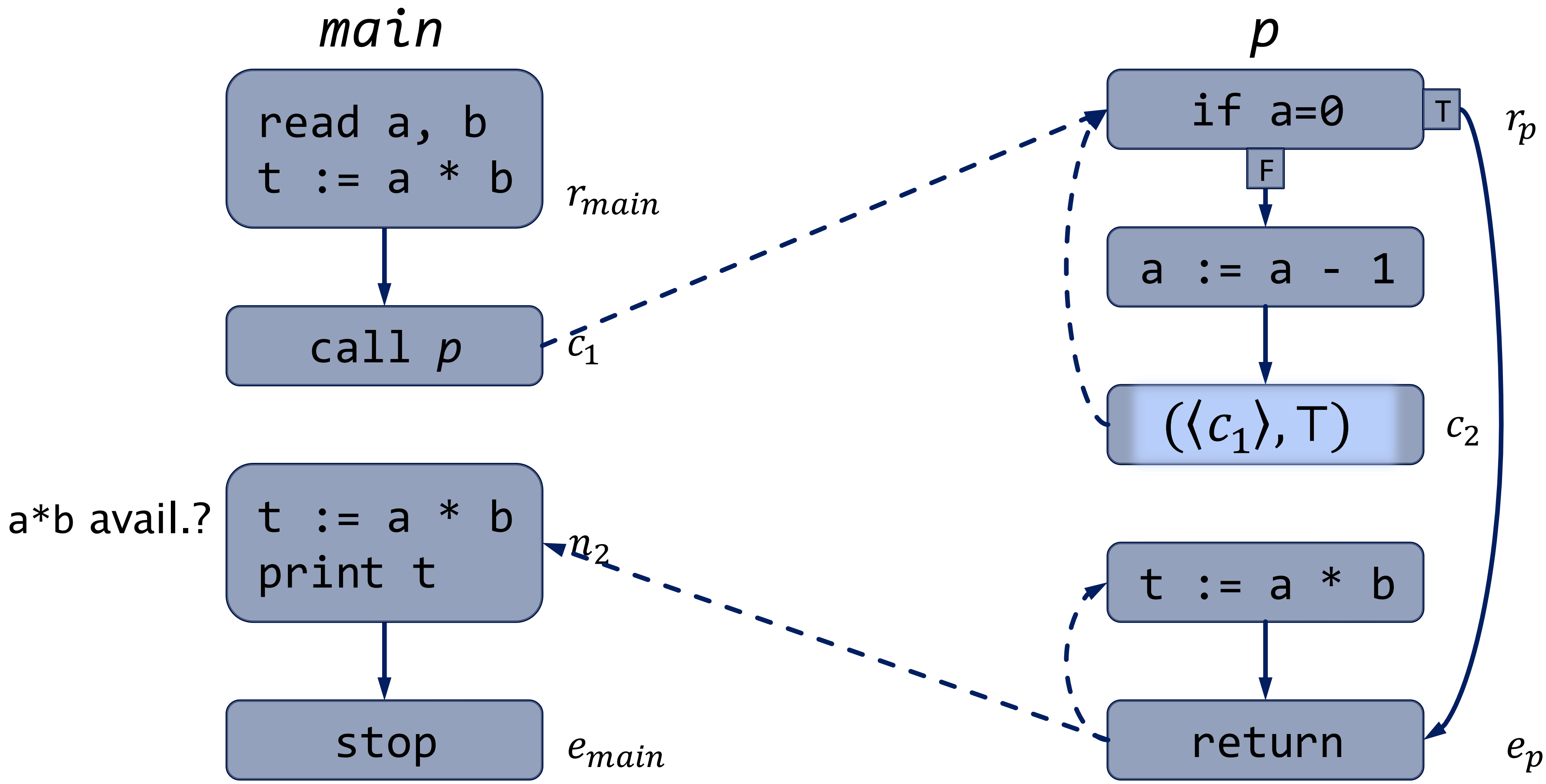
Basic idea

Call String (CS) approach



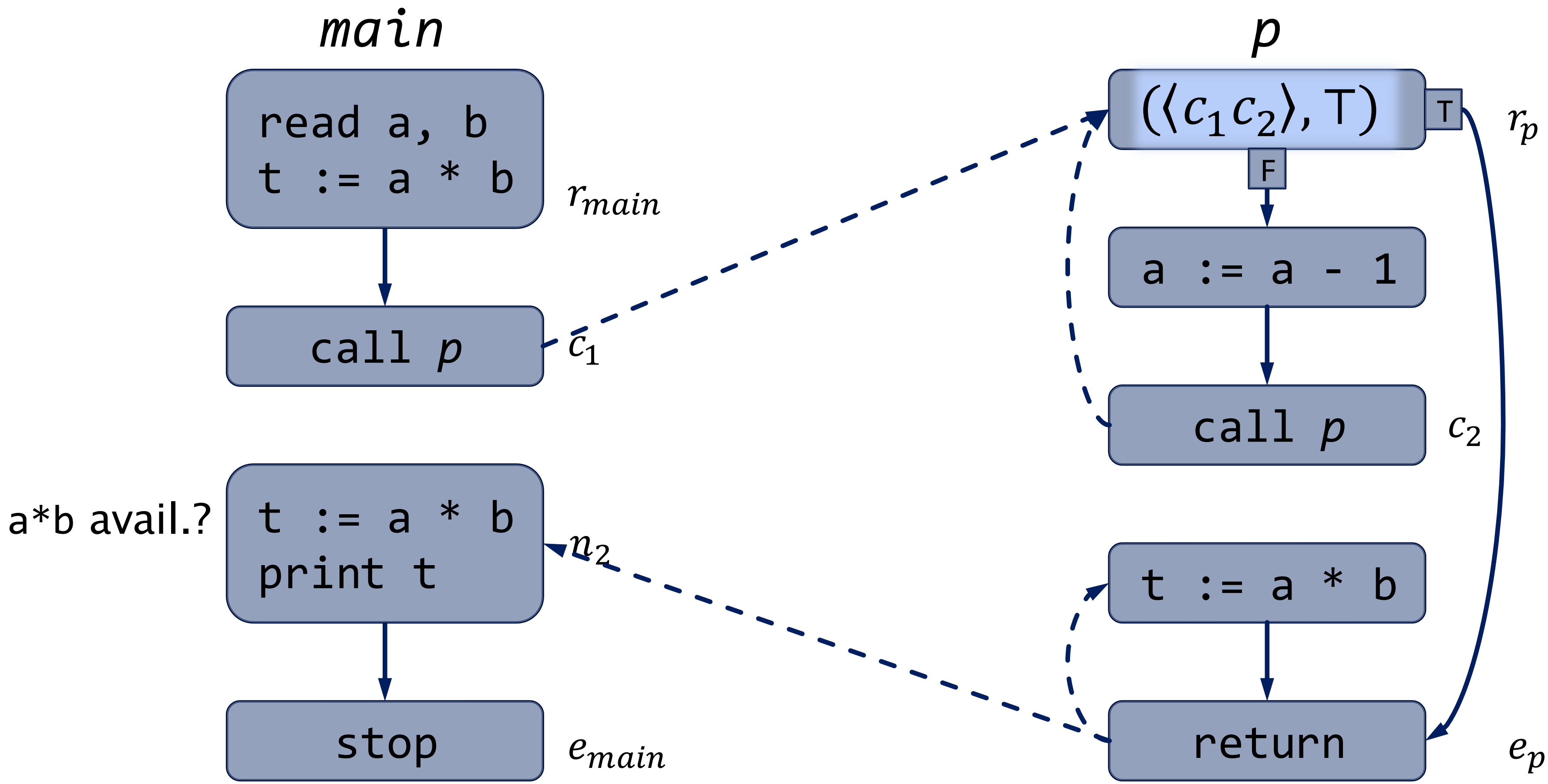
Basic idea

Call String (CS) approach



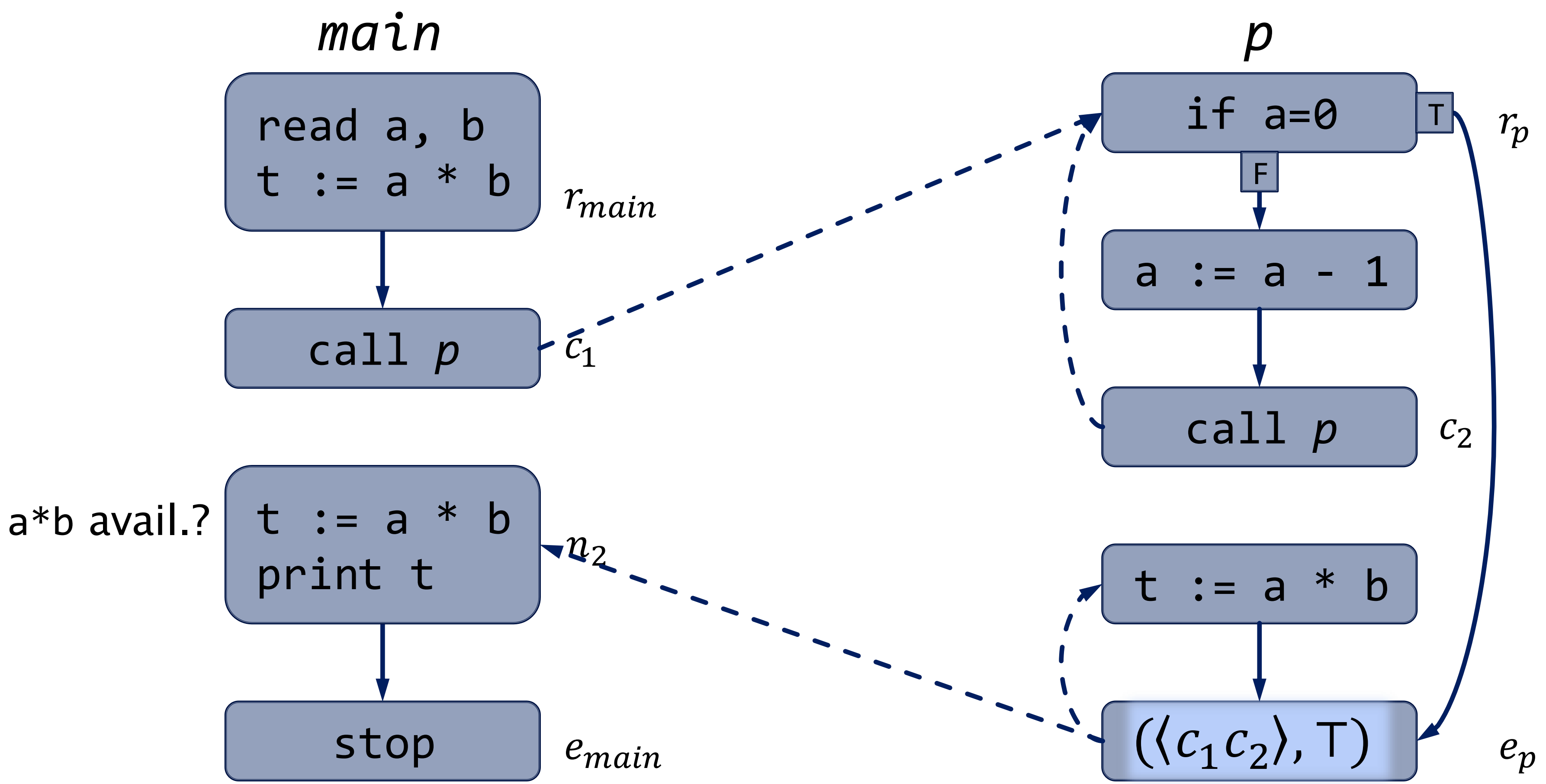
Basic idea

Call String (CS) approach



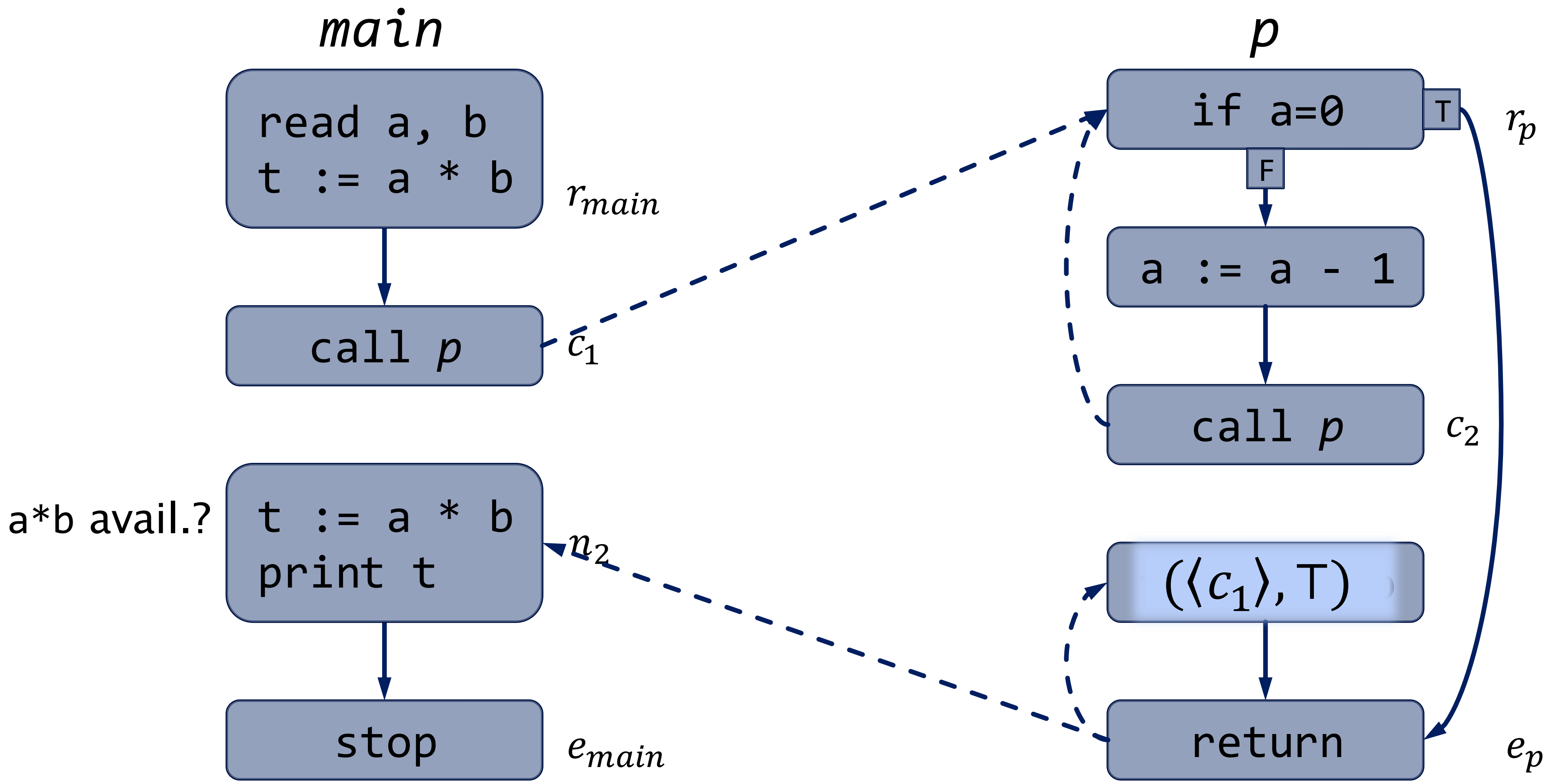
Basic idea

Call String (CS) approach



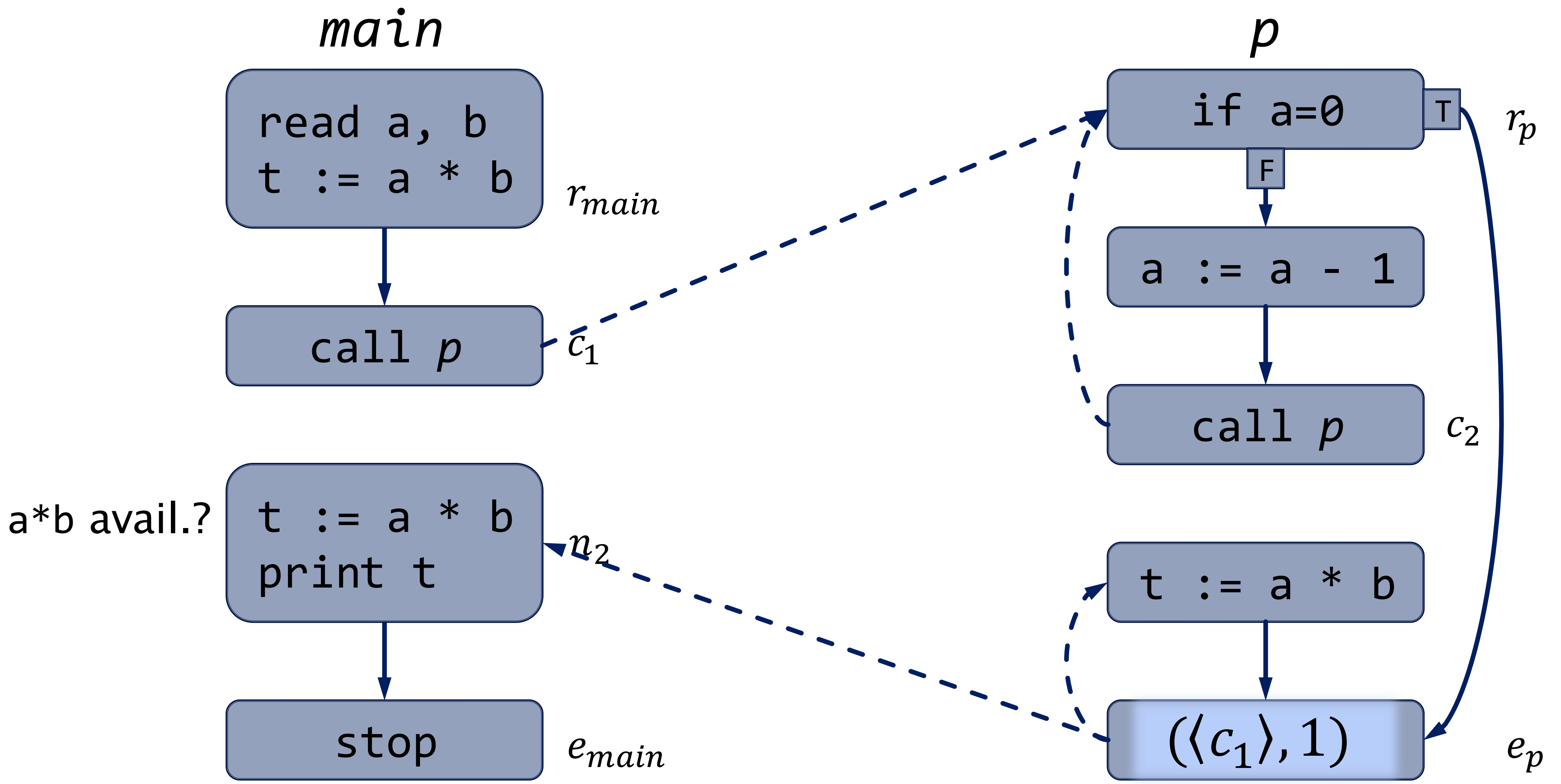
Basic idea

Call String (CS) approach



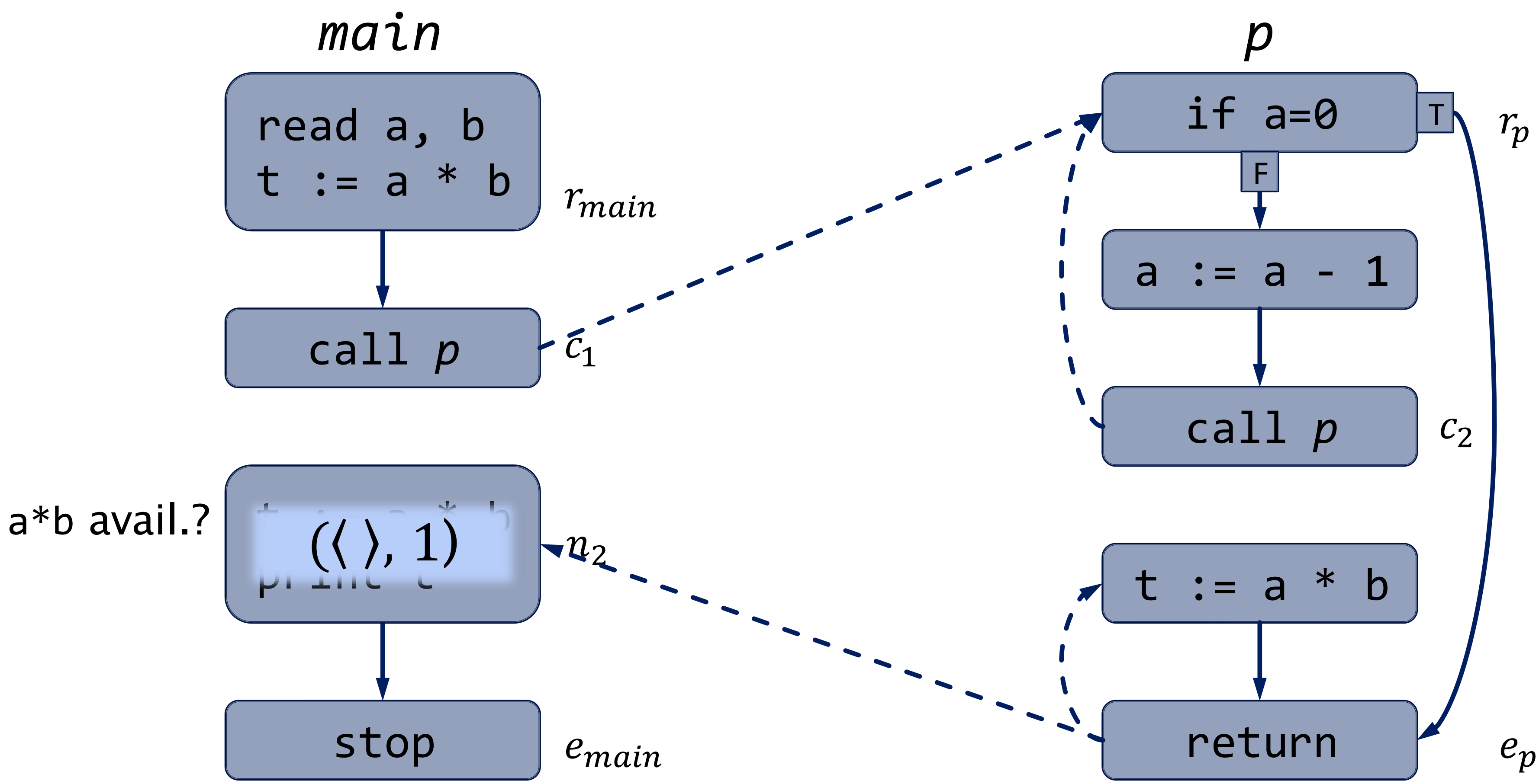
Basic idea

Call String (CS) approach



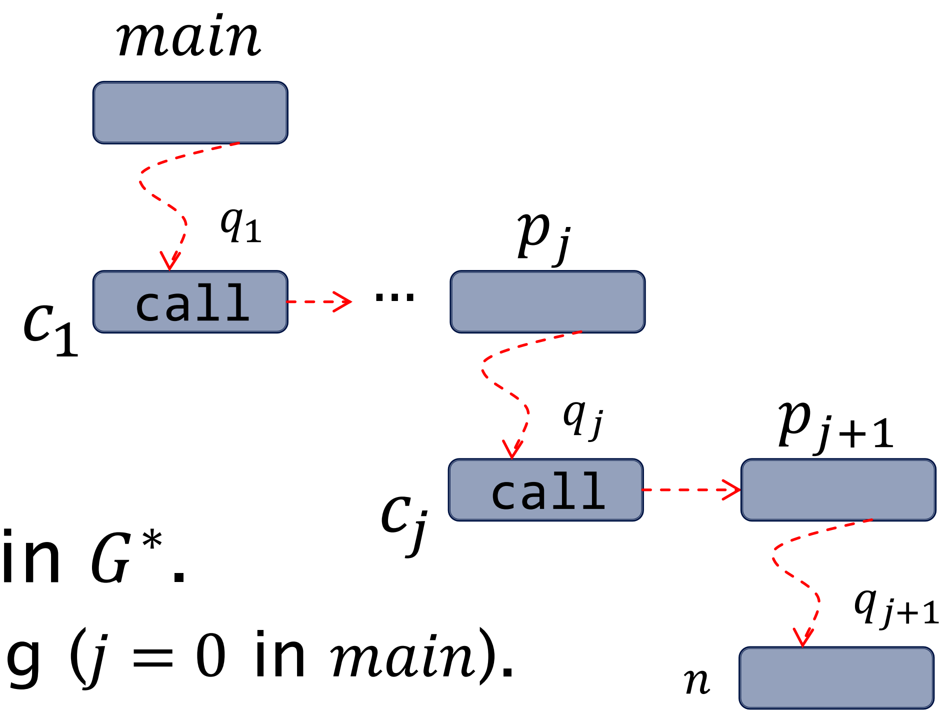
Basic idea

Call String (CS) approach



Definition: Call strings

- Let $q \in IVP(r_{main}, n)$ decomposed as:



- $\langle c_1 c_2 \ldots c_j \rangle =: \gamma \in \Gamma$ **call string (CS)** to q in G^*.
  $\lambda \in \Gamma$ is empty call string ($j = 0$ in $main$).

- $\Gamma =$ space of valid call strings in G^*

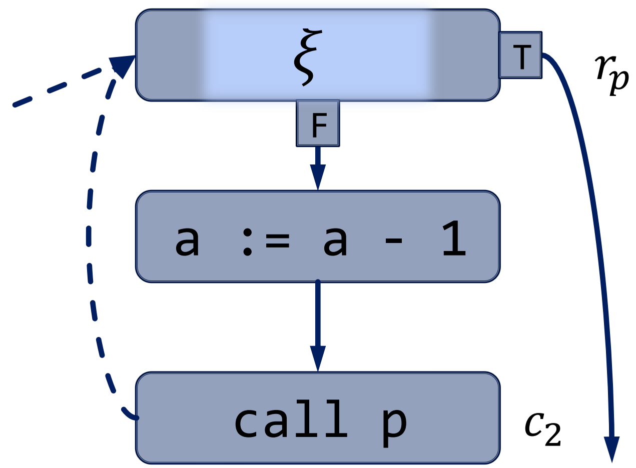
- $CM: IVP \to \Gamma$ with $CM(q) = \gamma$

New DF framework

- (L^*, F^*) := "Extended" version of (L,F)
  - Uses **interprocedural** flow graph G^*
  - L^*: Values $\in L$ tagged with call strings $\in \Gamma$
  - F^*: For data set $\xi \in \mathrm{L}^*$ apply edge effect to call string and value

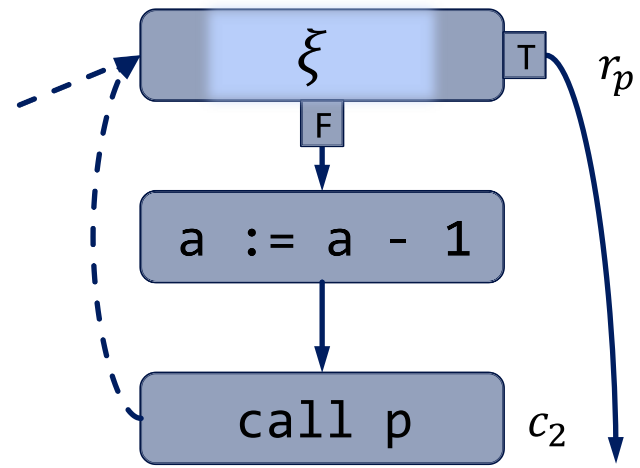
Value domain L^*

- $L^* := \Gamma \to L$ semi–lattice
  - $\top^* = \gamma \mapsto \top$
  - $\bot^* = \gamma \mapsto \bot$
  - $\wedge$ pointwise in L:
    $(\xi_1 \wedge \xi_2)(\gamma) = \xi_1(\gamma) \wedge \xi_2(\gamma)$
- "$\xi \in L^*$ maps call strings γ to values in L propagated via all paths $q \in CM^{-1}(\gamma)$"



Value domain L^*

- $L^* := \Gamma \to L$ semi–lattice
  - $\top^* = \gamma \mapsto \top$
  - $\bot^* = \gamma \mapsto \bot$
  - $\wedge$ pointwise in L:
    $(\xi_1 \wedge \xi_2)(\gamma) = \xi_1(\gamma) \wedge \xi_2(\gamma)$



- "$\xi \in L^*$ maps call strings γ to values in L propagated via all paths $q \in CM^{-1}(\gamma)$"

$$\xi = \left\{ \begin{array}{c} (\langle c_1 \rangle, 1), \\ (\langle c_1 c_2 \rangle, \top), \\ (\langle c_1 c_2 c_2 \rangle, \top) \end{array} \right\}$$

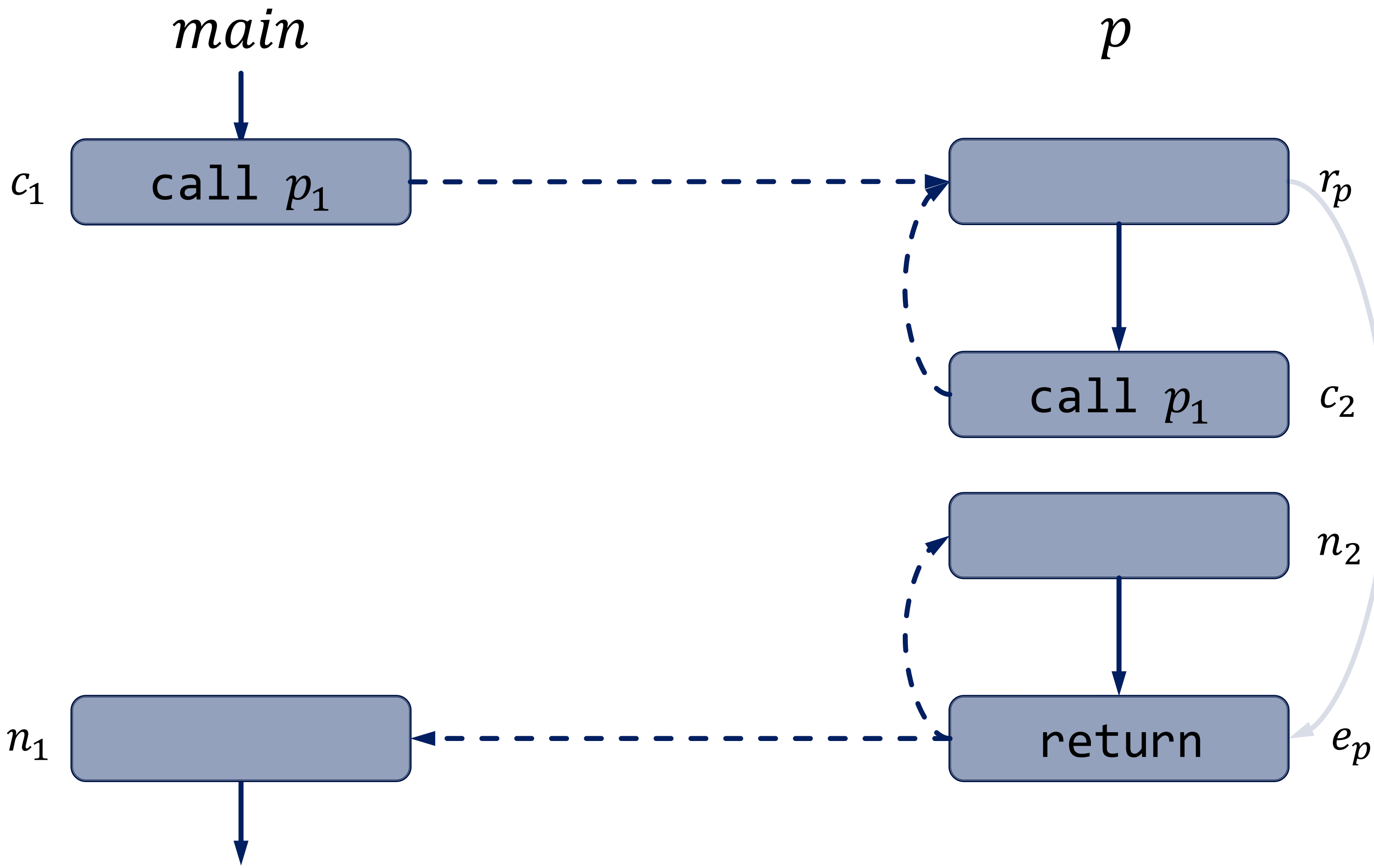
$$\xi(\langle c_1 \rangle) = 1$$
$$\xi(\langle c_2 \rangle) = \bot$$

Edge effects domain F^*

- $F^* := L^* \to L^*$ embeds edge effects from F into L^* domain
- For $(m, n) \in E^*$ and data $\xi = \{.., (\gamma', v'), ..\} \in L^*$ at m, data at n is $f^*_{(m,n)}(\xi) = \{.., (\gamma, v), ..\}$
  - update each γ' to reflect (m, n) taken: $\gamma = \gamma' \circ (m, n)$
  - propagate effect $f_{(m,n)} \in L$ applied to each v': $v = f_{(m,n)}(v')$

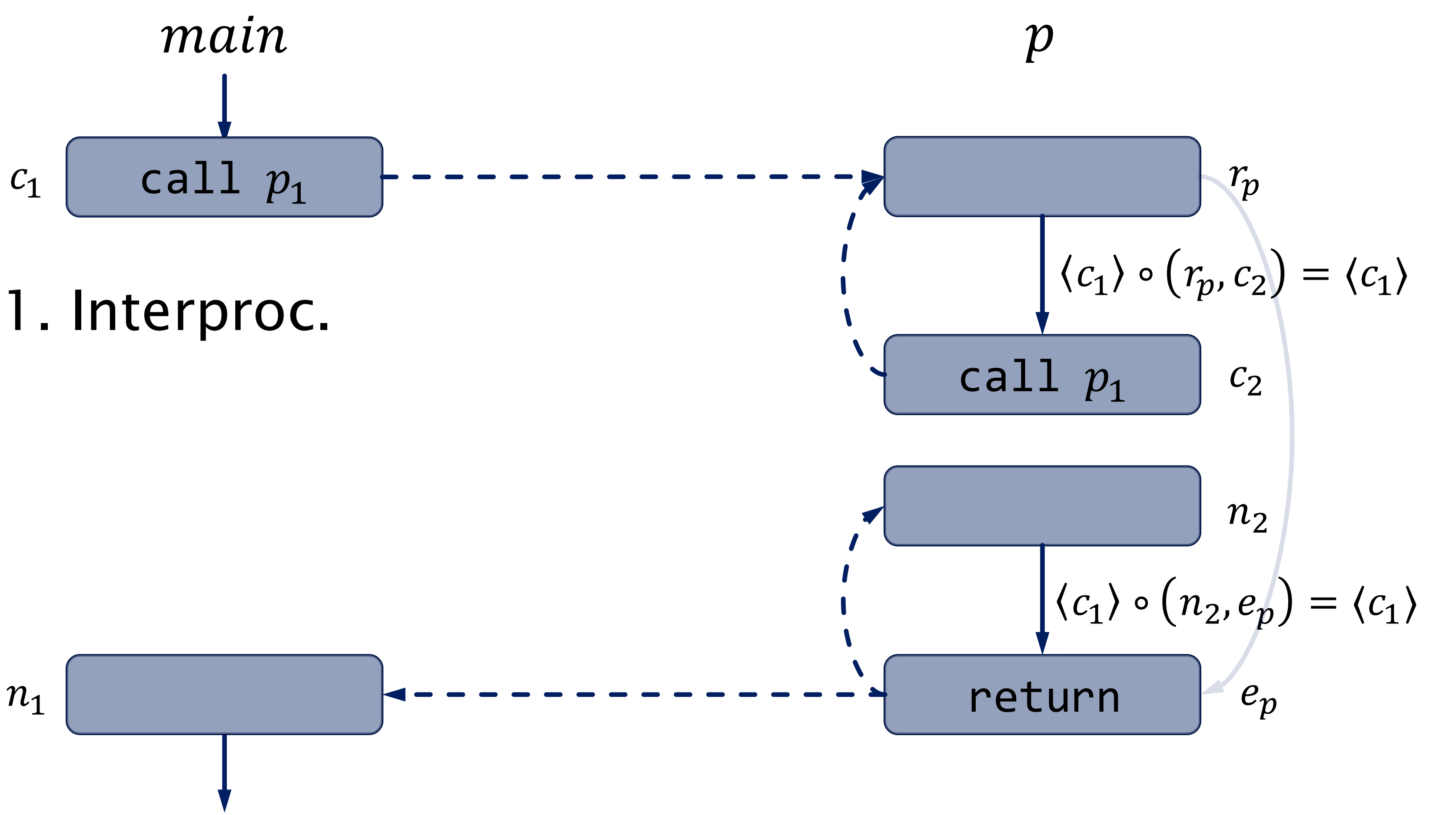
CS update operation $\circ$

- $\circ\colon \Gamma \times E^* \to \Gamma$ updates call strings along edges



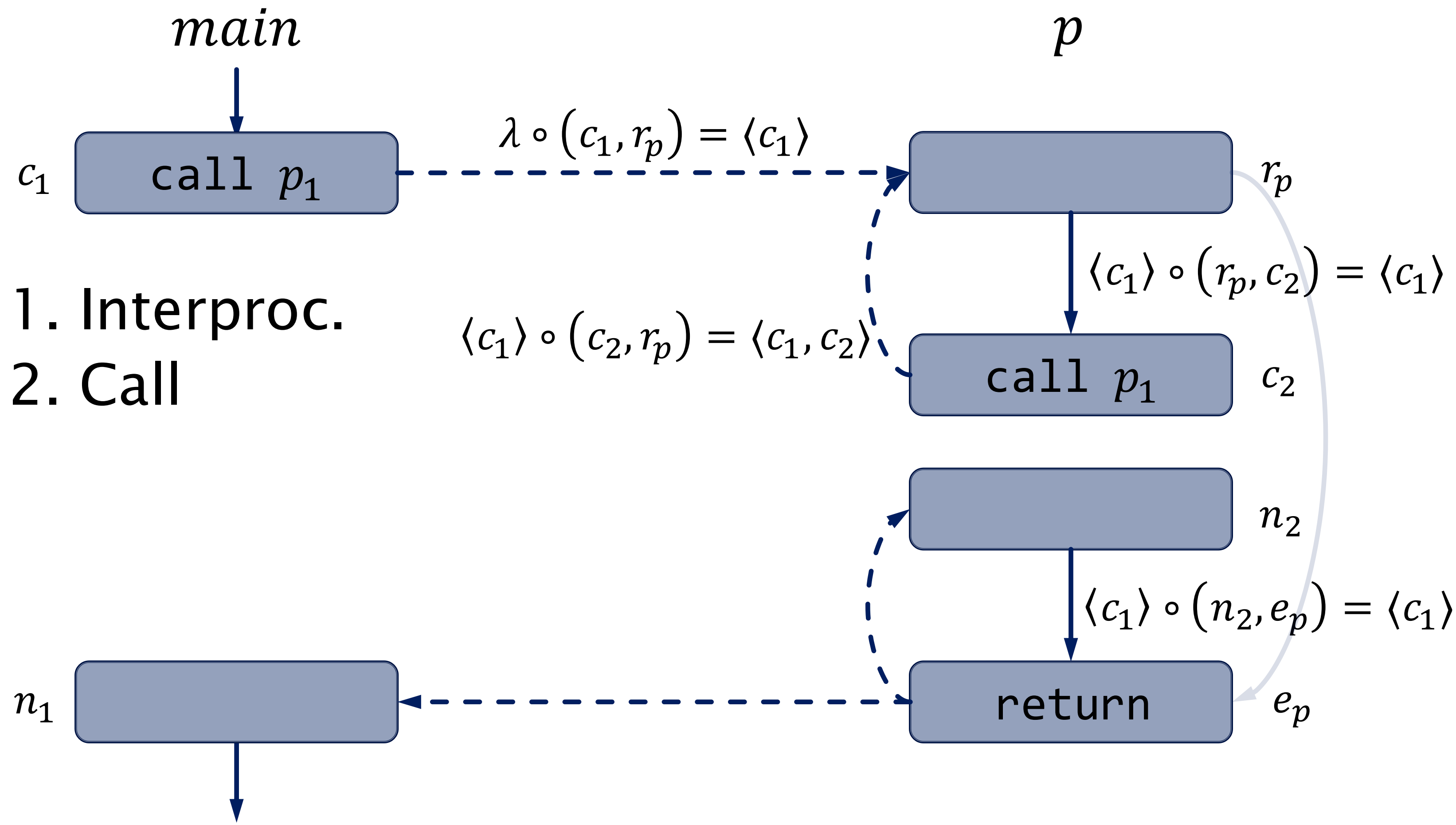
CS update operation $\circ$

- $\circ\colon \Gamma \times E^* \to \Gamma$ updates call strings along edges



CS update operation ∘

- $\circ: \Gamma \times E^* \to \Gamma$ updates call strings along edges

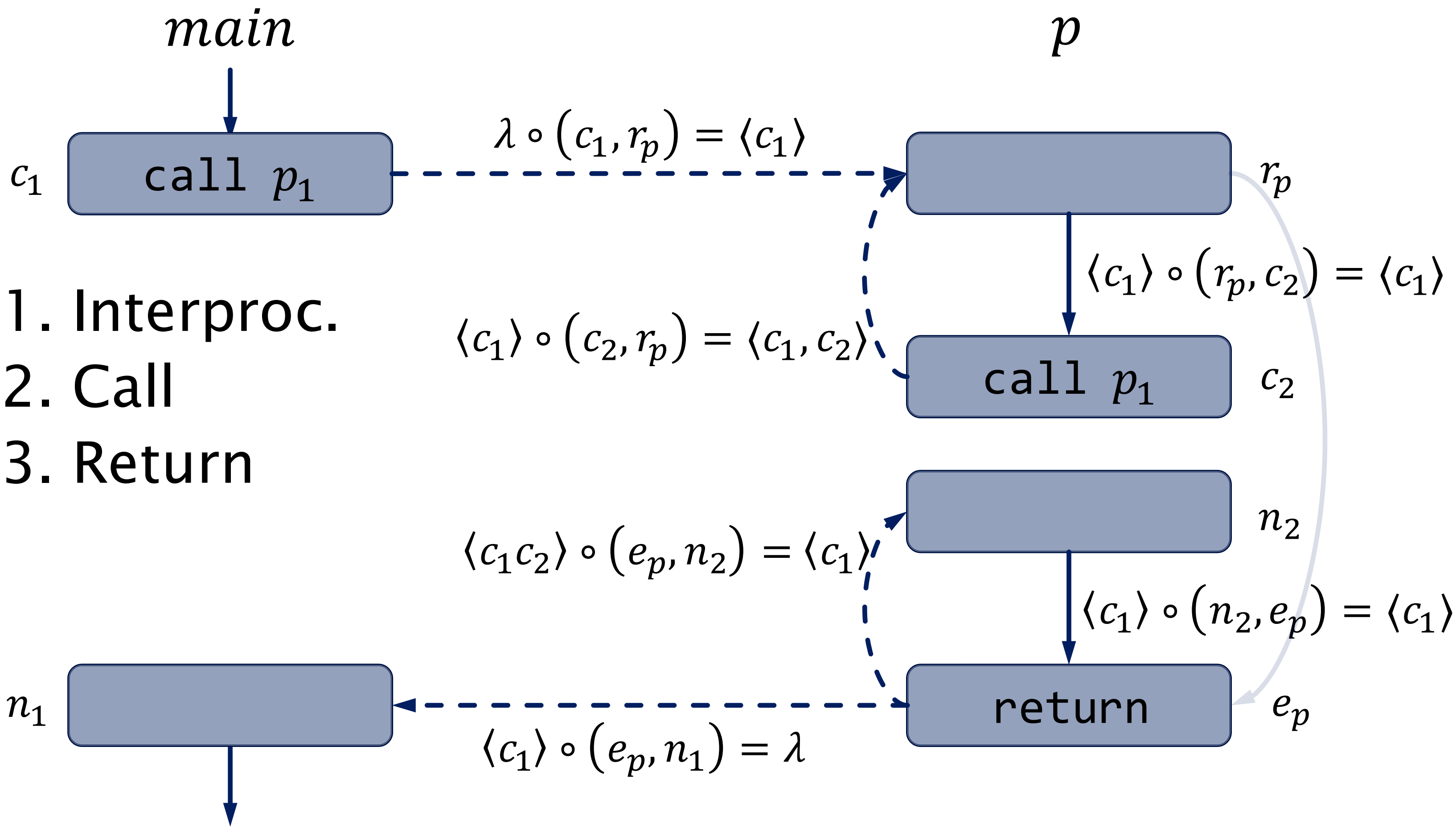


1. Interproc.
2. Call

CS update operation ∘

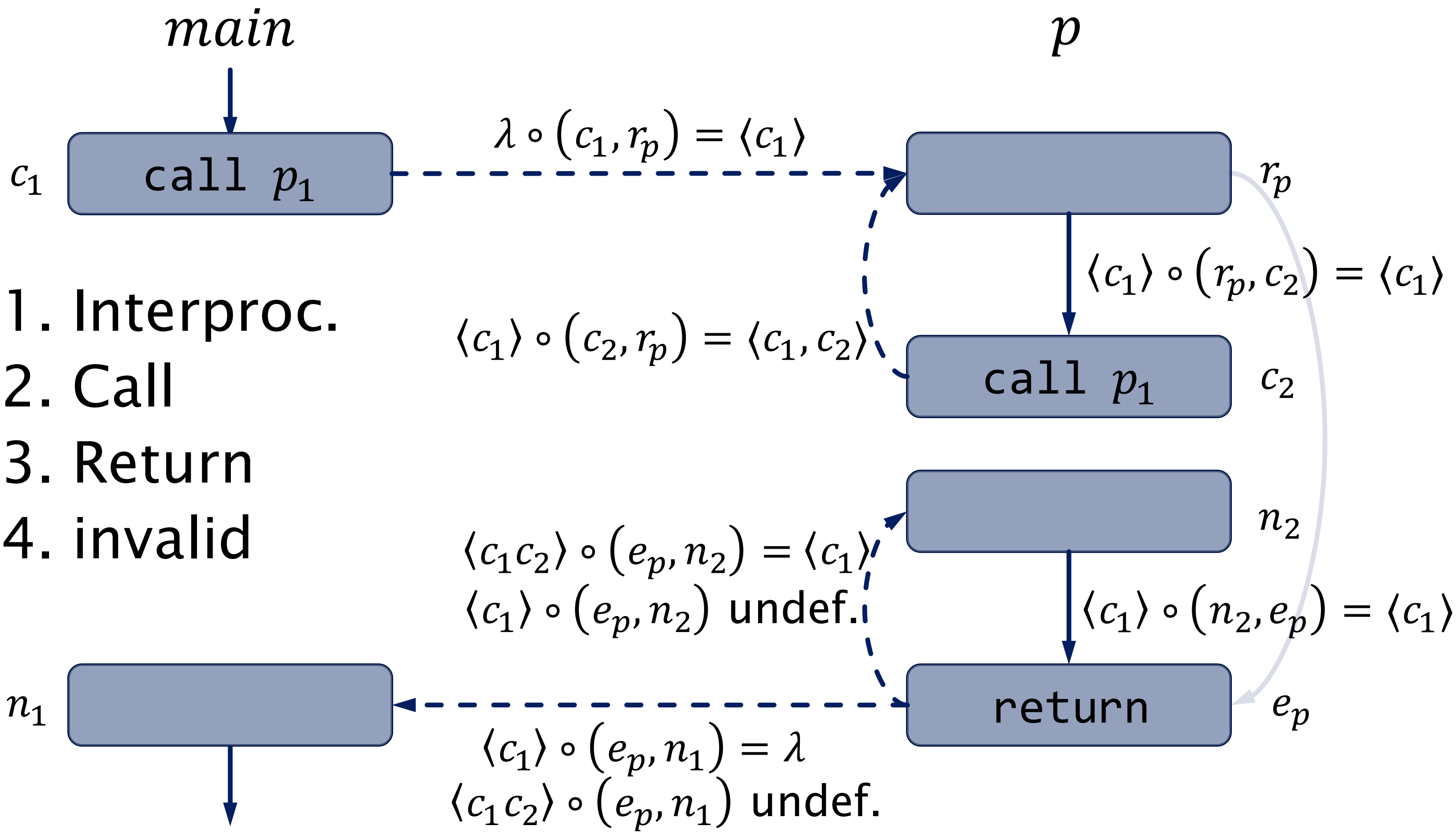
- $\circ: \Gamma \times E^* \rightarrow \Gamma$ updates call strings along edges

1. Interproc.
2. Call
3. Return



CS update operation $\circ$

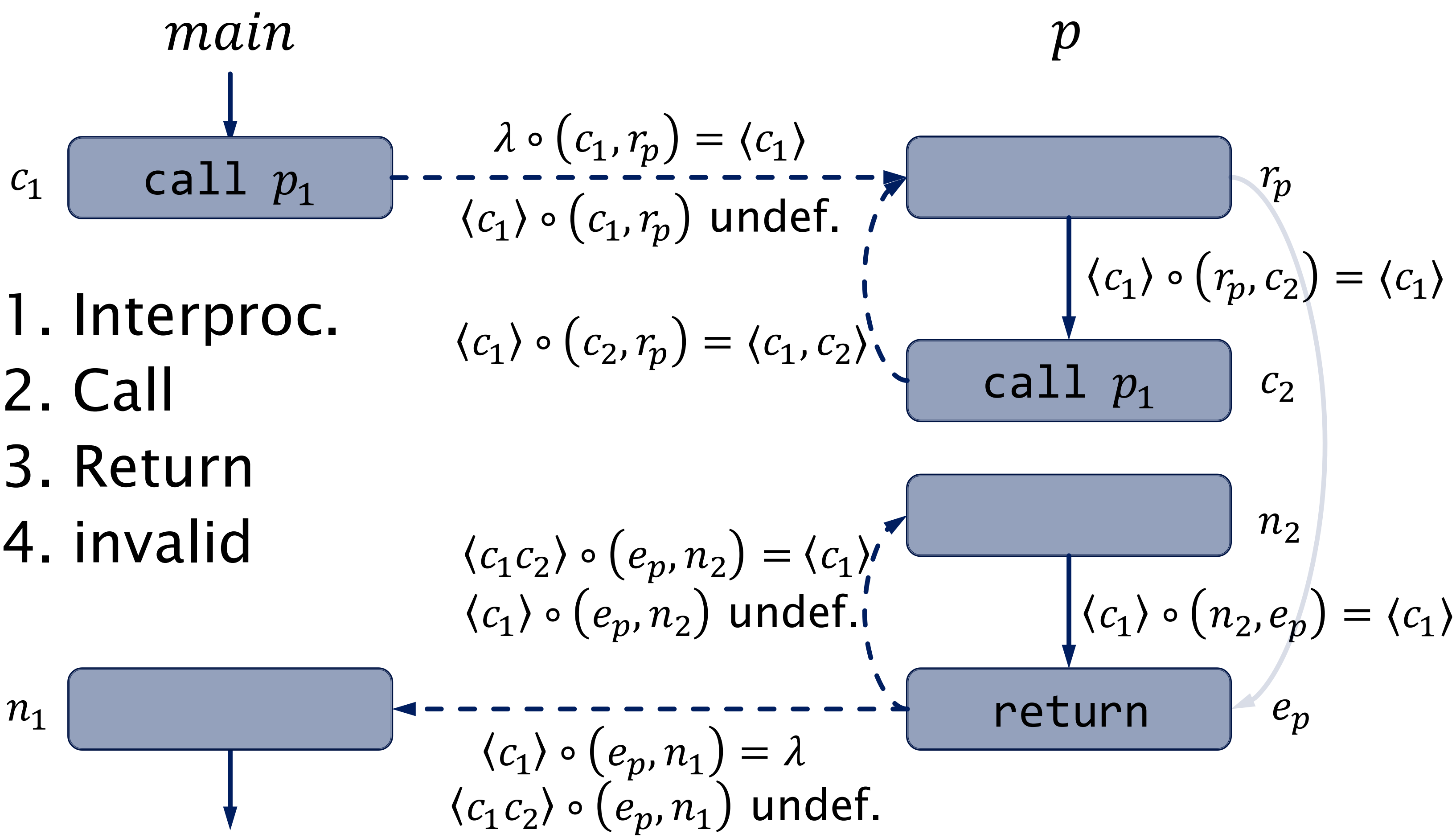
- $\circ: \Gamma \times E^* \rightarrow \Gamma$ updates call strings along edges



1. Interproc.
2. Call
3. Return
4. invalid

CS update operation $\circ$

- $\circ\colon \Gamma \times E^* \to \Gamma$ updates call strings along edges



1. Interproc.
2. Call
3. Return
4. invalid

$\circ$ consistent with IVP

- $\gamma' \circ (m, n) = \gamma$ only defined iff:

  1. (By definition)
     A path $q' \in IVP(r_{main}, m)$ exists with $CM(q') = \gamma'$

  2. (Lemma)
     $q = q' || (m, n)$ lies in $IVP(r_{main}, n)$ and $CM(q) = \gamma$

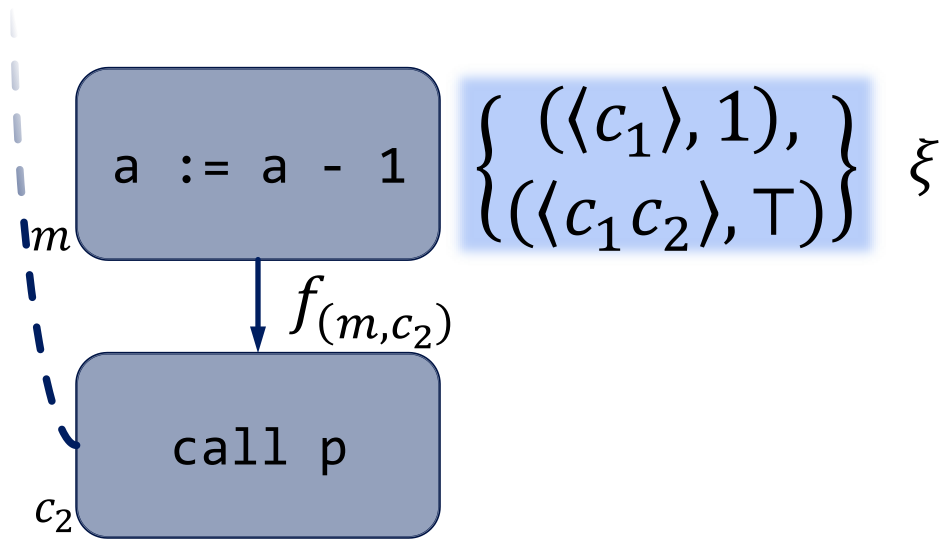
$\Rightarrow$ "$\circ$ only updates and yields call strings for interprocedurally valid paths"

Edge effects domain F^*

- $F^* := L^* \to L^*$ embeds edge effects from F into L^* domain
- For $(m,n) \in E^*$ and data $\xi = \{..,(\gamma',v'),..\} \in L^*$ at m, data at n is $f^*_{(m,n)}(\xi) = \{..,(\gamma,v),..\}$
  - update each γ' to reflect (m,n) taken: $\gamma = \gamma' \circ (m,n)$
  - propagate effect $f_{(m,n)} \in L$ applied to each v': $v = f_{(m,n)}(v')$

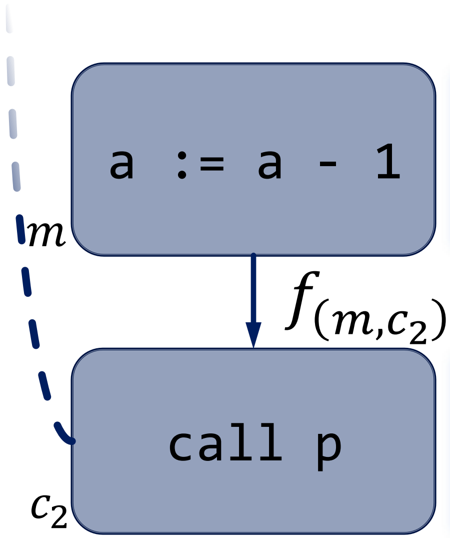
Edge effects $f^*_{(m,n)}$

Definition: Let $(m, n) \in E^*,\ f_{(m,n)} \in F, \xi \in L^*$.



Edge effects $f^*_{(m,n)}$

Definition: Let $(m,n) \in E^*,\; f_{(m,n)} \in F, \xi \in L^*$.

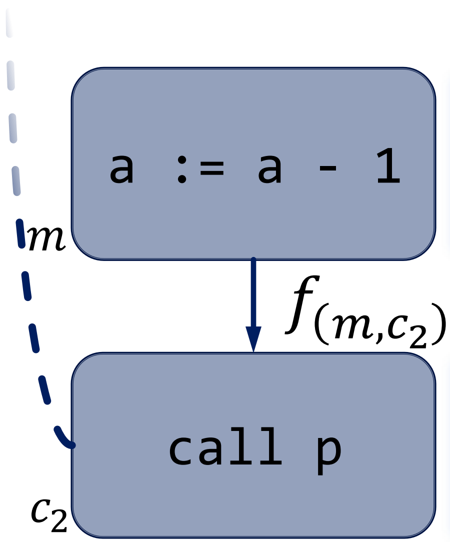


$$\left\{\begin{matrix} ((\langle c_1 \rangle, 1), \\ ((\langle c_1 c_2 \rangle, \top) \end{matrix}\right\} \quad \xi$$

$$\left\{\begin{matrix} ((\langle c_1 \rangle, \top), \\ ((\langle c_1 c_2 \rangle, \top) \end{matrix}\right\} \quad f^*_{(m,c_2)}(\xi)$$

Edge effects $f^*_{(m,n)}$

Definition: Let $(m,n) \in E^*,\ f_{(m,n)} \in F, \xi \in L^*$.



$$\xi = \left\{ \begin{array}{l} (\langle c_1 \rangle, 1), \\ (\langle c_1 c_2 \rangle, \top) \end{array} \right\}$$

$$f^*_{(m,c_2)}(\xi) = \left\{ \begin{array}{l} (\langle c_1 \rangle, \top), \\ (\langle c_1 c_2 \rangle, \top) \end{array} \right\}$$

$\xi(\langle c_1 \rangle)$

$f^*_{(m,c_2)}(\xi)(\langle c_1 \rangle)$

$\forall\, \gamma \in \Gamma$

$$f^*_{(m,n)}(\xi)(\gamma) := \begin{cases} f_{(m,n)}\big(\xi(\gamma')\big) & \text{if } \exists \gamma' \text{ with } \gamma = \gamma' \circ (m,n) \\ \bot & \text{else} \end{cases}$$

Edge effects domain F^*

Define $F^* \subseteq L^* \to L^*$ such that

1. F^* smallest set containing
$$\left\{f^*_{(m,n)} \middle| (m,n) \in E^*\right\} \cup \{id_{L^*}\}$$

2. F^* closed under functional composition and $\wedge$

Lemma: Properties of F^*

1. $F \begin{Bmatrix} \text{monotone} \\ \text{distributive} \end{Bmatrix}$ in $L \Rightarrow F^* \begin{Bmatrix} \text{monotone} \\ \text{distributive} \end{Bmatrix}$ in L^*

2. F distributive in $L \Rightarrow F^*$ **continuous** in L^*:

   For an infinite collection $\{\xi_k\}_{k\geq 1} \subseteq L^*$ and $(m,n) \in E^*$:

$$f^*_{(m,n)}\left(\bigwedge_k \xi_k\right) = \bigwedge_k f^*_{(m,n)}(\xi_k)$$

Dataflow equations x_n^*

- The dataflow problem for (F^*, L^*):
  - $x_{r_{main}}^* = \{(\lambda, \top)\}$
  - $x_n^* = \bigwedge_{(m,n)\in E^*} \left(f_{(m,n)}^*(x_m^*) \right) \qquad n \in N^* - \{r_{main}\}$
- **Claim:**
  ∃ maximum fixed-point solution (***MFP***$_{CS}$) for $\{x_n^*\}_{n\in N^*}$

Proof: Existence of max. FP solution

- Iteratively solving the equations yields decreasing (F^* monotone) chains
$$x_n^{*(i)} \geq x_n^{*(i+1)} \quad \forall n \in N^*.$$
Then $\forall \gamma \in \Gamma$
$$x_n^{*(i)}(\gamma) \geq x_n^{*(i+1)}(\gamma)$$
a decreasing chain in L with limit $\lim x_n^*(\gamma)$ (exists as L bounded).

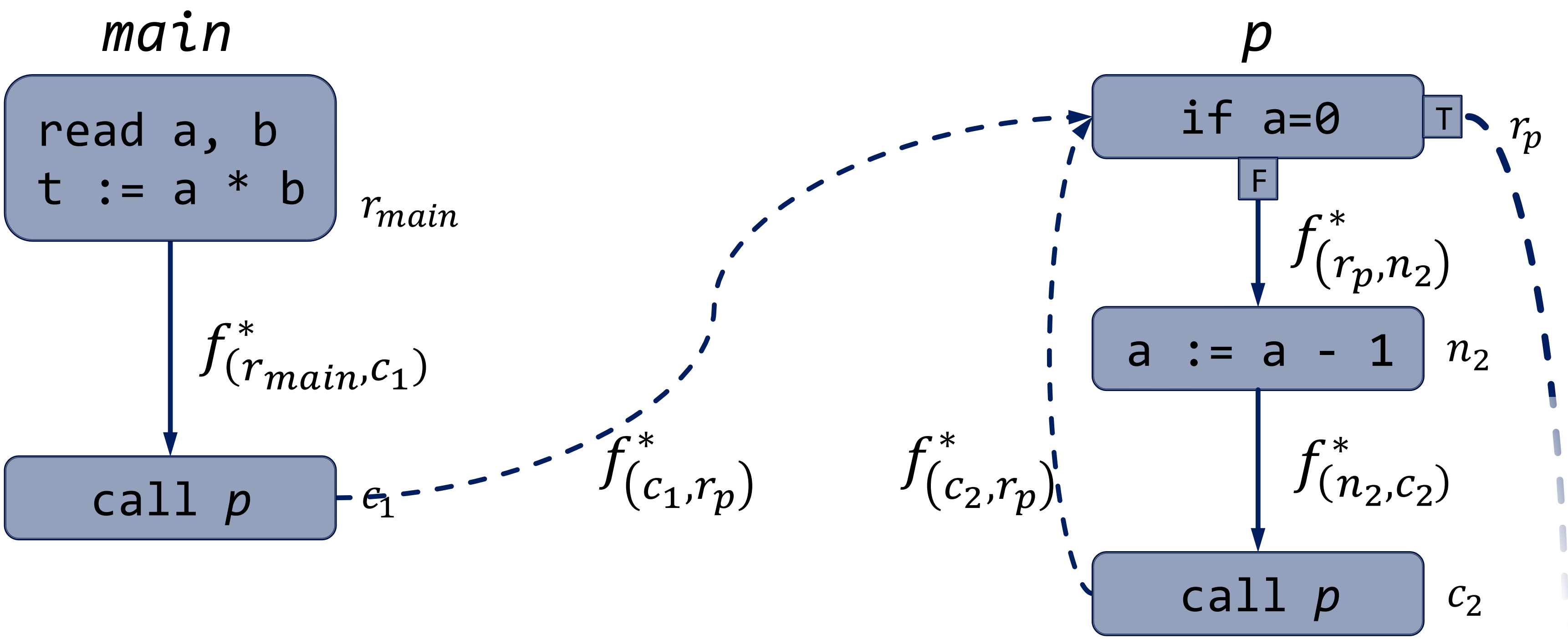
- $x_n^* \coloneqq \lim_{i \to \infty} x_n^{*(i)}$ possibly infinite but well-defined

$\Rightarrow \{x_n^*\}_{n \in N^*}$ is maximal FP solution to DF problem.

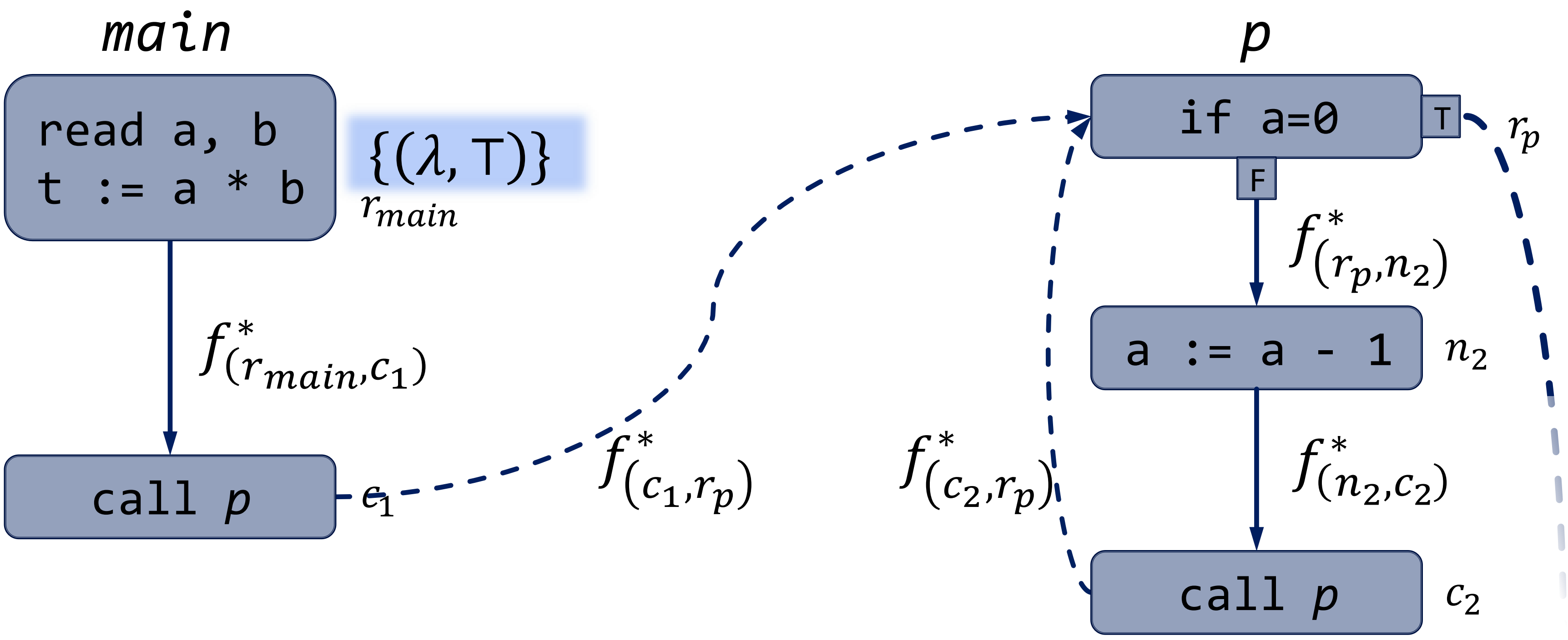
Maximum FP solution of (L^*, F^*)

- Initialise $\{x_n^*\}_{n \in N^*}$ with
  - $x_{r_{main}}^{*(0)} = \{(\lambda, \top)\}$
  - $x_n^{*(0)} = \bot^*$ $\quad n \in N^* - \{r_{main}\}$
- Result if Γ
  - **finite**: Any iterative algorithm terminates and reaches MFP.
  - **infinite**: Iteration may diverge, intermediate steps *unsafe* approx. of MFP

Example

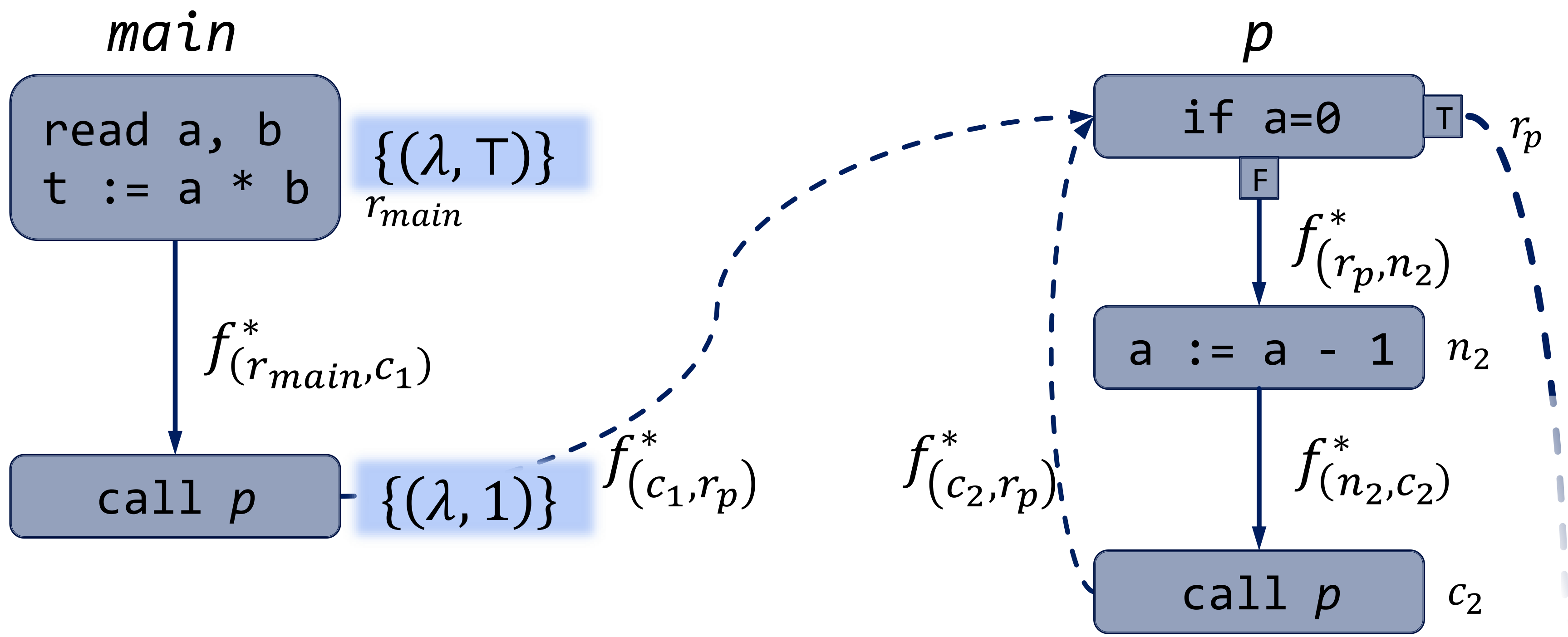

main
read a, b
t := a * b
r_{main}
$f^*_{(r_{main},c_1)}$
call p
c_1
$f^*_{(c_1,r_p)}$
p
if a=0
T
F
r_p
$f^*_{(r_p,n_2)}$
a := a - 1
n_2
$f^*_{(c_2,r_p)}$
$f^*_{(n_2,c_2)}$
call p
c_2

Example



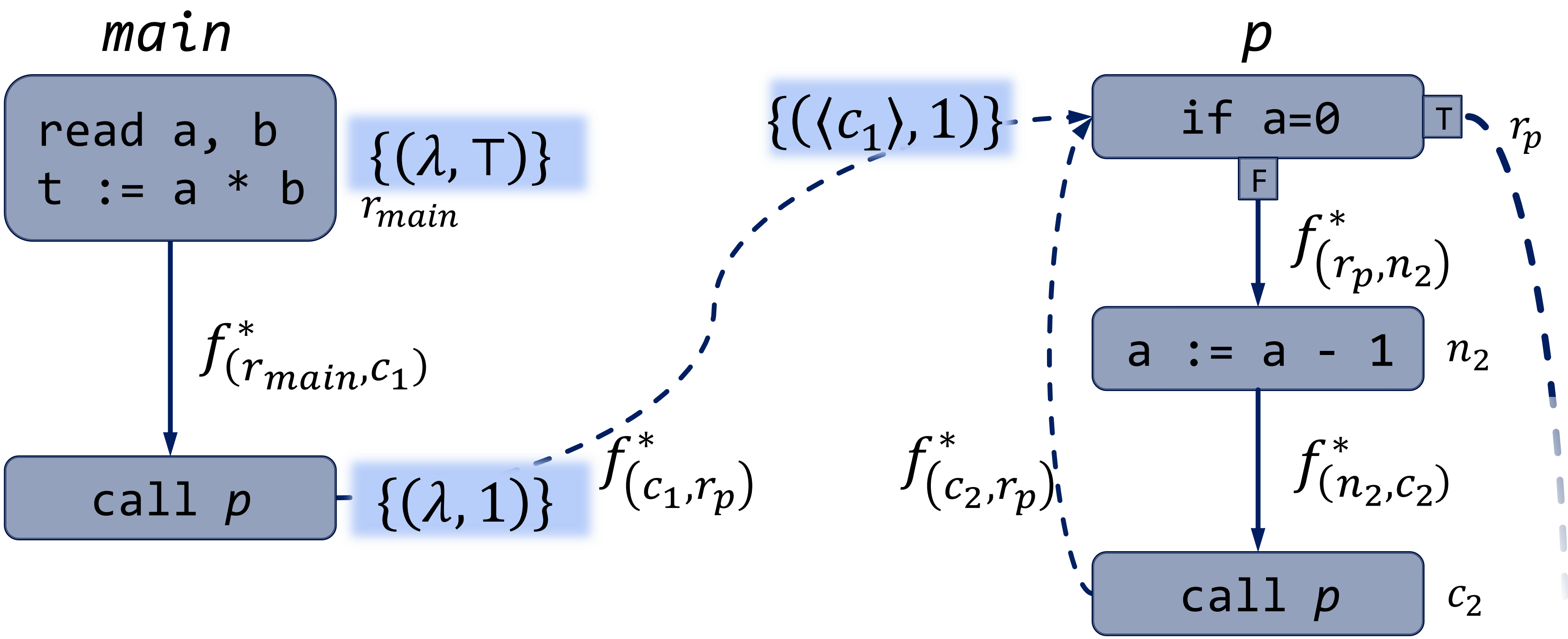
main
read a, b
t := a * b
{(λ, T)}
r_{main}
$f^*_{(r_{main},c_1)}$
call p
c_1
$f^*_{(c_1,r_p)}$
p
if a=0
T
F
r_p
$f^*_{(r_p,n_2)}$
a := a - 1
n_2
$f^*_{(c_2,r_p)}$
$f^*_{(n_2,c_2)}$
call p
c_2

Example



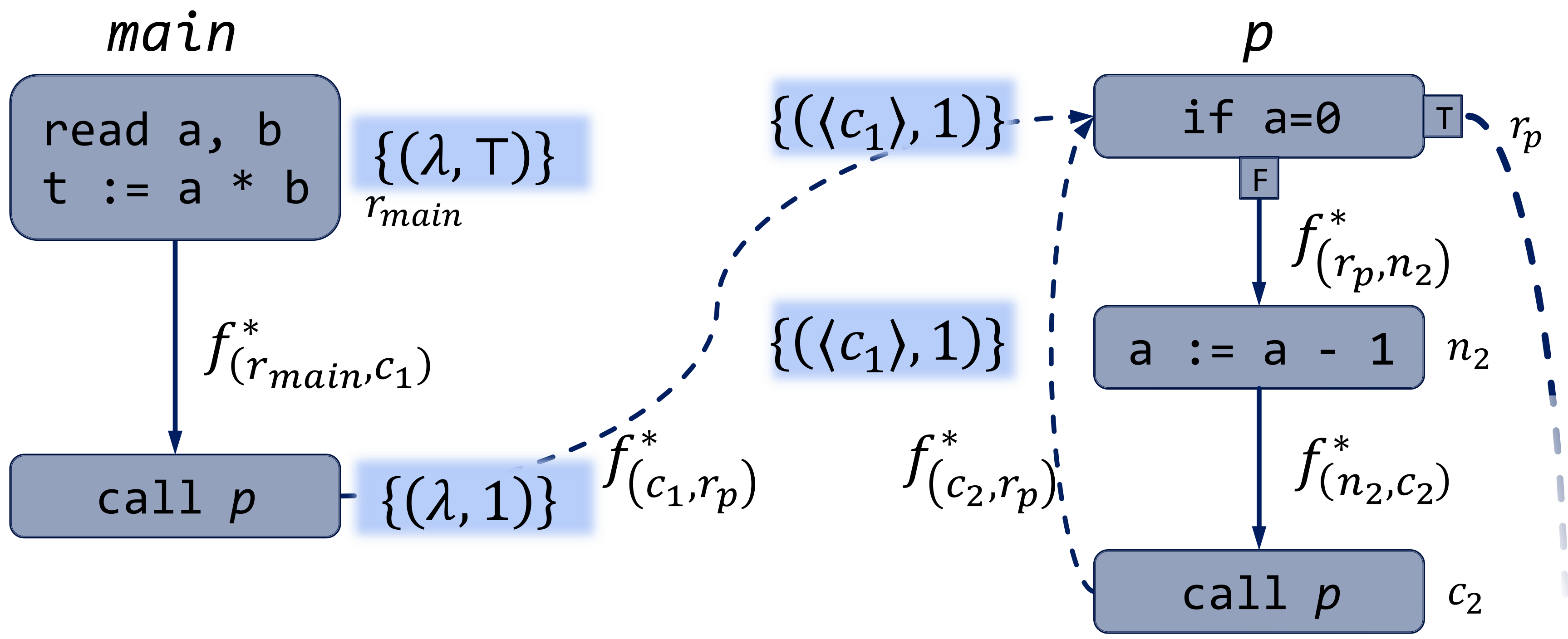
main
read a, b
t := a * b
{(λ, T)}
r_{main}
$f^*_{(r_{main},c_1)}$
call p
{(λ, 1)}
$f^*_{(c_1,r_p)}$
p
if a=0
T
r_p
F
$f^*_{(r_p,n_2)}$
a := a - 1
n_2
$f^*_{(c_2,r_p)}$
$f^*_{(n_2,c_2)}$
call p
c_2

Example



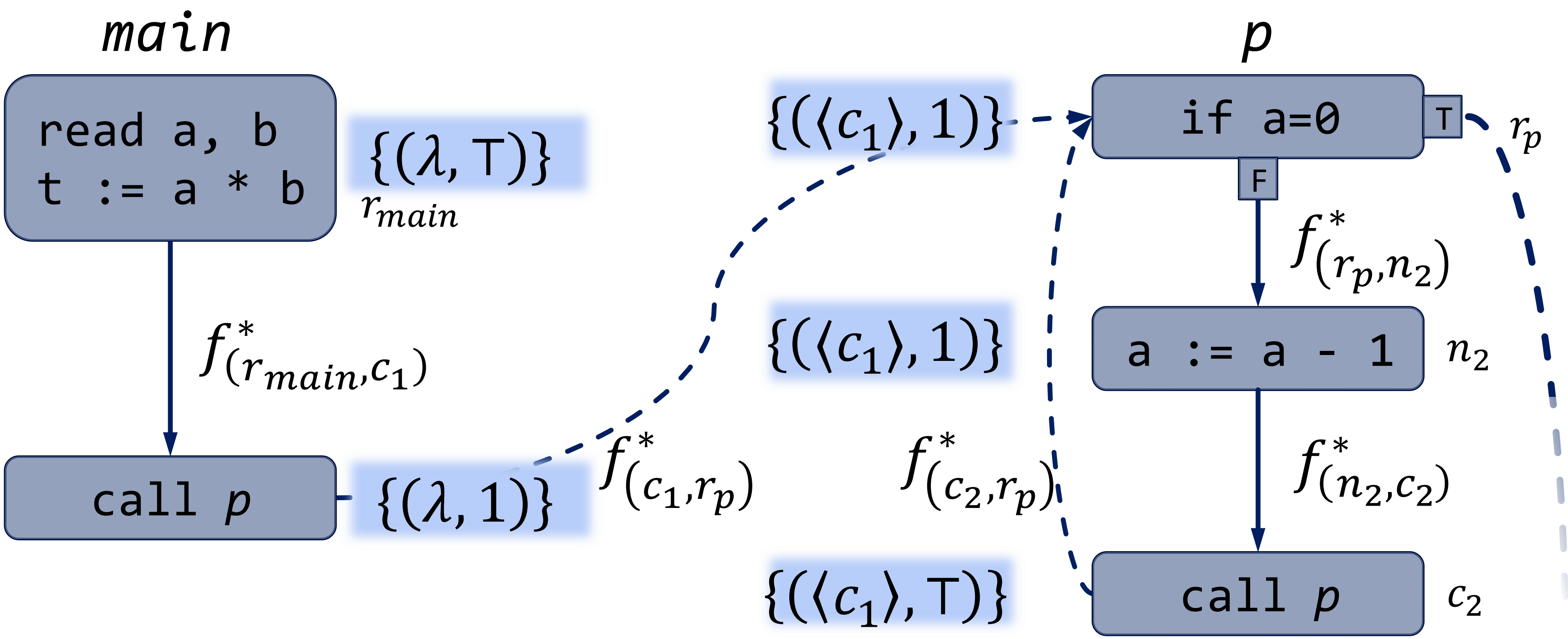
main
read a, b
t := a * b
{(λ, T)}
r_{main}
$f^*_{(r_{main},c_1)}$
call p
{(λ, 1)}
$f^*_{(c_1,r_p)}$
{((⟨c₁⟩, 1)}
p
if a=0
T
F
r_p
$f^*_{(r_p,n_2)}$
a := a - 1
n_2
$f^*_{(n_2,c_2)}$
$f^*_{(c_2,r_p)}$
call p
c_2

Example



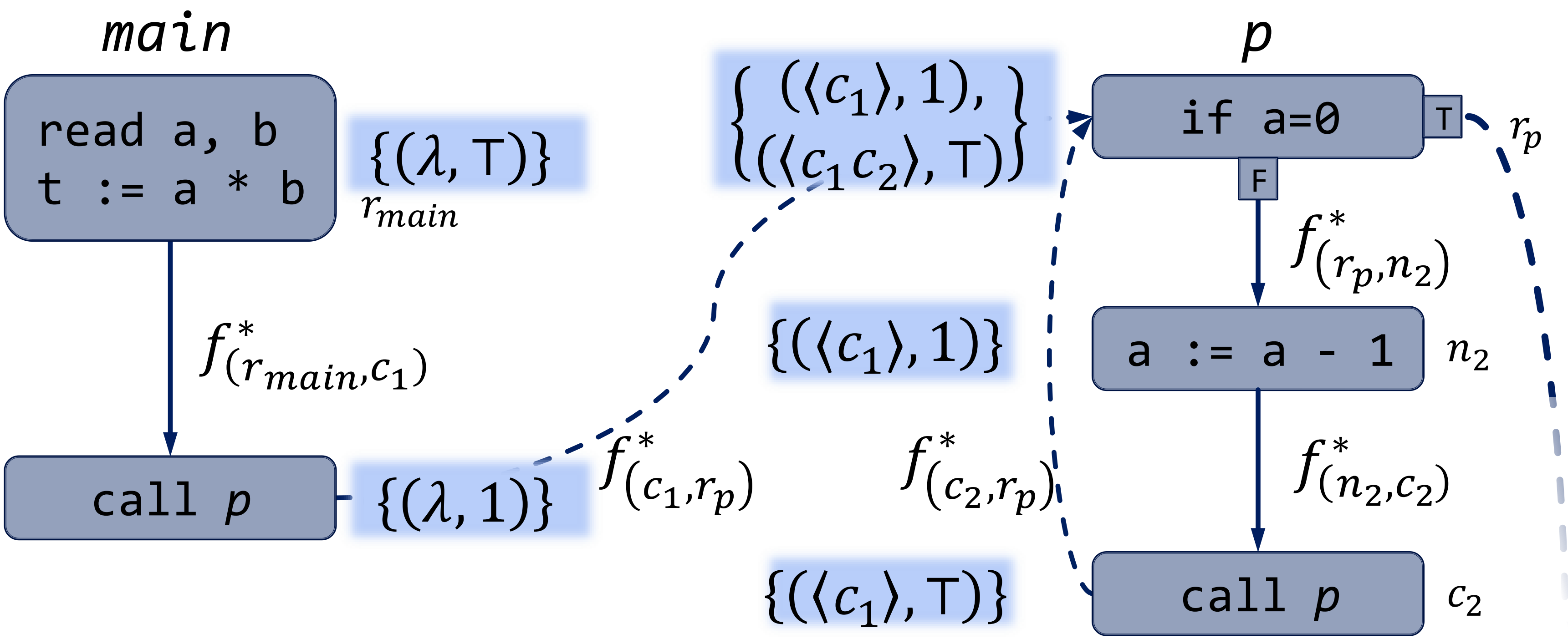
main
read a, b
t := a * b
$\{(\lambda, \top)\}$
r_{main}
$f^*_{(r_{main},c_1)}$
call p
$\{(\lambda, 1)\}$
$f^*_{(c_1,r_p)}$
$\{(\langle c_1 \rangle, 1)\}$
$\{(\langle c_1 \rangle, 1)\}$
$f^*_{(c_2,r_p)}$
p
if a=0
T
r_p
F
$f^*_{(r_p,n_2)}$
a := a - 1
n_2
$f^*_{(n_2,c_2)}$
call p
c_2

Example

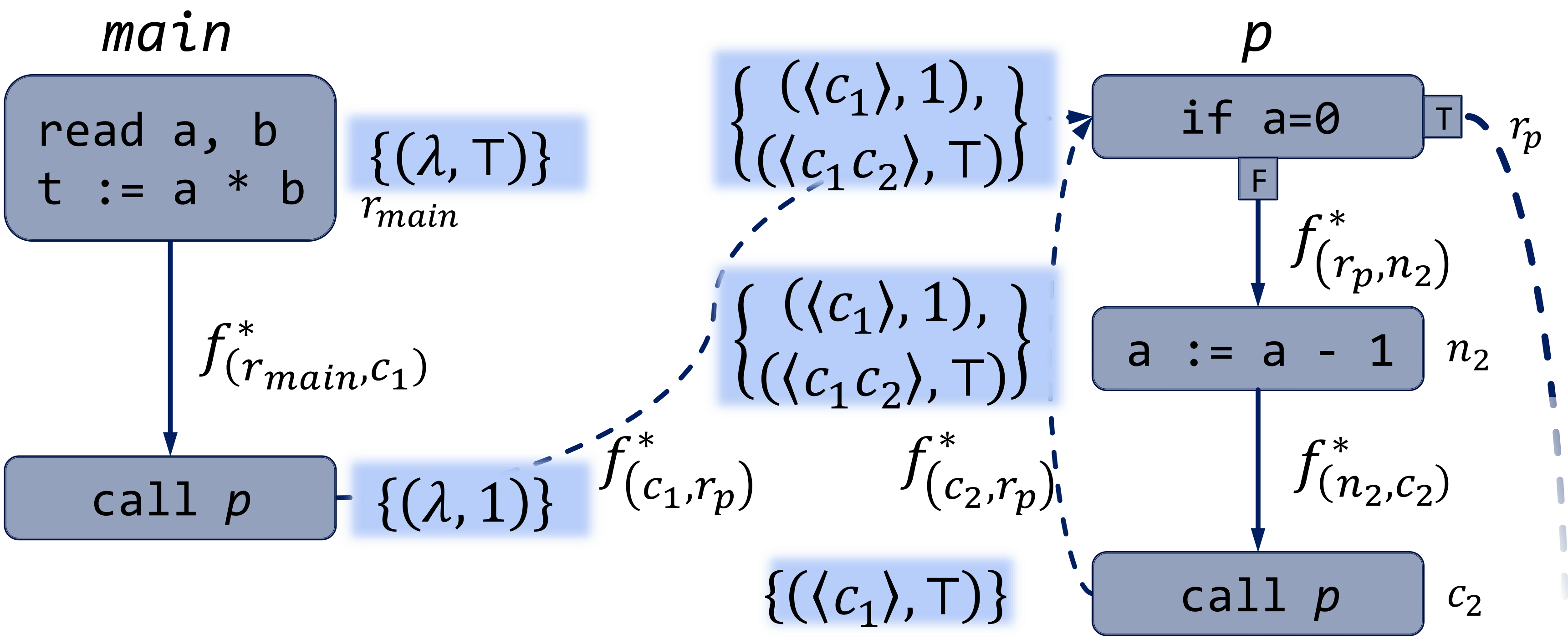


main
p
read a, b
t := a * b
{(λ, T)}
r_main
f*(r_main, c1)
call p
{(λ, 1)}
f*(c1, rp)
{((⟨c1⟩, 1)}
{((⟨c1⟩, 1)}
f*(c2, rp)
{((⟨c1⟩, T)}
if a=0
T
F
rp
f*(rp, n2)
a := a - 1
n2
f*(n2, c2)
call p
c2

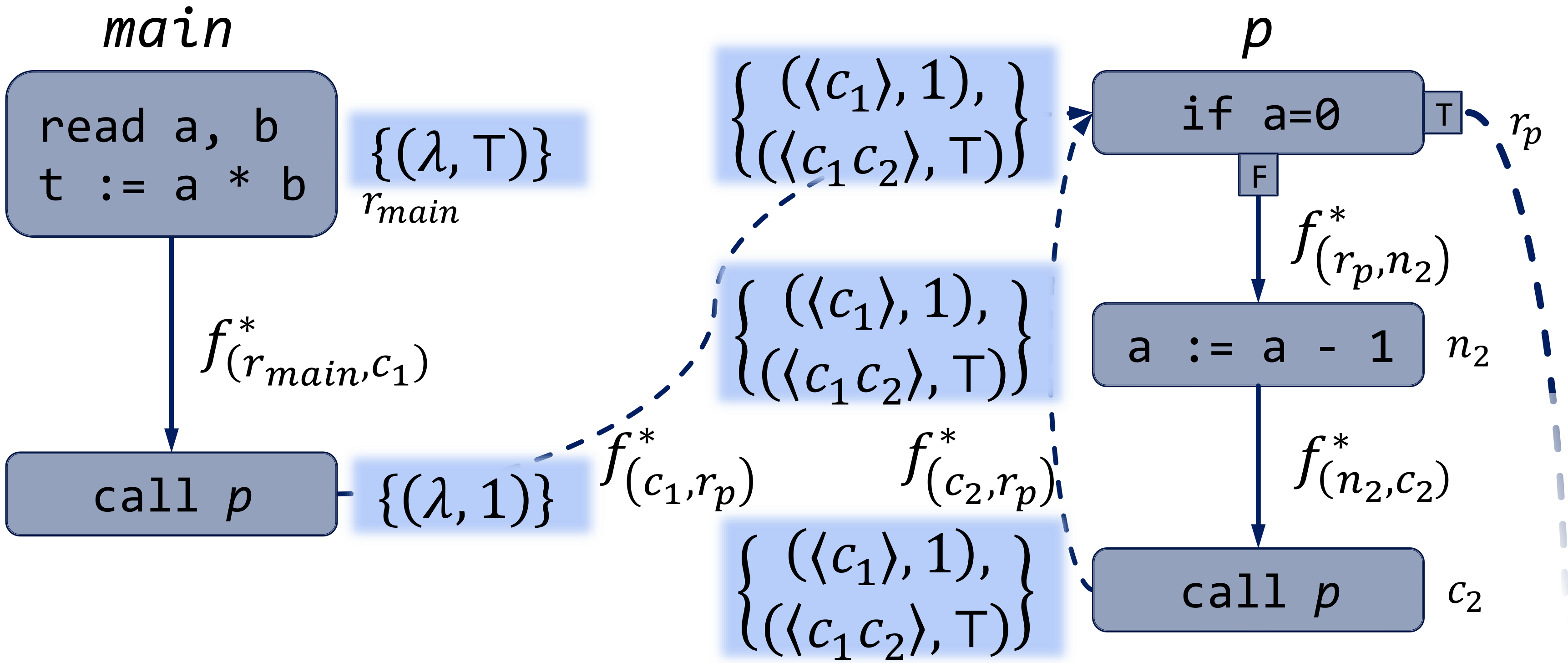
Example


main
read a, b
t := a * b
{(λ, ⊤)}
r_{main}
$f^*_{(r_{main},c_1)}$
call p
{(λ, 1)}
$f^*_{(c_1,r_p)}$
{ ((⟨c_1⟩, 1), ((⟨$c_1 c_2$⟩, ⊤)) }
{((⟨c_1⟩, 1)}
{((⟨c_1⟩, ⊤)}
$f^*_{(c_2,r_p)}$
p
if a=0
T
F
r_p
$f^*_{(r_p,n_2)}$
a := a - 1
n_2
$f^*_{(n_2,c_2)}$
call p
c_2

Example

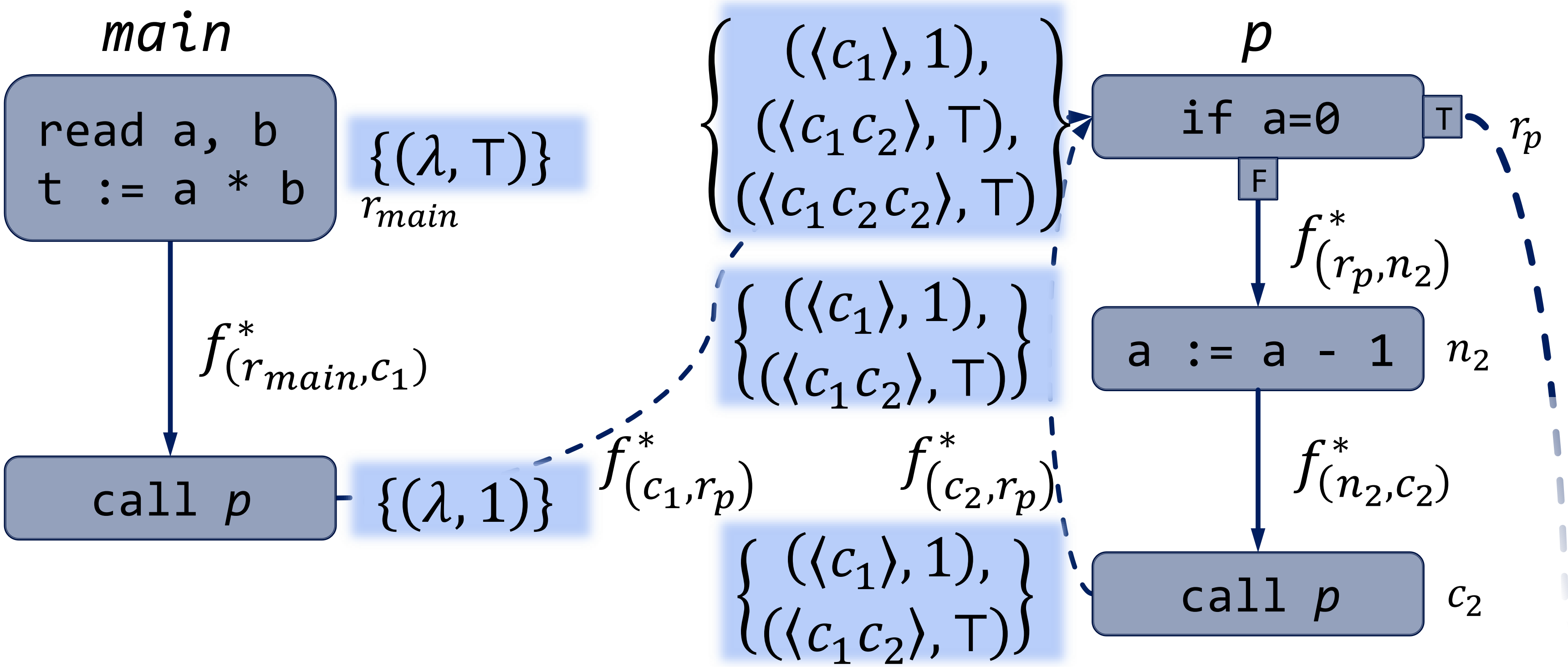

main
read a, b
t := a * b
$\{(\lambda, \top)\}$
r_{main}
$f^*_{(r_{main}, c_1)}$
call p
$\{(\lambda, 1)\}$
$f^*_{(c_1, r_p)}$
$\left\{ \begin{array}{c} (\langle c_1 \rangle, 1), \\ ((\langle c_1 c_2 \rangle, \top) \end{array} \right\}$
$\left\{ \begin{array}{c} (\langle c_1 \rangle, 1), \\ ((\langle c_1 c_2 \rangle, \top) \end{array} \right\}$
$f^*_{(c_2, r_p)}$
$\{(\langle c_1 \rangle, \top)\}$
p
if a=0
T
F
r_p
$f^*_{(r_p, n_2)}$
a := a - 1
n_2
$f^*_{(n_2, c_2)}$
call p
c_2

Example



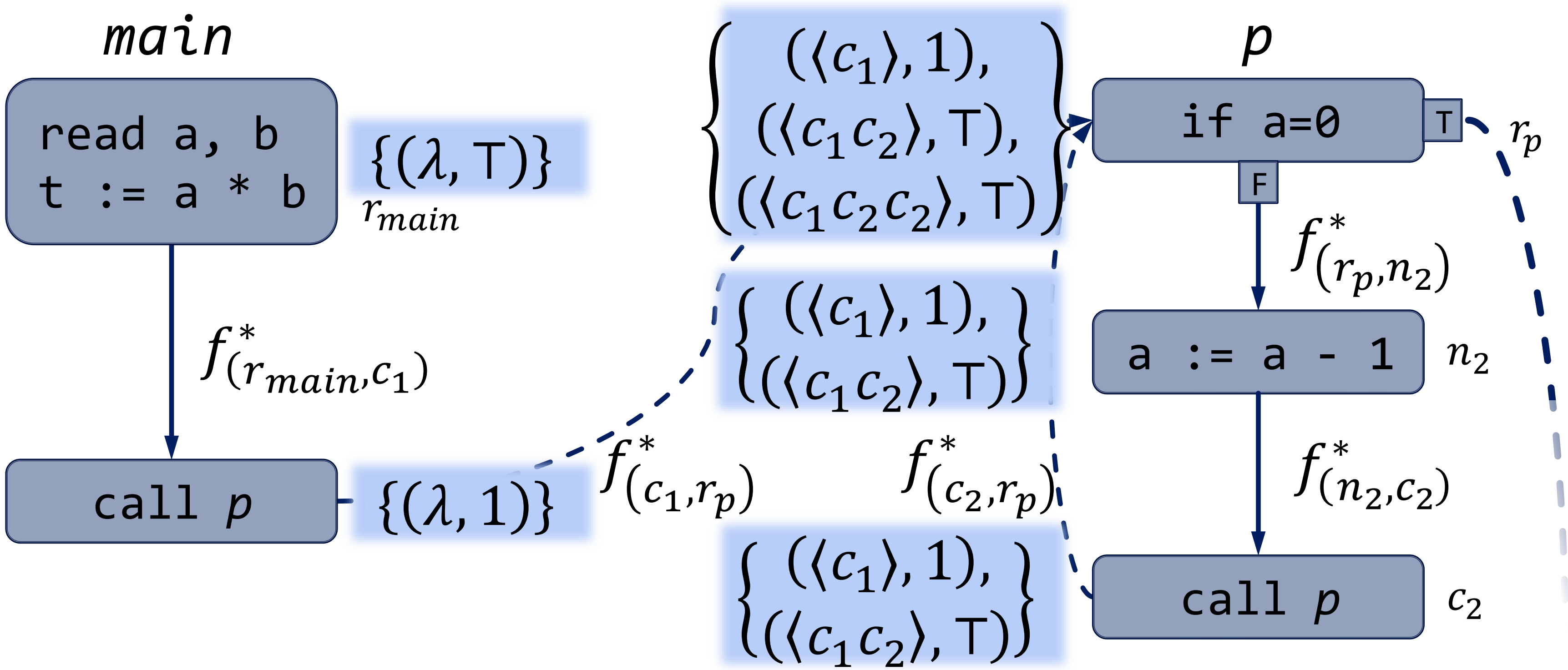
main
read a, b
t := a * b
{(λ, T)}
r_main
f*(r_main, c1)
call p
{(λ, 1)}
f*(c1, rp)
{ ((⟨c1⟩, 1), ((⟨c1 c2⟩, T)) }
{ ((⟨c1⟩, 1), ((⟨c1 c2⟩, T)) }
{ ((⟨c1⟩, 1), ((⟨c1 c2⟩, T)) }
f*(c2, rp)
p
if a=0
T
F
rp
f*(rp, n2)
a := a - 1
n2
f*(n2, c2)
call p
c2

Example



main
read a, b
t := a * b
$\{(\lambda, \mathsf{T})\}$
r_{main}
$f^*_{(r_{main}, c_1)}$
call p
$\{(\lambda, 1)\}$
$f^*_{(c_1, r_p)}$
$\left\{ \begin{array}{c} (\langle c_1 \rangle, 1), \\ (\langle c_1 c_2 \rangle, \mathsf{T}), \\ (\langle c_1 c_2 c_2 \rangle, \mathsf{T}) \end{array} \right\}$
$\left\{ \begin{array}{c} (\langle c_1 \rangle, 1), \\ (\langle c_1 c_2 \rangle, \mathsf{T}) \end{array} \right\}$
$f^*_{(c_2, r_p)}$
$\left\{ \begin{array}{c} (\langle c_1 \rangle, 1), \\ (\langle c_1 c_2 \rangle, \mathsf{T}) \end{array} \right\}$
p
if a=0
T
F
r_p
$f^*_{(r_p, n_2)}$
a := a - 1
n_2
$f^*_{(n_2, c_2)}$
call p
c_2

Example



main
read a, b
t := a * b
$\{(\lambda, \top)\}$
r_{main}
$f^*_{(r_{main},c_1)}$
call p
$\{(\lambda, 1)\}$
$f^*_{(c_1,r_p)}$
$\left\{ (\langle c_1\rangle, 1), (\langle c_1 c_2\rangle, \top), (\langle c_1 c_2 c_2\rangle, \top) \right\}$
$\left\{ (\langle c_1\rangle, 1), (\langle c_1 c_2\rangle, \top) \right\}$
$f^*_{(c_2,r_p)}$
$\left\{ (\langle c_1\rangle, 1), (\langle c_1 c_2\rangle, \top) \right\}$
p
if a=0
T
r_p
F
$f^*_{(r_p,n_2)}$
a := a - 1
n_2
$f^*_{(n_2,c_2)}$
call p
c_2

...

Chapter summary

Call string approach:

- New dataflow problem on top of (L, F)
  - Data "tagged" with propagation history
  - Interprocedural flow explicit
- Existence of $\mathrm{MFP}_{\mathrm{CS}}$
  - No functional compositions
  - Only iteratively computable if Γ finite

Structure

1. Definition of a new DF problem (L^*, F^*)
2. **Proof: Solution to $(\boldsymbol{L}^*, \boldsymbol{F}^*) \equiv$ MOP solution**
3. Feasibility and precise variants
4. Approximative solution

MOP solution and MFP_{CS}

- This chapter: Show that MOP solution

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

can be acquired from MFP_{CS}.

- Chapter outline:
  1. Define MOP_{CS}
  2. Show $\text{MOP}_{\text{CS}} = \text{MFP}_{\text{CS}}$
  3. Extract results $x'_n \in L$ from MFP_{CS}
  4. Using MOP_{CS} and MFP_{CS}, show $x'_n = \text{MOP}$

MOP solution for call strings (MOP_{CS})

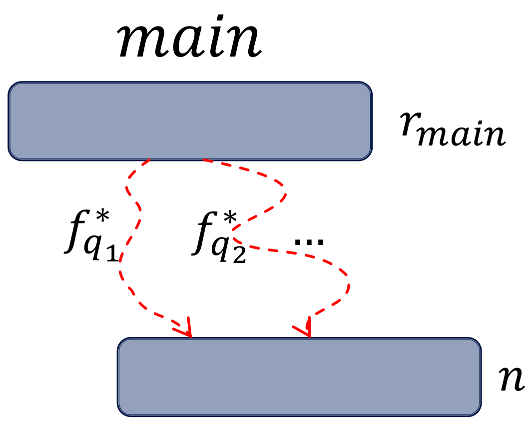
- Propagation of data along paths in G^*

  For $q = (r_{main}, s_1, \dots, s_k, n) \in path_{G^*}$:

  $$f_q^* \coloneqq f_{(s_k,n)}^* \circ f_{(s_{k-1},s_k)}^* \circ \cdots \circ f_{(s_1,s_2)}^* \circ f_{(r_{main},s_1)}^*$$

- Call string MOP solution (***MOP_{CS}***)

  $$\forall n \in N^*: \; y_n^* \coloneqq \bigwedge \left\{ f_q^*\left(x_{r_{main}}^*\right) : q \in path_{G^*}(r_{main}, n) \right\}$$



MOP solution for call strings (MOP_{CS})

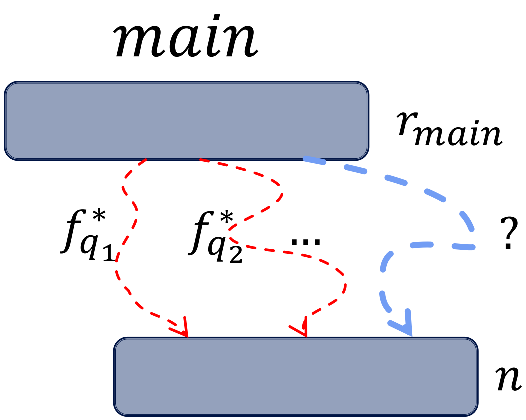
- Propagation of data along paths in G^*

  For $q = (r_{main}, s_1, \dots, s_k, n) \in path_{G^*}$:

  $$f_q^* \coloneqq f_{(s_k,n)}^* \circ f_{(s_{k-1},s_k)}^* \circ \cdots \circ f_{(s_1,s_2)}^* \circ f_{(r_{main},s_1)}^*$$

- Call string MOP solution (***MOP_{CS}***)

  $$\forall n \in N^*: \; y_n^* \coloneqq \bigwedge\left\{f_q^*\left(x_{r_{main}}^*\right) : q \in path_{G^*}(r_{main}, n)\right\}$$



$path_{G^*} \supseteq IVP$

Valid-path-only propagation

Lemma:

Let $n \in N^*, q \in path_{G^*}(r_{main}, n), \gamma \in \Gamma$. Then

1. $q \in IVP(r_{main}, n)$ and $CM(q) = \gamma$
   $\Leftrightarrow\ f_q^*\left(x_{r_{main}}^*\right)(\gamma)$ defined

2. $f_q^*\left(x_{r_{main}}^*\right)(\gamma) = f_q(\top)$

$\Rightarrow$ "MOP_{CS} only contains paths in IVP and for each such p yields the same result as (L, F)"

Valid-path-only propagation (Proof)

By induction on $l(q)$ for $q = (r_{main}, s_1, \dots, s_k, n)$:

- $\boldsymbol{l(q) = 0}$: Let $n \in N^*$.

  $q = (r_{main}) \in IVP(r_{main}, r_{main}) \Rightarrow CM(q) = \lambda$.

  $f_q^*(r_{main}) = \{(\lambda, \top)\}$ only def. for λ with $\top = f_q(\top)$.

- IH:

  Iff $q \in IVP(r_{main}, n), l(q) < k$ and $\gamma = CM(q)$

  then $f_q^*\left(x_{r_{main}}^*\right)(\lambda) = f_p(\top)$.

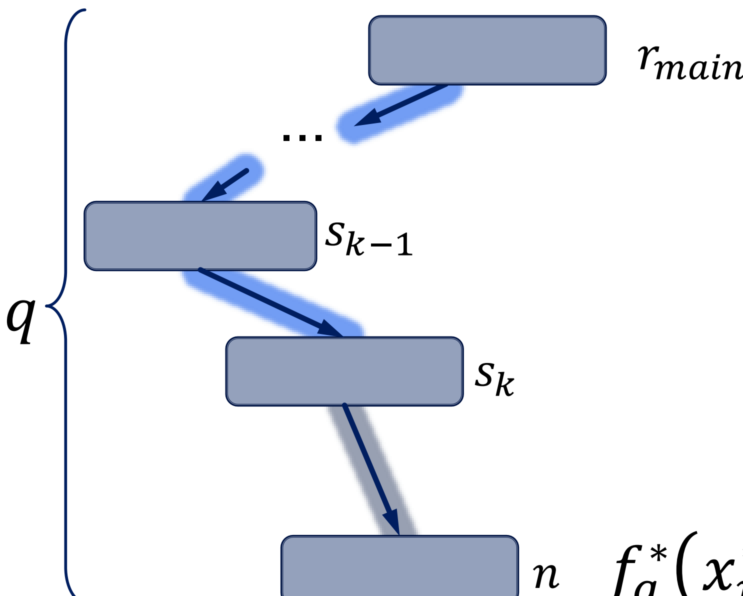
Valid-path-only propagation (Proof)

- **IH:**

  Iff $q \in IVP(r_{main}, n), l(q) < k$ and $\gamma = CM(q)$

  then $f_q^*\left(x_{r_{main}}^*\right)(\lambda) = f_p(\top)$

- **IS:** $l(q) = k$.



$$f_q^*\left(x_{r_{main}}^*\right)(\gamma) \stackrel{\text{def}}{=} f_{(s_k,n)}\left(x_{s_k}^*\right)$$

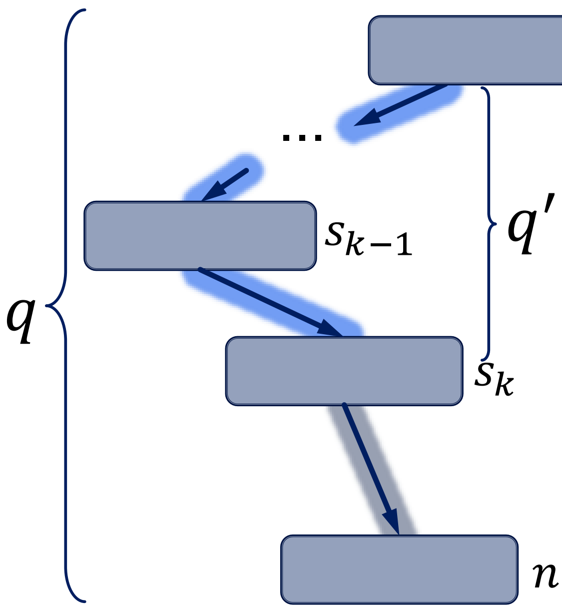
Valid-path-only propagation (Proof)

- **IH:**

  Iff $q \in IVP(r_{main}, n), l(q) < k$ and $\gamma = CM(q)$

  then $f_q^*\left(x_{r_{main}}^*\right)(\lambda) = f_p(\top)$

- **IS:** $l(q) = k$.



$\gamma = \gamma' \circ (m, n), \qquad \gamma' = CM(q')$

$f_{q'}^*\left(x_{r_{main}}^*\right)(\gamma')$

$f_q^*\left(x_{r_{main}}^*\right)(\gamma) \stackrel{\text{def}}{=} f_{(s_k, n)}\left(x_{s_k}^*\right)$

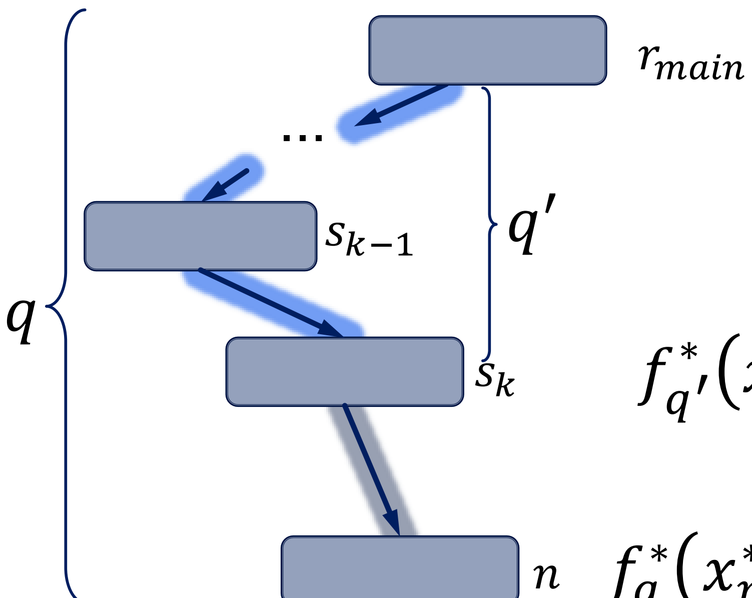
Valid-path-only propagation (Proof)

- **IH:**

  Iff $q \in IVP(r_{main}, n), l(q) < k$ and $\gamma = CM(q)$

  then $f_q^*(x_{r_{main}}^*)(\lambda) = f_p(\top)$

- **IS:** $l(q) = k$.



$$\gamma = \gamma' \circ (m, n), \qquad \gamma' = CM(q')$$

$$f_{q'}^*(x_{r_{main}}^*)(\gamma') =^{IH} f_{q'}(\top)$$

$$f_q^*(x_{r_{main}}^*)(\gamma) \stackrel{\text{def}}{=} f_{(s_k, n)}(x_{s_k}^*)$$

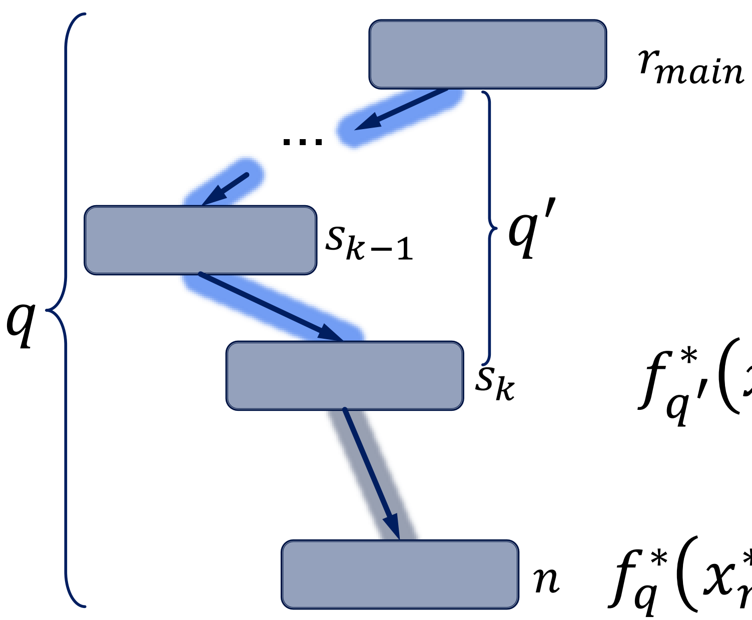
Valid–path–only propagation (Proof)

- **IH:**

  Iff $q \in IVP(r_{main}, n), l(q) < k$ and $\gamma = CM(q)$

  then $f_q^*\left(x_{r_{main}}^*\right)(\lambda) = f_p(\top)$

- **IS:** $l(q) = k$.



$$\gamma = \gamma' \circ (m, n), \qquad \gamma' = CM(q')$$

$$f_{q'}^*\left(x_{r_{main}}^*\right)(\gamma') =^{IH} f_{q'}(\top)$$

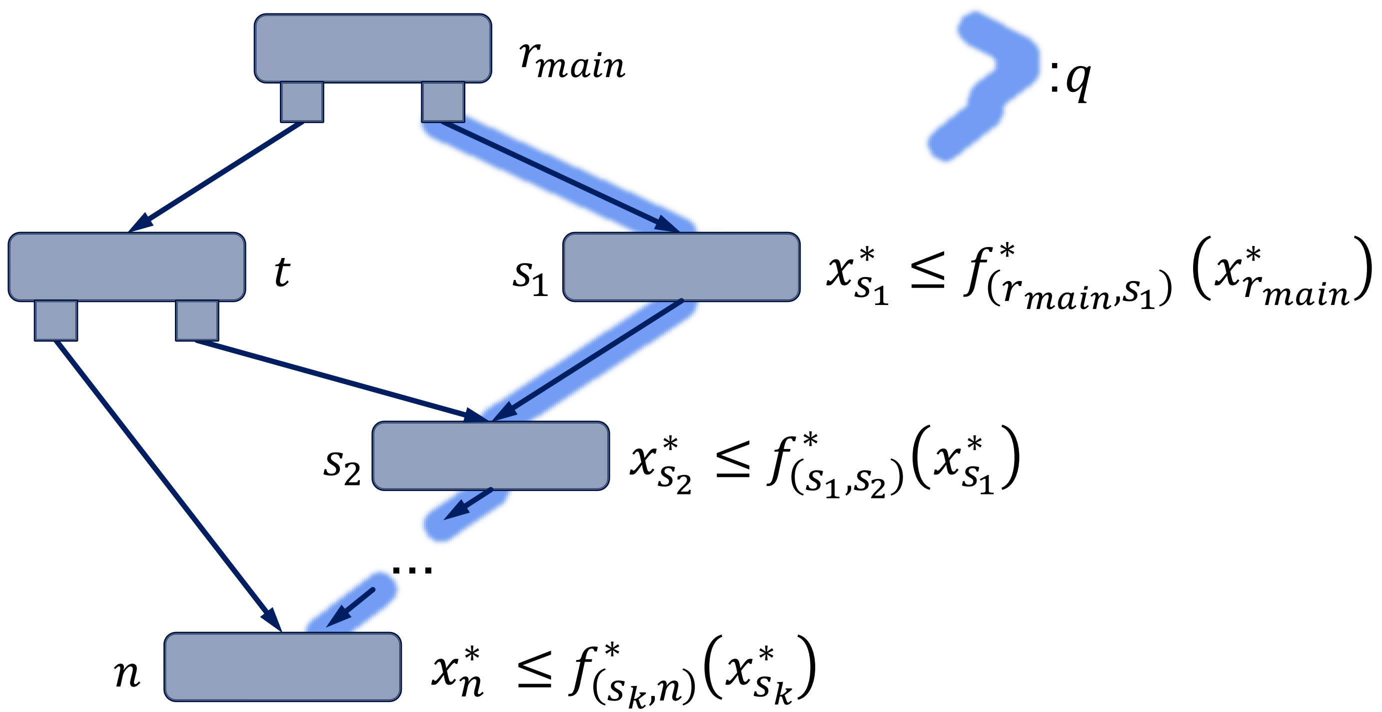
$$f_q^*\left(x_{r_{main}}^*\right)(\gamma) = f_{(s_k, n)}\left(f_{q'}(\top)\right) \stackrel{\text{def}}{=} f_q(\top) \quad \blacksquare$$

MFP$_{CS}$ = MOP$_{CS}$: $x_n^* = y_n^*$

Theorem: (L,F) distributive $\Rightarrow \forall n \in N^*$: $x_n^* = y_n^*$

Proof:

- $\boldsymbol{x_n^* \le y_n^*}$: Let $q \in path_{G^*} = (r_{main}, s_1, \dots, s_k, n)$.

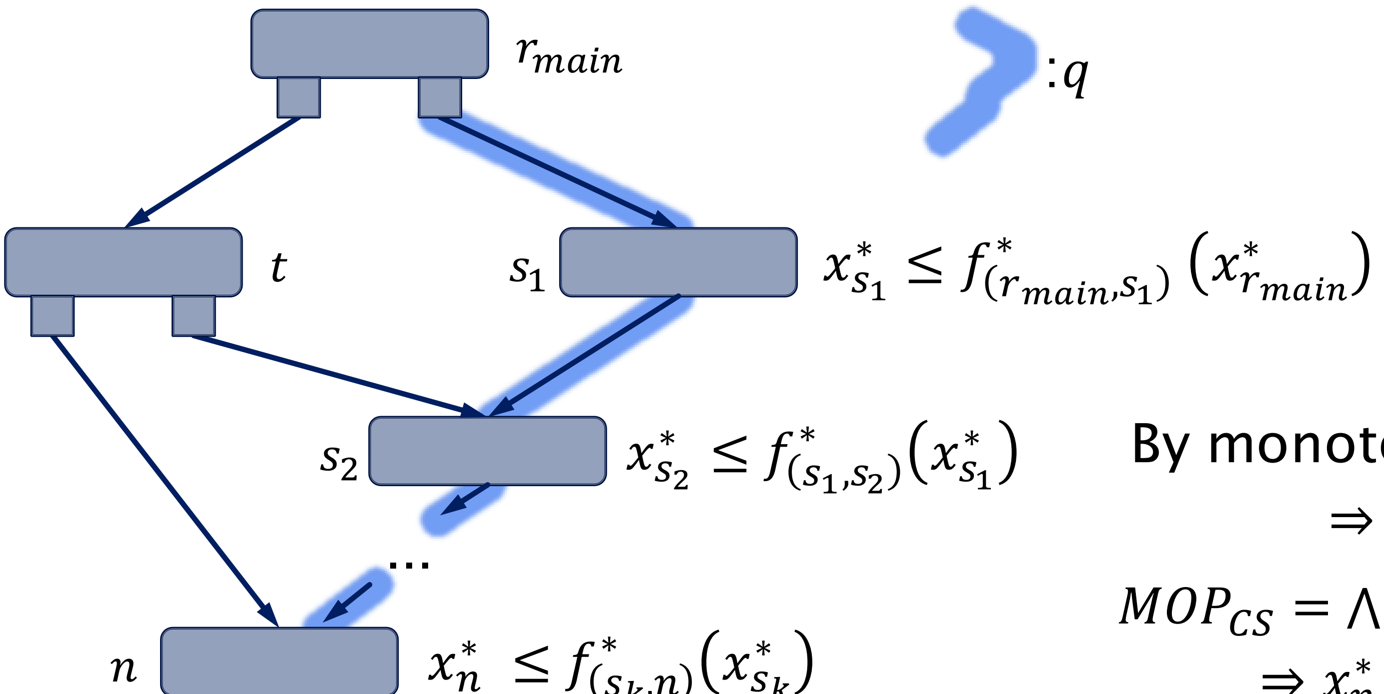


$\text{MFP}_{CS} = \text{MOP}_{CS}$: $x_n^* = y_n^*$

Theorem: (L, F) distributive $\Rightarrow \forall n \in N^*$: $x_n^* = y_n^*$

Proof:

- $\boldsymbol{x_n^* \leq y_n^*}$: Let $q \in path_{G^*} = (r_{main}, s_1, \ldots, s_k, n)$.



By monotonicity

$$\Rightarrow x_n^* \leq f_q^*\left(x_{r_{main}}^*\right)$$

$$MOP_{CS} = \bigwedge_q f_q^*$$

$$\Rightarrow x_n^* \leq \bigwedge_q f_q^*\left(x_{r_{main}}^*\right)$$

$\text{MFP}_{\text{CS}} = \text{MOP}_{\text{CS}}$: $x_n^* = y_n^*$

- $\boldsymbol{x_n^* \geq y_n^*}$:
  By induction on i: $\forall i \geq 0, n \in N^*$: $x_n^{*(i)} \geq y_n^*$
  - $\boldsymbol{i = 0}$:

    $$n = r_{main} : x_{r_{main}}^{*(0)} \overset{\text{def}}{\equiv} \{(\lambda, \top)\} = f_{q_0}^*\left(x_{r_{main}}^*\right) \geq y_{r_{main}}^*$$

    $$n \neq r_{main} : x_n^{*(0)} \overset{\text{def}}{\equiv} \bot^* \geq y_n^*$$
  - IH:

    $$\forall i \leq k, n \in N^*: x_n^{*(i)} \geq y_n^*$$
  - IS:

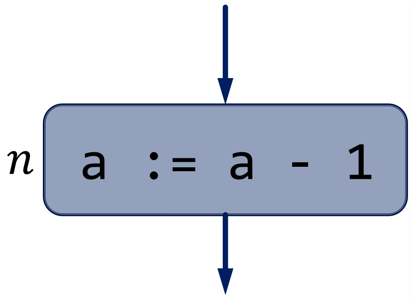
$$x_n^{*(k+1)} \overset{\text{def}}{\equiv} \bigwedge_{(m,n) \in E^*} f_{(m,n)}^*\left(x_m^{*(k)}\right) \overset{\text{IH}}{\geq} \bigwedge_{(m,n) \in E^*} f_{(m,n)}^*\left(y_m^*\right) \overset{\substack{\text{Cont.+Dist.} \\ \text{of } f_{(m,n)}^*}}{\geq} y_n^*$$

■

Result for (L, F) from MFP_{CS}

Define $x'_n \in L$ as follows:

$$x'_n := \bigwedge_{\gamma \in \Gamma} x^*_n(\gamma)$$



$$\left\{\begin{matrix} (\langle c_1 \rangle, 1), \\ (\langle c_1 c_2 \rangle, \top), \\ (\langle c_1 c_2 c_2 \rangle, \top), \\ \ldots \end{matrix}\right\} = x^*_n$$

$$\begin{aligned} x'_n &= x^*_n(\lambda) \wedge x^*_n(\langle c_1 \rangle) \wedge \\ &\quad x^*_n(\langle c_1 c_2 \rangle) \wedge \cdots \\ &= \bot \wedge 1 \wedge \top \wedge \cdots \\ &= \quad 1 \wedge \top \wedge \cdots \\ &= \top \end{aligned}$$

Claim: x'_n is the MOP solution in (L, F)

$MFP_{CS} \equiv MOP$

Theorem: For each $n \in N^*$: $x'_n = y_n$

Proof:

$$x'_n = \bigwedge_{\gamma \in \Gamma} x^*_n(\gamma)$$ Def x'_n

$$= \bigwedge_{\gamma \in \Gamma} \{ f^*_q(x^*_{r_{main}})(\gamma) \mid q \in path_{G^*}(r_{main}, n) \}$$ MFP_{CS}= MOP_{CS}

$$= \bigwedge_{\gamma \in \Gamma} \{ f_q(\top) \mid q \in IVP(r_{main}, n) \text{ sth. } CM(q) = \gamma \}$$ Only-valid-path propag. Lemma

$$= \bigwedge \{ f_q(\top) \mid q \in IVP(r_{main}, n) \}$$ $CM^{-1}(\Gamma) \cap IVP = IVP$

$$= y_n$$ ■

Chapter summary

- Shown:
  - Get MOP solution from MFP_{CS}
  - Conversion $x_n^* \rightarrow x_n'$ straightforward and without functional composition

$\Rightarrow$ Call string approach is actually useful

- But: Computing MFP_{CS} still problematic

Structure

1. Definition of a new DF problem (L^*, F^*)
2. Proof: Solution to $(L^*, F^*) \equiv$ MOP solution
3. **Feasibility and precise variants**
4. Approximative solution

Motivation

- Observation: L finite $\Rightarrow$ functional approach converges, CS approach not necessarily
- Idea: Ensure termination of CS approach by limiting to finite CS subset $\Gamma_0 \subseteq \Gamma$
- Show: Can be done for all (L, F) with finite L without losing precision

Redefinitions using Γ_o

- Γ_0 is finite subset of Γ fulfilling:
  If $\gamma \in \Gamma_0$ and γ' initial subtuple of $\gamma \Rightarrow \gamma' \in \Gamma_0$
- $IVP' := \{q \in IVP \mid \forall \text{ prefix } q' \text{of } q\text{: } CM(q') \in \Gamma_0\}$
- Let $\circ_0$ only act in Γ_0
  - discards γ' entirely iff $\gamma' \circ_0 (m,n) \notin \Gamma_0$
  - $\circ_0$ is consistent with IVP'

Definitions: DF analysis with Γ_0

Now consider (L_0^*, F_0^*) with $L_0^* \coloneqq \Gamma_0 \to L$ finite

- Dataflow equations now iteratively solvable

$$x'_{n_0} \coloneqq \bigwedge_{\gamma \in \Gamma_0} x_n^*(\gamma) \qquad \textbf{(}\boldsymbol{MFP_{CS0}}\textbf{)}$$

- MOP solution using only paths in IVP'

$$y'_{n_0} \coloneqq \bigwedge \{ f_q(0) \mid q \in IVP'(r_{main}, n) \} \qquad \textbf{(}\boldsymbol{MOP_{CS0}}\textbf{)}$$

$MOP_{CS0} = MFP_{CS0}$

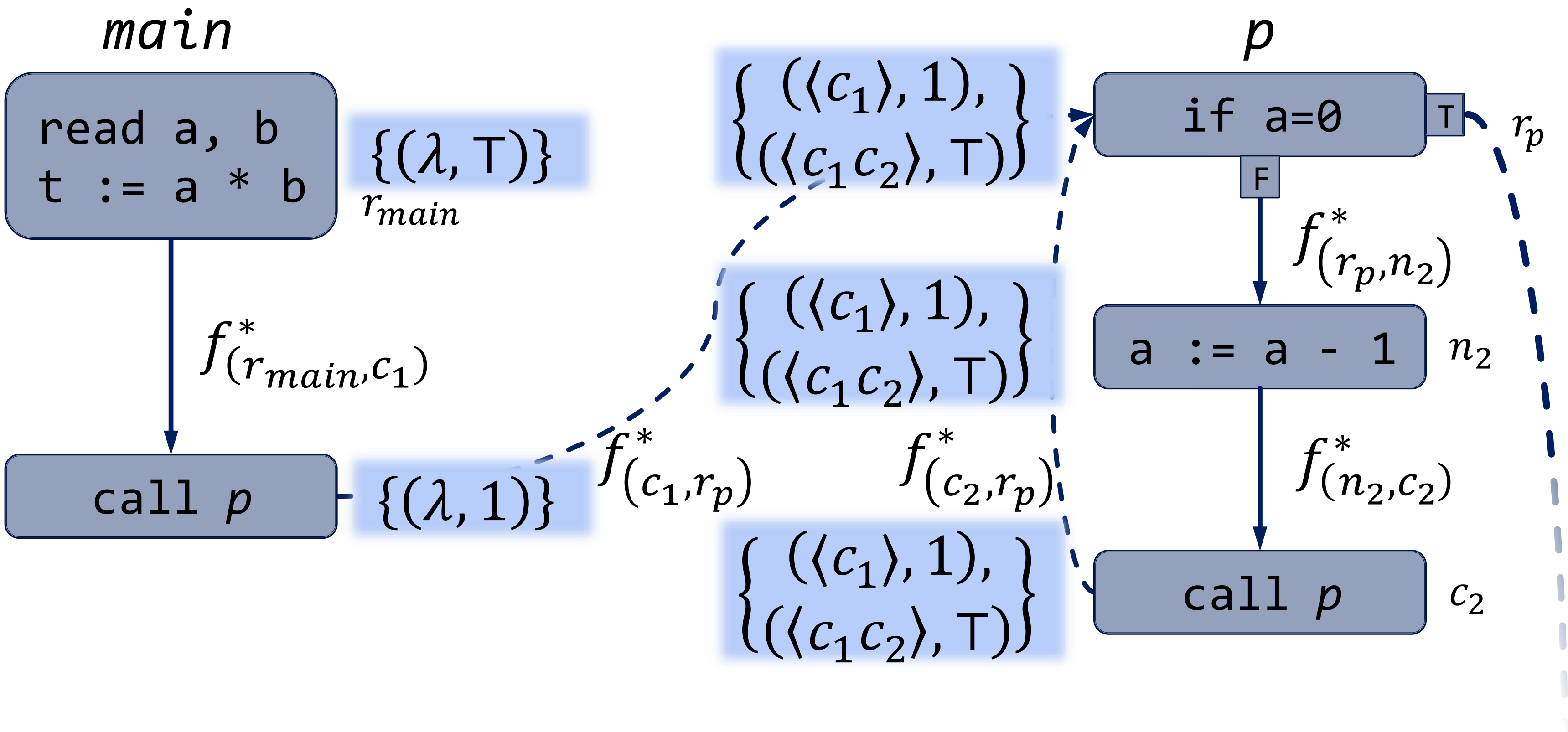
Theorem: $\forall n \in N^*: x'_{n_0} = y'_{n_0}$

Proof:

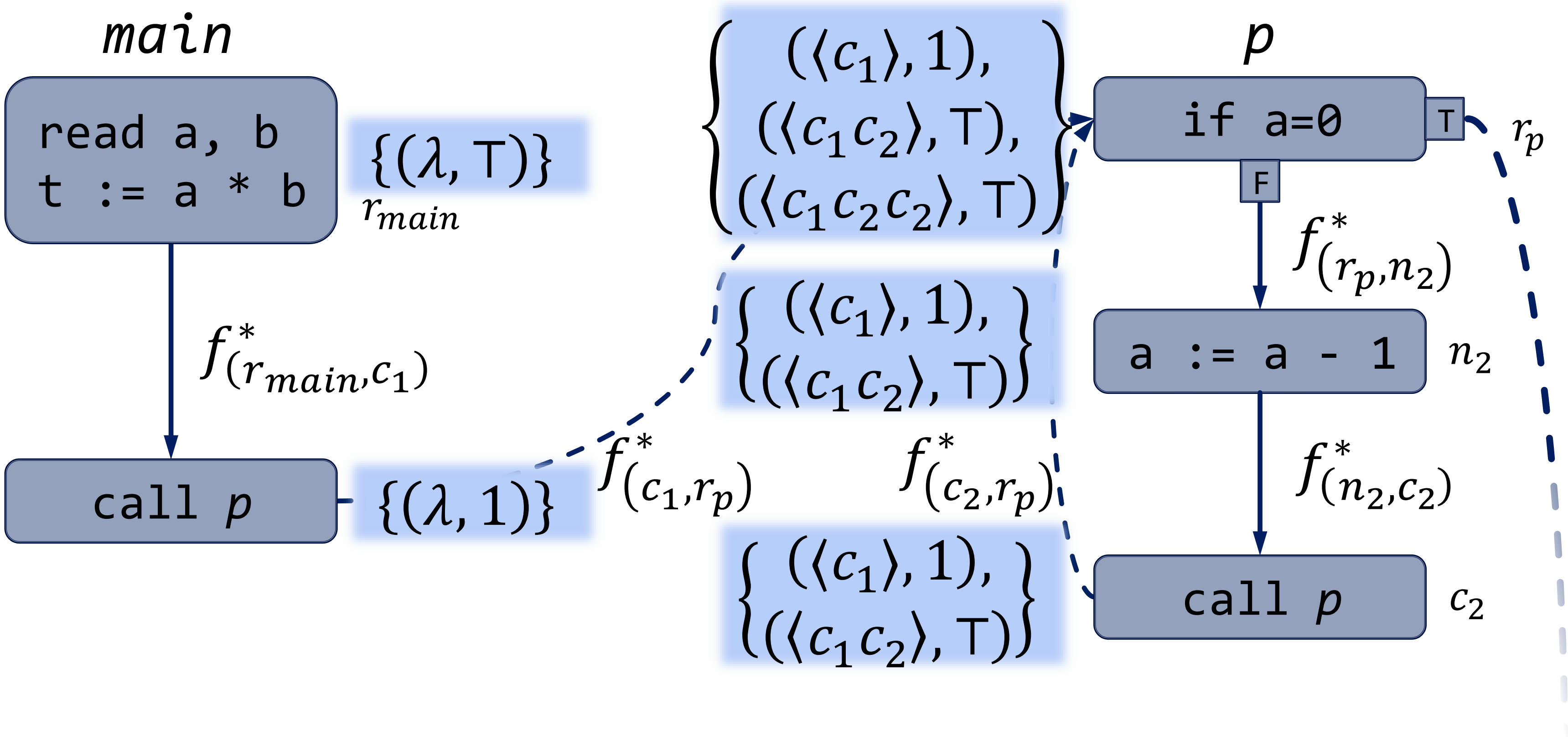
- Completely analogous to MOP_{CS}=MFP_{CS} proof by replacing $\Gamma, IVP, \circ$ by $\Gamma_0, IVP', \circ_0$
- No reasoning about infinite meets or continuity of F_0^* required

■

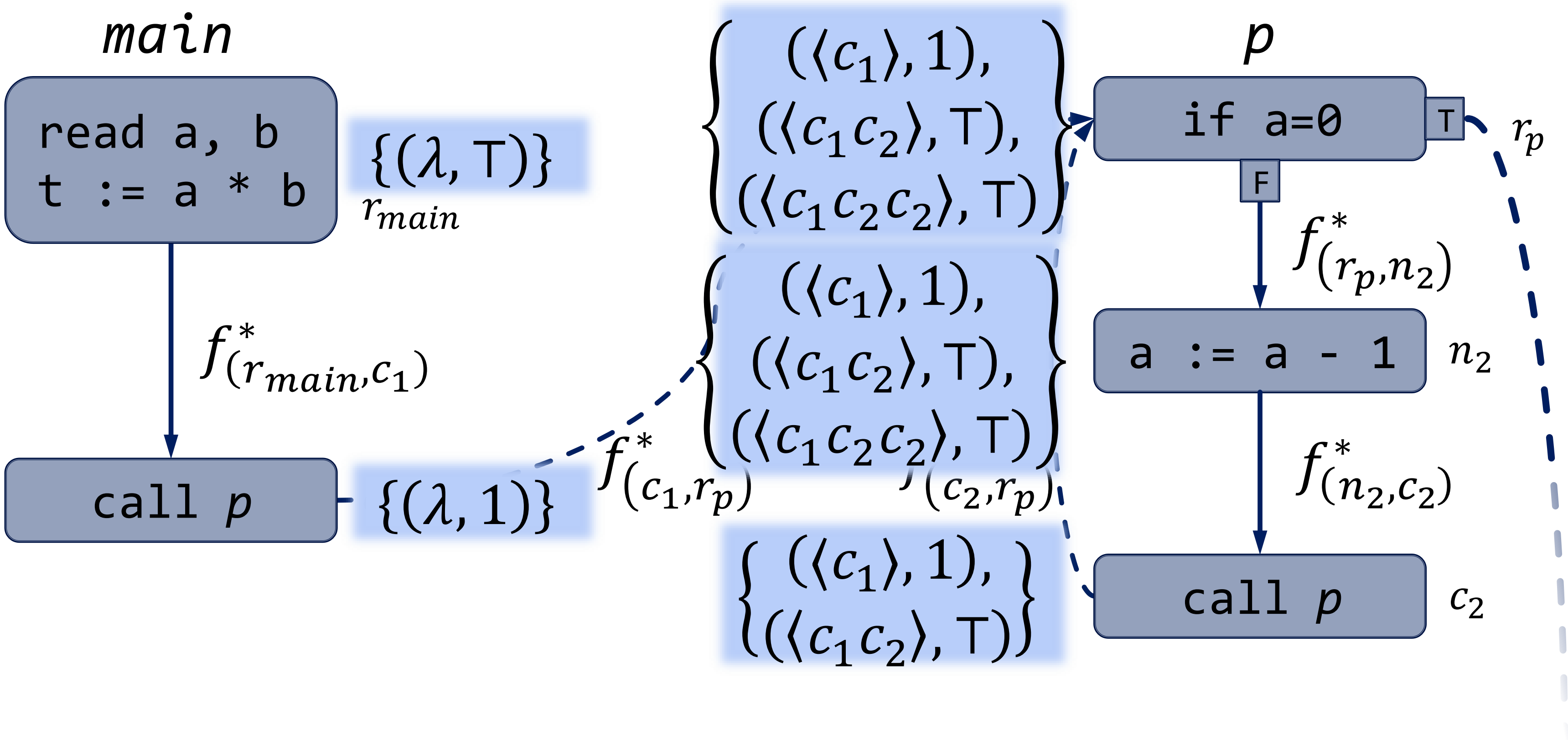
Motivation: CS of limited length



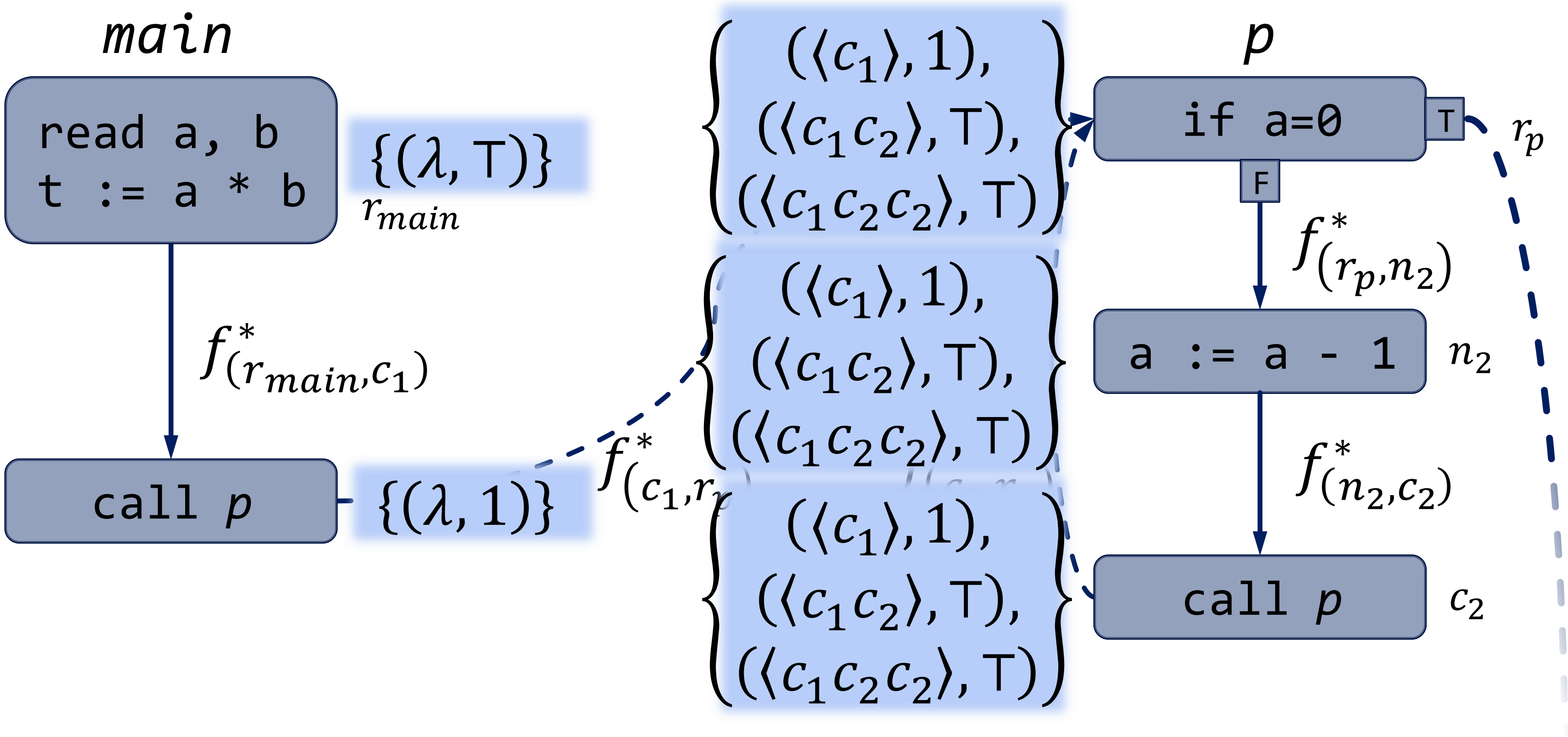
Motivation: CS of limited length



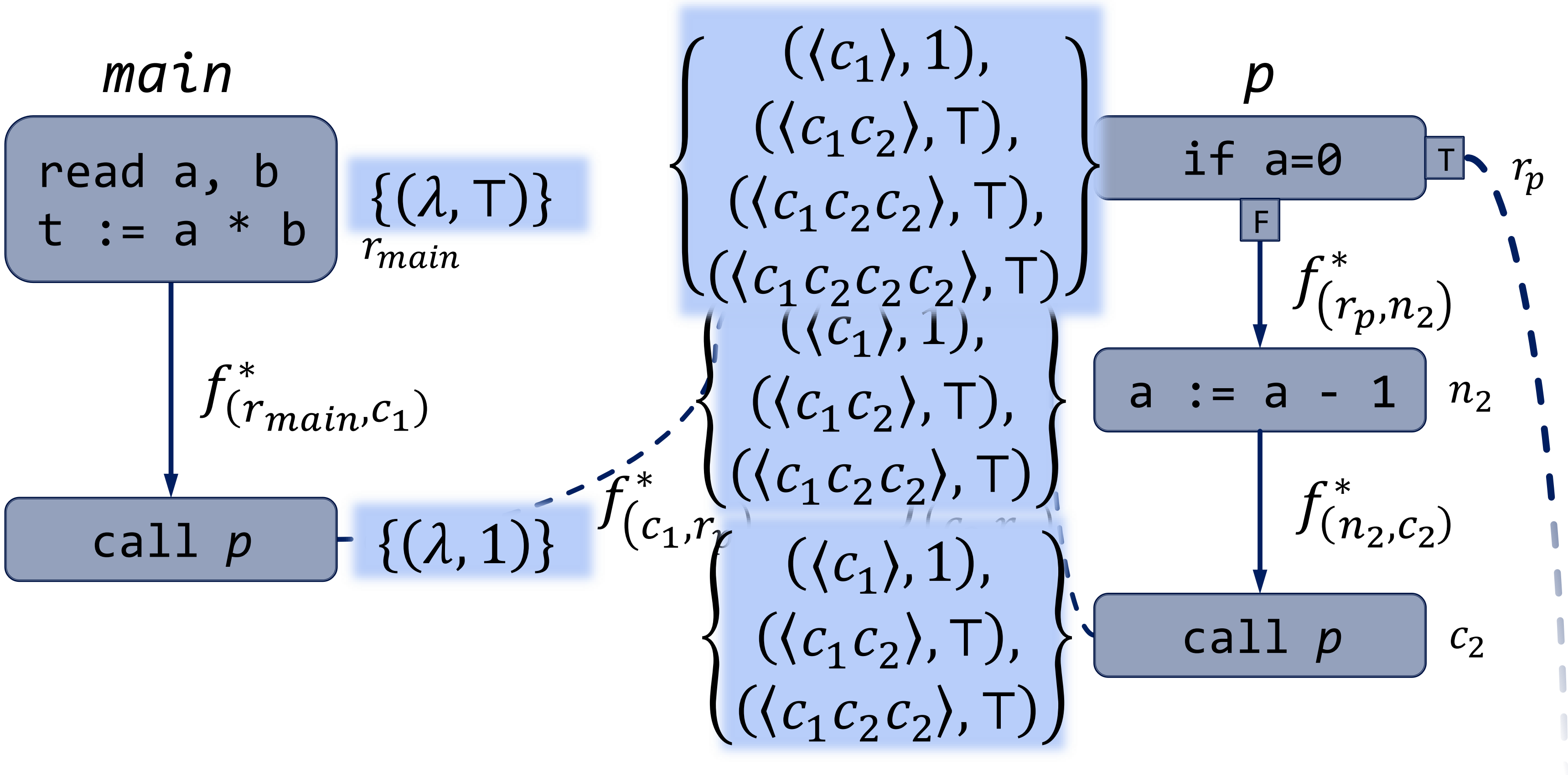
Motivation: CS of limited length



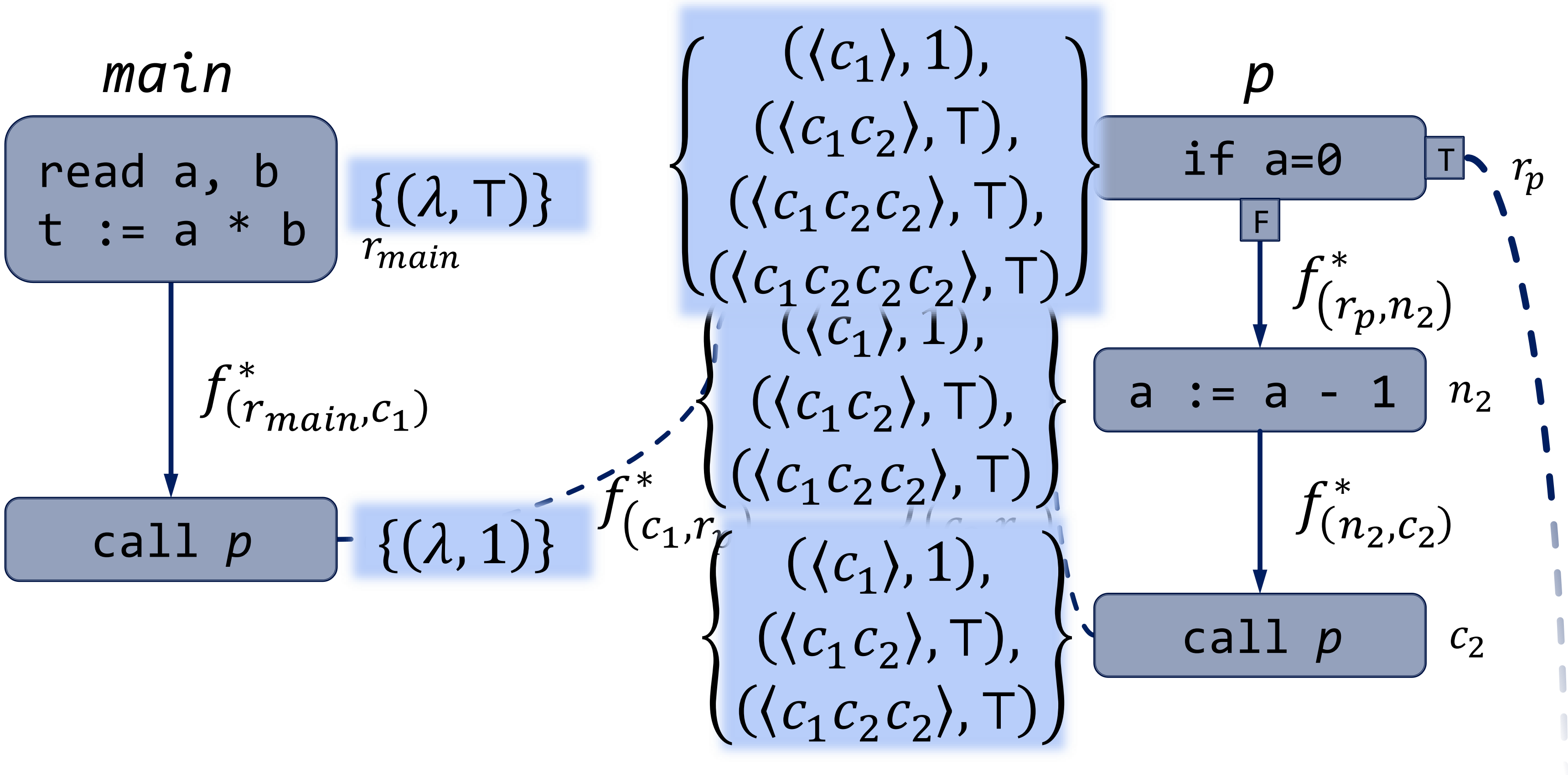
Motivation: CS of limited length



Motivation: CS of limited length

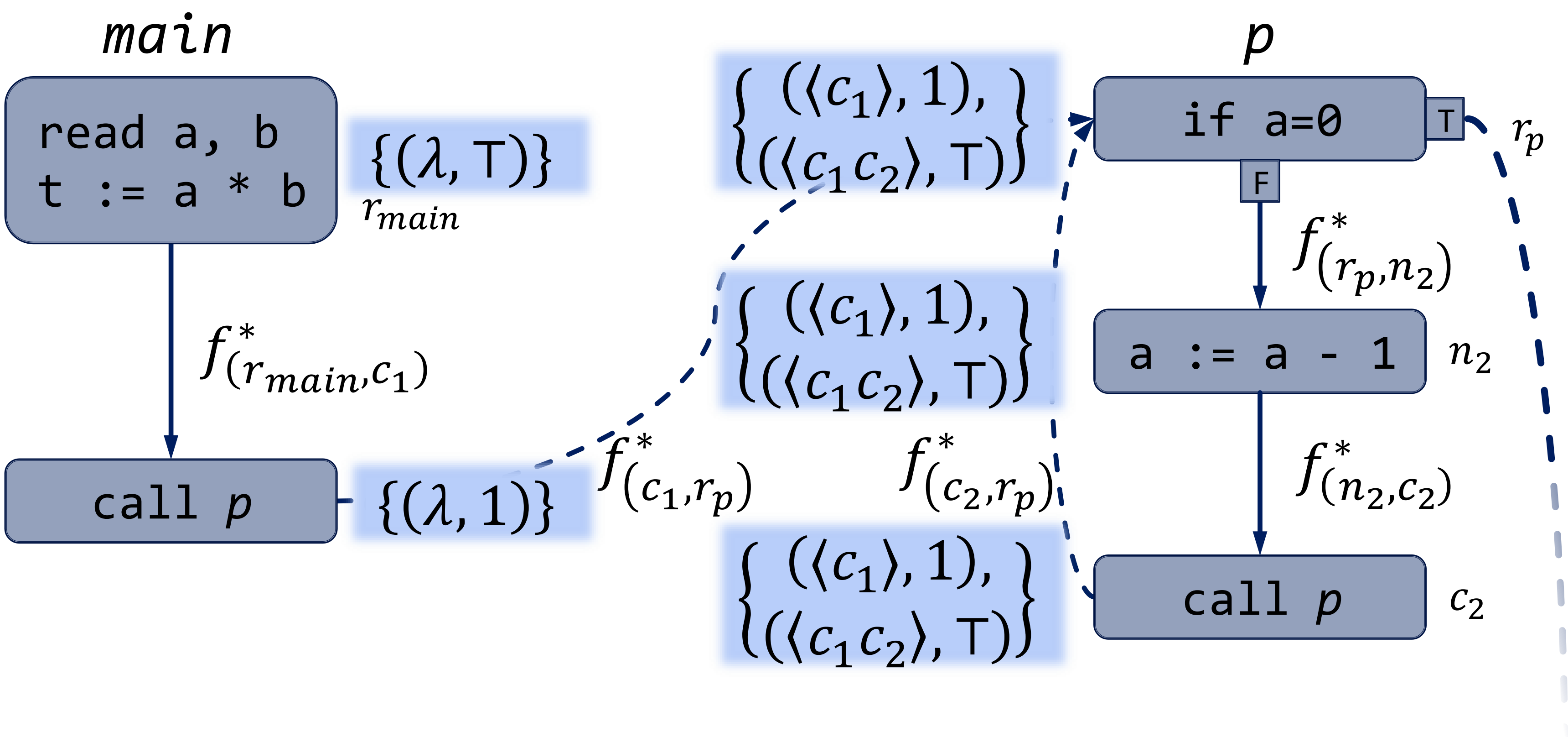


Motivation: CS of limited length



- No information gain on further iteration
- Stop if data tagged by longer CS „irrelevant“

Motivation: CS of limited length



- No information gain on further iteration
- Stop if data tagged by longer CS „irrelevant“

Redundant information in analysis

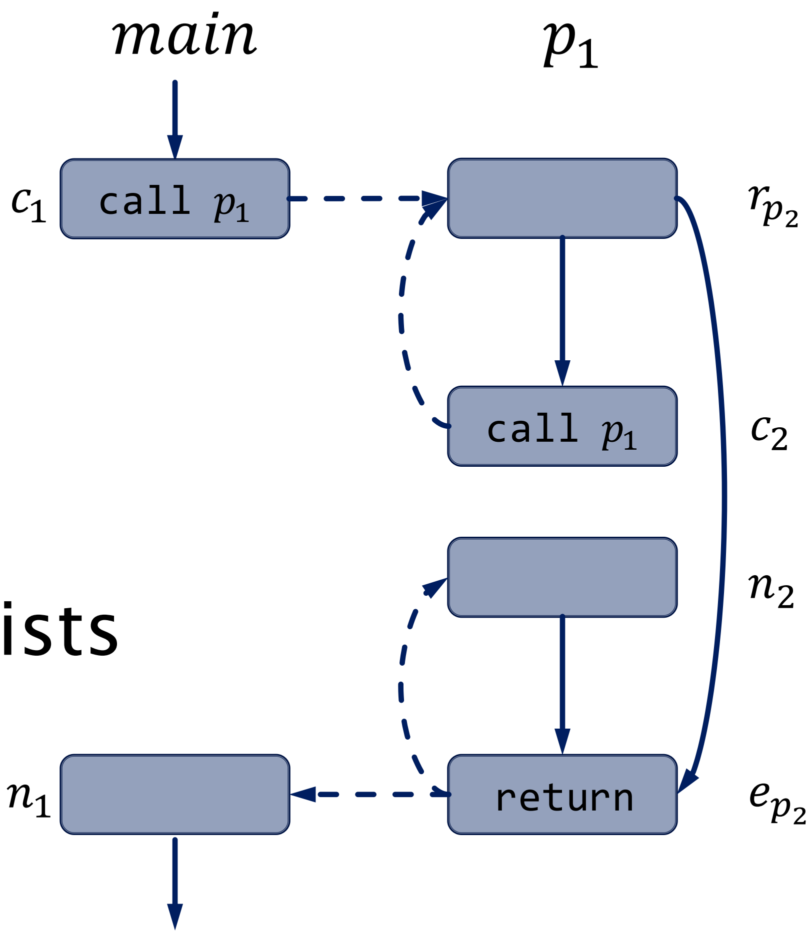
- Why paths of arbitrary length?
  - K := number of call sites in program $\in \mathbb{N}$
    $\Rightarrow$ only possible by recurring on call site(s)
  - At each call, analysis arrives with value $x \in L$ and returns with value $y \in L$.
    L is finite $\Rightarrow |L|^2$ many distinct begin/end value combinations per call
- **Idea:** Recurring on any of the K calls more than $|L|^2$ times is redundant.

Lemma: Path shortening

Let

- (L, F) with L finite
- $M := K * |L|^2$
  - $K :=$ #(call blocks in G^*)
- $\Gamma_0 = \Gamma_M$.

Then $\forall\, n \in N^*$:
If $q \in IVP(r_{main}, n)$ then exists $q' \in IVP'(r_{main}, n)$ with $f_q(0) = f_{q'}(0)$.

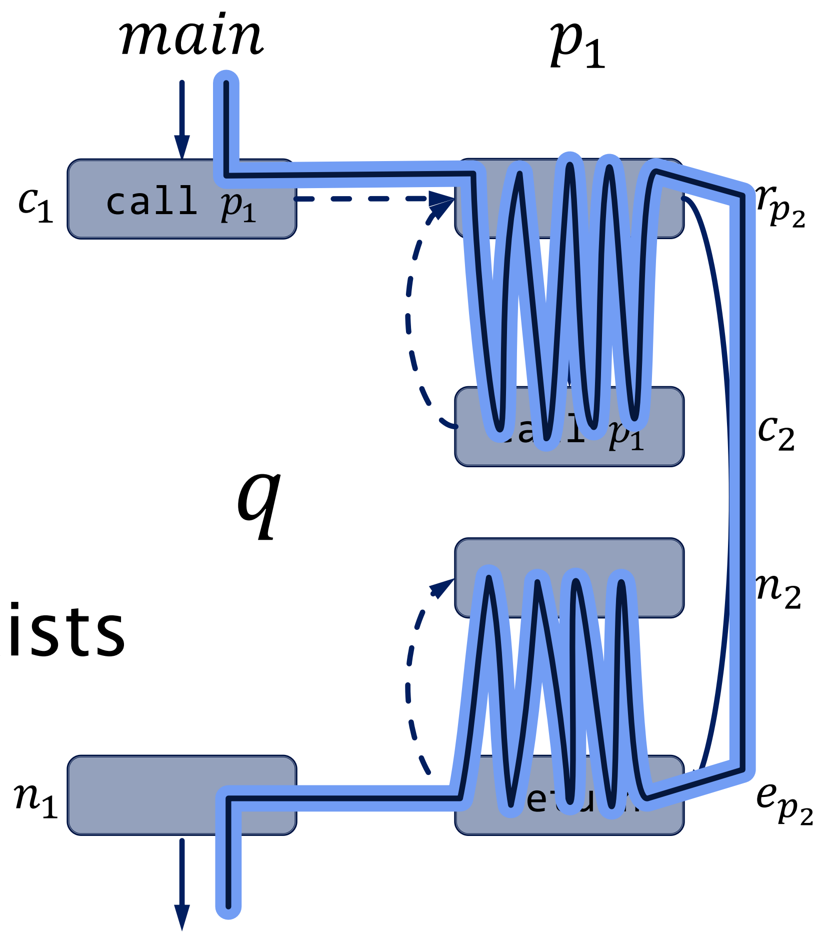


Lemma: Path shortening

Let

- (L, F) with L finite
- $M := K * |L|^2$
  - $K := \#(\text{call blocks in } G^*)$
- $\Gamma_0 = \Gamma_M$.

Then $\forall\, n \in N^*$:
If $q \in IVP(r_{main}, n)$ then exists $q' \in IVP'(r_{main}, n)$ with $f_q(0) = f_{q'}(0)$.

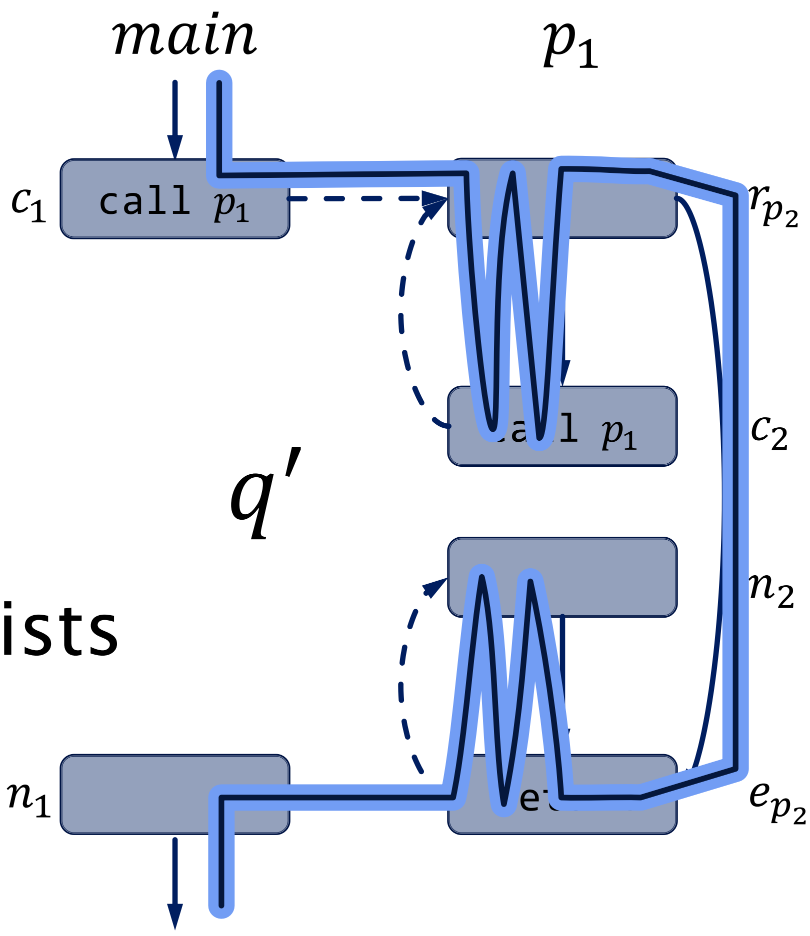


Lemma: Path shortening

Let

- (L, F) with L finite
- $M \coloneqq K * |L|^2$
  - $K \coloneqq \#(\text{call blocks in } G^*)$
- $\Gamma_0 = \Gamma_M$.

Then $\forall\, n \in N^*$:
If $q \in IVP(r_{main}, n)$ then exists $q' \in IVP'(r_{main}, n)$ with $f_q(0) = f_{q'}(0)$.



Lemma: Path shortening (proof)

- **Proof:** By induction on $l(q) \coloneqq$ length of q.
  - $\boldsymbol{l(q) = 0}$: $\lambda \in \Gamma$ and $\lambda \in \Gamma_0$.
  - **IH**: Lemma holds for all paths with $l(q) < k, k \in \mathbb{N}^+$.
  - **IS**: Let $n \in N^*, q \in IVP(r_{main}, n), l(q) = k$.

    Assume $q \notin IVP'(r_{main}, n)$.
    Let q' the shortest prefix with $q' \notin IVP'(r_{main}, n)$.
    Then q' contains $M + 1$ unreturned calls.

    By decomposition Lemma:

$$q' = q_0 || \left(c_1, r_{p_1}\right) || q_1 || \left(c_2, r_{p_2}\right) || \dots || \left(c_{M+1}, r_{p_{M+1}}\right) || q_{M+1}$$

Lemma: Path shortening (proof)

$$q' = q_0 || \left(c_1, r_{p_1}\right) || q_1 || \left(c_2, r_{p_2}\right) || \dots || \left(c_{M+1}, r_{p_{M+1}}\right) || q_{M+1}$$

- $M + 1$ calls, but max. M distinct elements. Let $\left(\hat{c}, r_{\hat{p}}\right)$ a duplicate call. If $\hat{c}$ returns, let $\left(e_{\hat{p}}, n\right)$ the return edge.
- Rewrite q as $q_0' \, || \left(\hat{c}, r_{\hat{p}}\right) || \, q_1' \, || \left(\hat{c}, r_{\hat{p}}\right) || \, q_2' \, || \left(e_{\hat{p}}, n\right) || \, q_3' \, || \left(e_{\hat{p}}, n\right) || \, q_4'$
- Shorter $\hat{q} \in IVP(r_{main}, n)$ with $f_{\hat{q}} = f_q$ by dropping redundant parts
- By **IH**, $\exists q' \in IVP'(r_{main}, n)$ for $\hat{q}$ with $f_{q'} = f_{\hat{q}}$ and thus $f_{q'} = f_q$.

$\blacksquare$

Lemma: Path shortening (proof)

$$q' = q_0 || \left(c_1, r_{p_1}\right) || q_1 || \left(c_2, r_{p_2}\right) || \ldots || \left(c_{M+1}, r_{p_{M+1}}\right) || q_{M+1}$$

- $M + 1$ calls, but max. M distinct elements. Let $\left(\hat{c}, r_{\hat{p}}\right)$ a duplicate call. If $\hat{c}$ returns, let $\left(e_{\hat{p}}, n\right)$ the return edge.
- Rewrite q as $q'_0 || \left(\hat{c}, r_{\hat{p}}\right) || q'_1 || \left(\hat{c}, r_{\hat{p}}\right) || q'_2 || \left(e_{\hat{p}}, n\right) || q'_3 || \left(e_{\hat{p}}, n\right) || q'_4$
- Shorter $\hat{q} \in IVP(r_{main}, n)$ with $f_{\hat{q}} = f_q$ by dropping redundant parts
- By **IH**, $\exists q' \in IVP'(r_{main}, n)$ for $\hat{q}$ with $f_{q'} = f_{\hat{q}}$ and thus $f_{q'} = f_q$.

■

MOP_{CS0}=MOP

- For $M \in \mathbb{N} \geq 0$, define Γ_M as set of all $\gamma \in \Gamma$ with length $\leq M$.

- **Theorem**:
  Let (L, F) a distributive DF framework with L finite, and $\Gamma_0 = \Gamma_M$ with $M = K * |L|^2$. Then

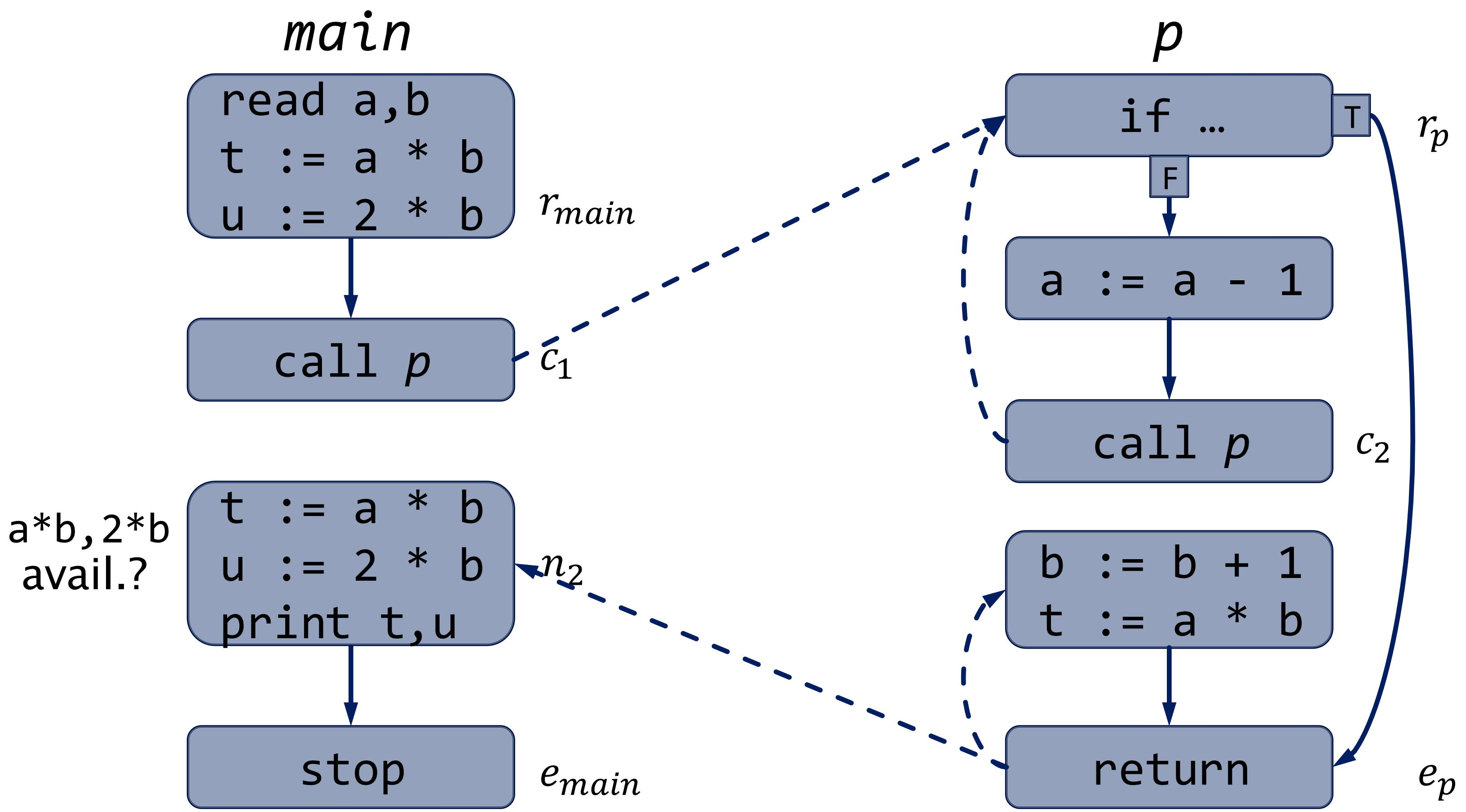
$$\forall n \in N^*: x'_{n_0} = y_n$$

⇒ "If γ is too long and gets discarded, a shorter one in the set represents the same data."

Better bounds for M

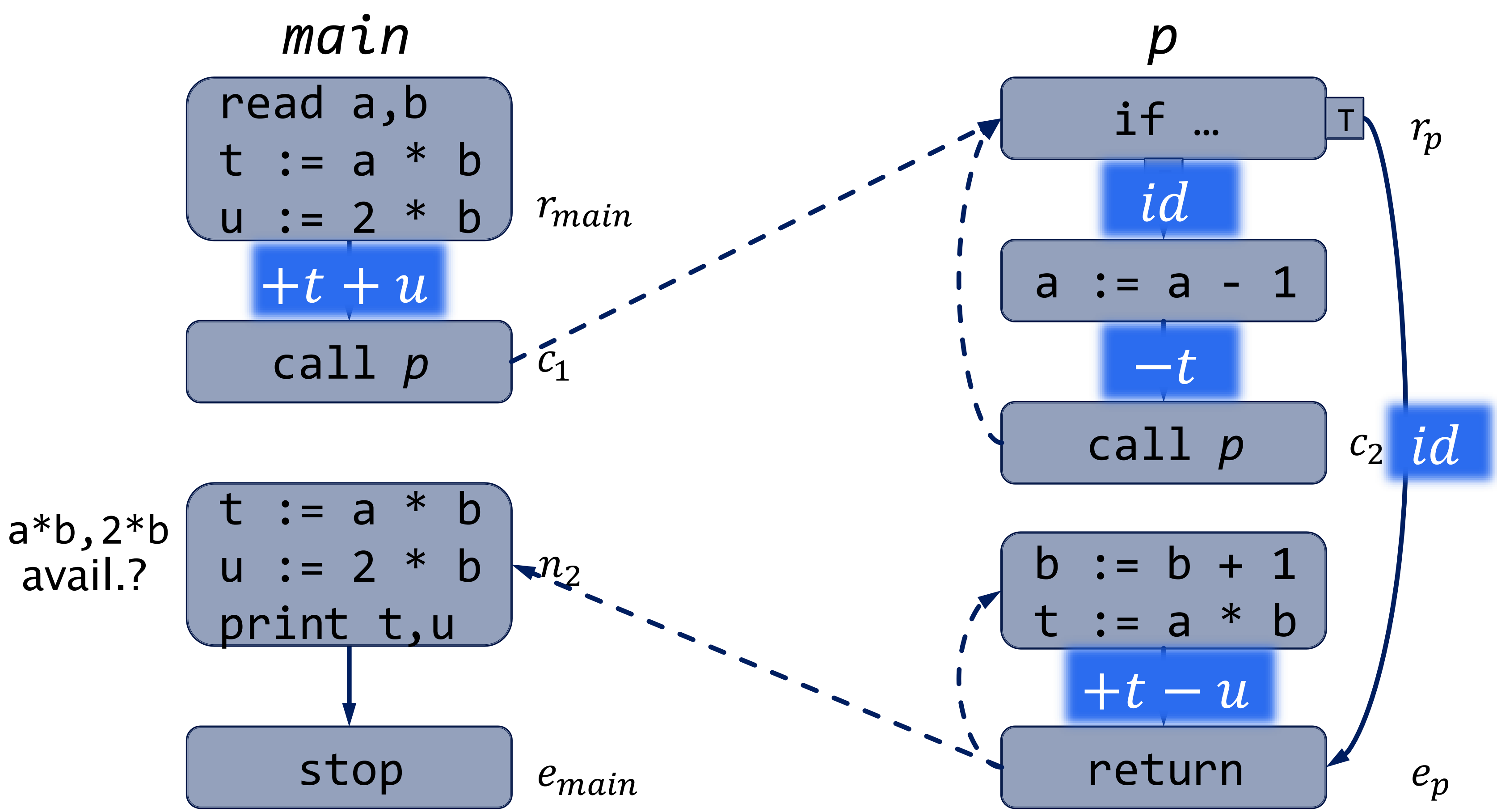
- M impractically large for most problems
- Lower bounds exists for certain problem classes.
  Example: Decomposable frameworks

Decomposable frameworks



$$L = \{\{\,\}, \{t\}, \{u\}, \{t,u\}, \bot\} \qquad M = 50$$

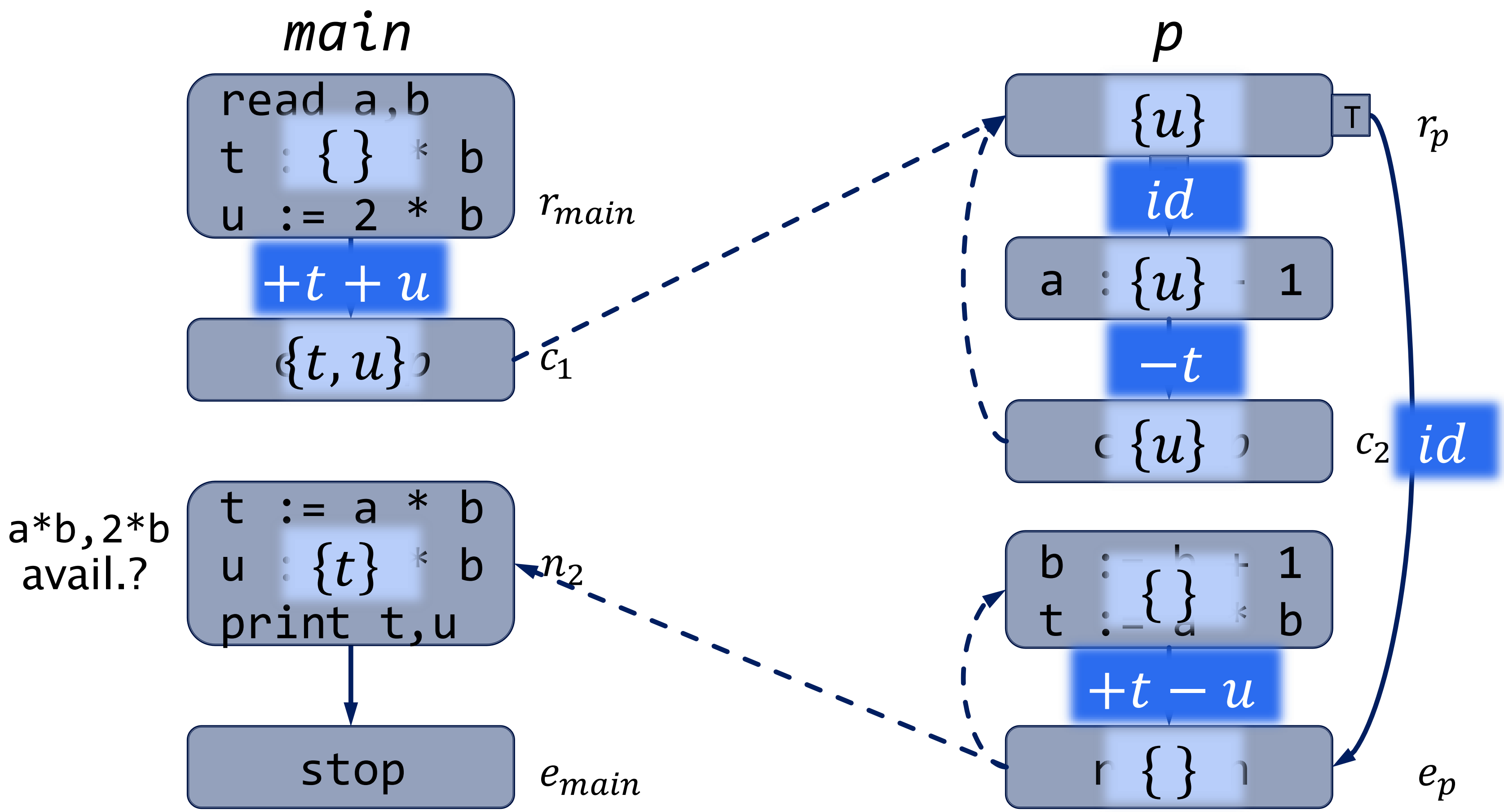
Decomposable frameworks



$$L = \{\{\,\}, \{t\}, \{u\}, \{t, u\}, \bot\}$$

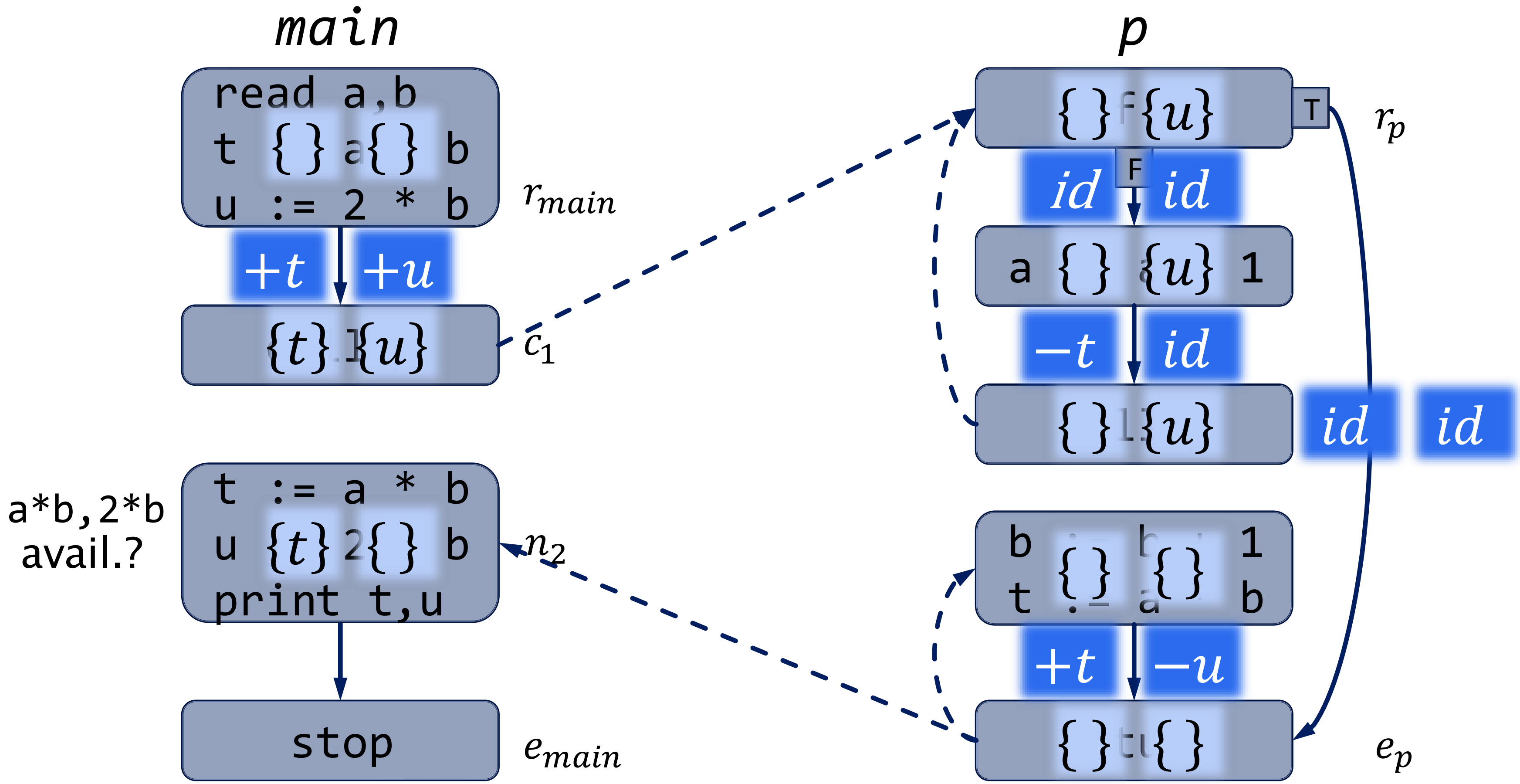
$$M = 50$$

Decomposable frameworks



$L = \{\{\,\},\{t\},\{u\},\{t,u\},\bot\}$ $M = 50$

Decomposable frameworks



$$L_t = \{\{\,\}, \{t\}, \bot\},\ L_u = \{\{\,\}, \{u\}, \bot\},\ \Rightarrow L = L_t \times L_u \qquad M = 18$$

Bound for decomposable frameworks

- **Theorem:** Let (L, F) decomposable into k frameworks $(L_i, F_i)_{i=1}^{k}$.
  Setting the maximum callstring length to

  $$M = K * \max_{i \in \{1..k\}} (|L_i|)^2$$

  yields $y'_{n_0} = y_n \; \forall n \in N^*$.

1−related frameworks

- **Definition:** A decomposable DF framework (L,F) is **1−related** if each F_i only consists of constant and identity functions.
- **Theorem:** In this case, using $\Gamma_0 = \Gamma_{3K}$ yields $y'_{n\,0} = y_n \; \forall n \in N^*$.
- **Example:** Available expressions is 1−related. Decomp. into subproblems (L_e, F_e) for each expression e with $L_e = \{\top, 1, \bot\}$, $F_e = \{id, f_\top, f_1, f_\bot\}$

Chapter summary

- If L finite, MFP_{CS} iteratively computable on Γ_0
  - Precision/safeness depending on choice for Γ_0
- Enforce termination by limiting max. CS length
  - Precision preserving bounds exist, but only of theoretical value
  - Better bounds exist for special problem classes

Structure

1. Definition of a new DF problem (L^*, F^*)
2. Proof: Solution to $(L^*, F^*) \equiv$ MOP solution
3. Feasibility and precise variants
4. **Approximative solutions**

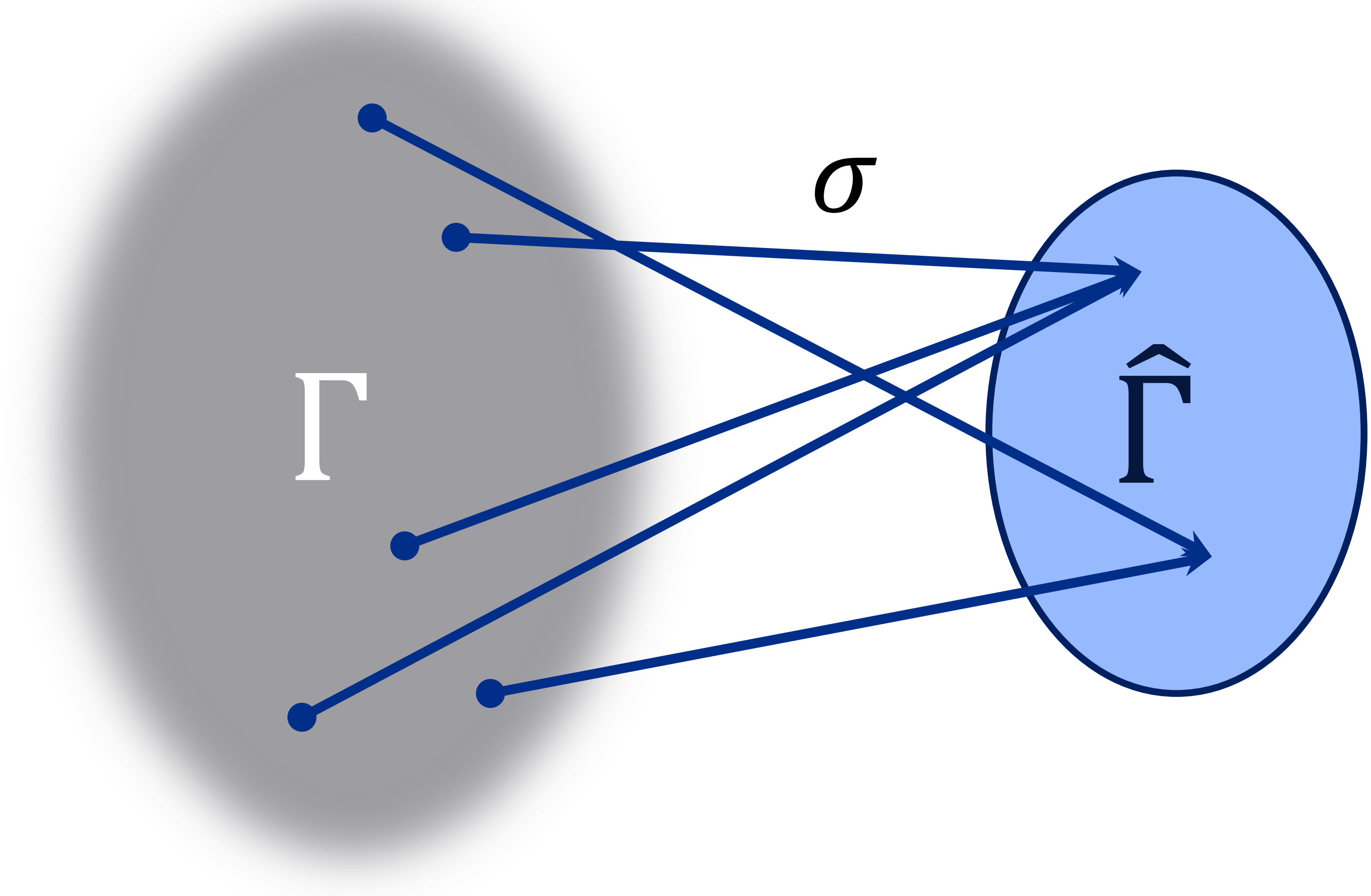
Motivation

- Prefer a safe approximative solution $\hat{x}_n \leq y_n$ over MFP_{CS} if
  - MFP_{CS} not (iteratively) computable
  - computation not feasible MFP_{CS} by time/space constraints

  and computation of $\hat{x}_n \leq y_n$ feasible.

⇒Accept precision loss for less complexity.

Reducing complexity of CS approach



Embedding Γ into $\hat{\Gamma}$

- Let $*$ an operation in $\hat{\Gamma}$ with left identity $w \in \hat{\Gamma}$.
- Encoding function $\sigma: \Gamma \to \hat{\Gamma}$
  - Define $\sigma(n)$ for each **call node** n in G^*.
  - For $\gamma = (c_1 c_2 \dots c_n) \in \Gamma$:

$$\sigma(\gamma) = \sigma(c_1) * \cdots * \sigma(c_n)$$

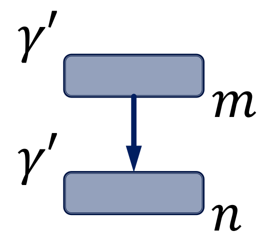
and

$$\sigma(\gamma) = w.$$

Transfer relation $R_{(m,n)}$

- For each $(m, n) \in E^*$, define transfer relation $R_{(m,n)} \in \hat{\Gamma} \times \hat{\Gamma}$:

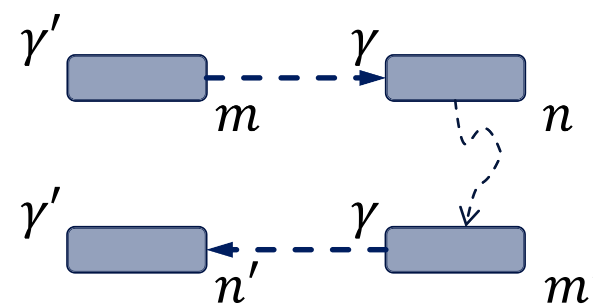
  - If (m, n) intraproc.:
    $\forall a \in \hat{\Gamma}$: $a \; R_{(m,n)} \; a$



$\sigma(\gamma') = a$

  - If (m, n) interproc.:
    $\sigma(\gamma') \; R_{(m,n)} \; \sigma(\gamma)$

    $\Rightarrow R_{call} = R_{return}^{-1}$

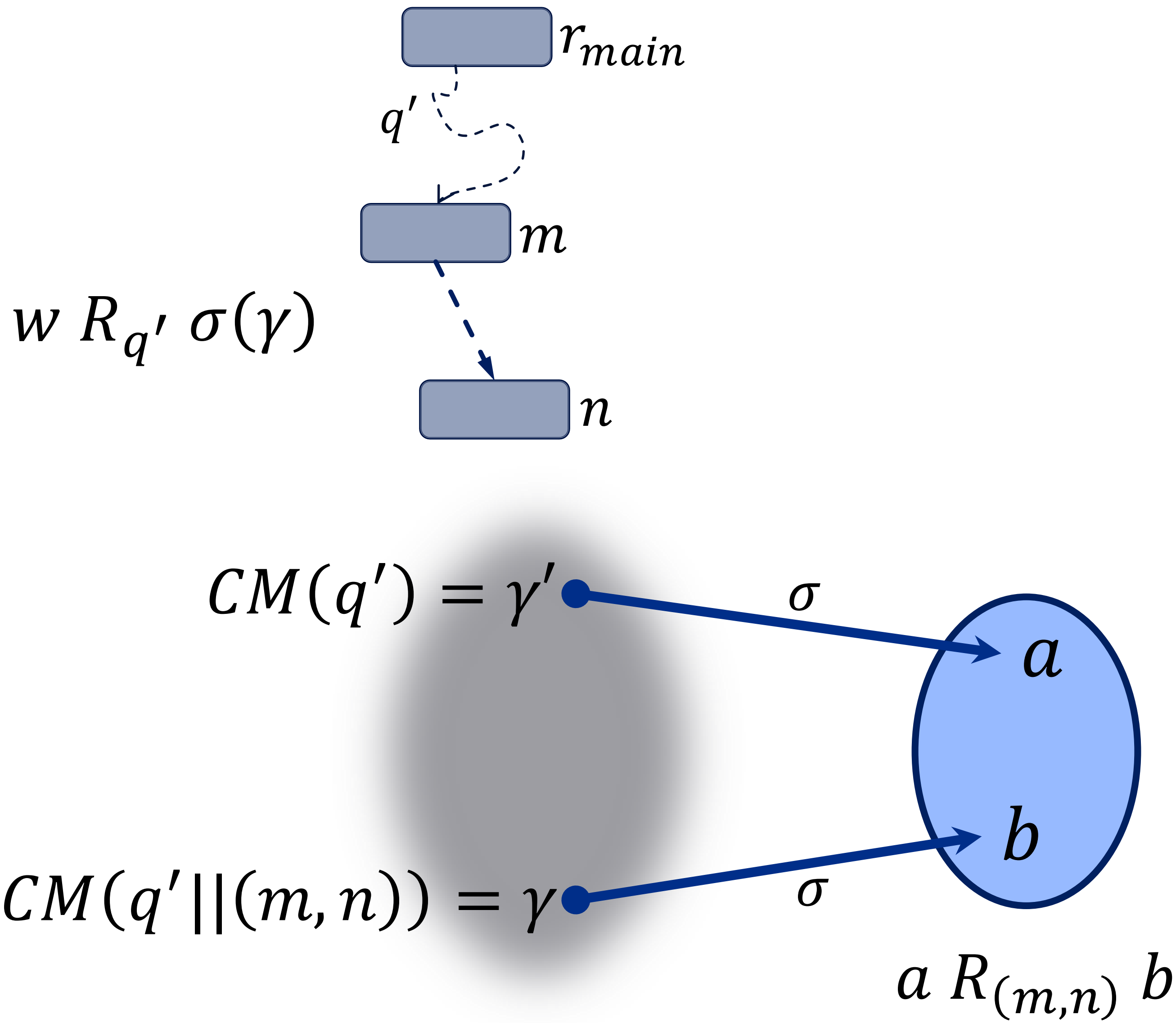


Acceptable paths

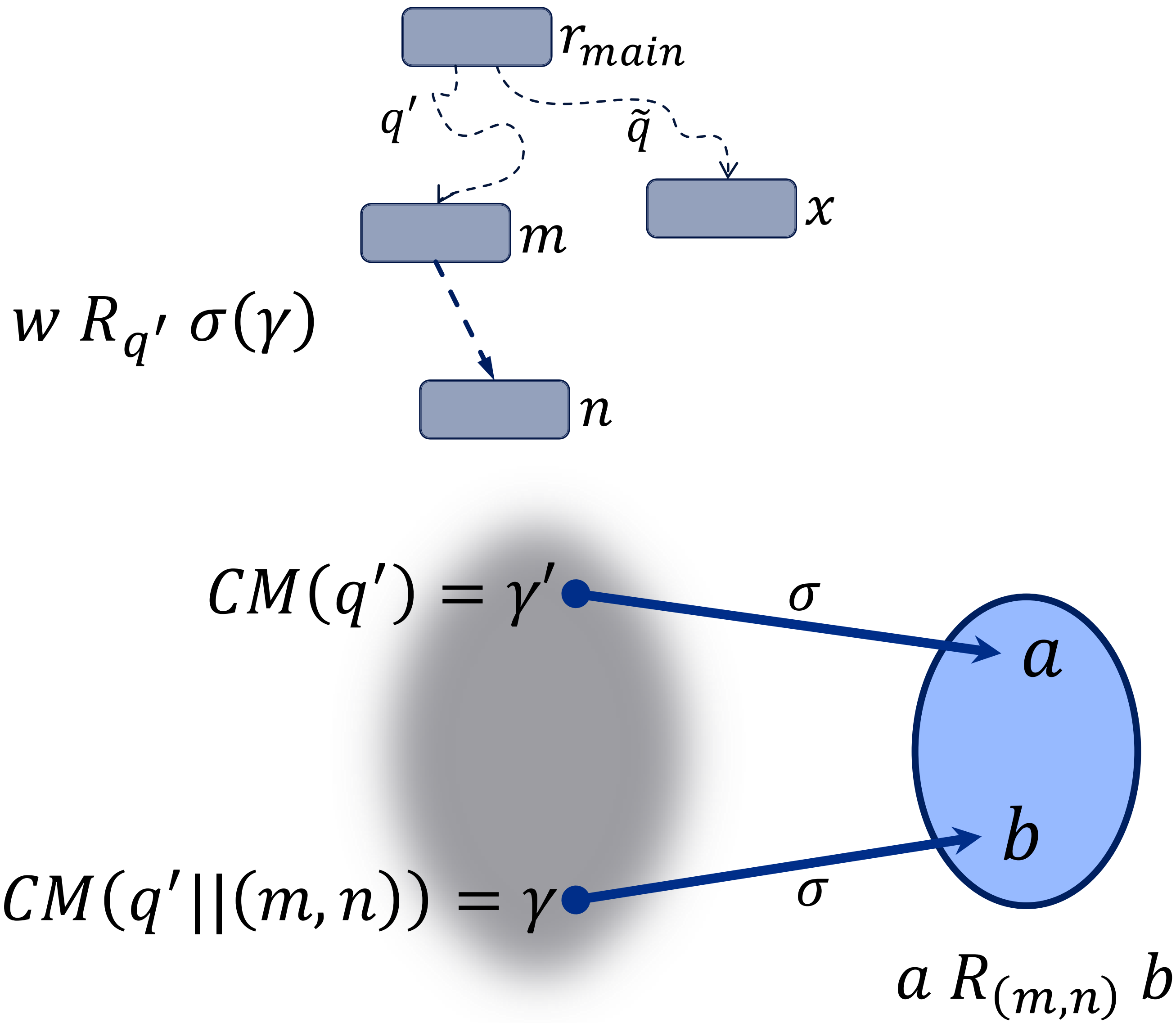
Let $q = (n_1, n_2, \dots, n_k) \in path_{G^*}$.

- Define $R_q := R_{(n_1,n_2)} \circ R_{(n_2,n_3)} \circ \cdots \circ R_{(n_{k-1},n_k)}$
- q **acceptable** $:\Leftrightarrow R_q \neq \emptyset$
- Intraprocedurally acceptable paths:
  $IAP(r_{main}, n) := \{q \in path_{G^*}(r_{main}, n) | R_q\{w\} \neq \emptyset\}$

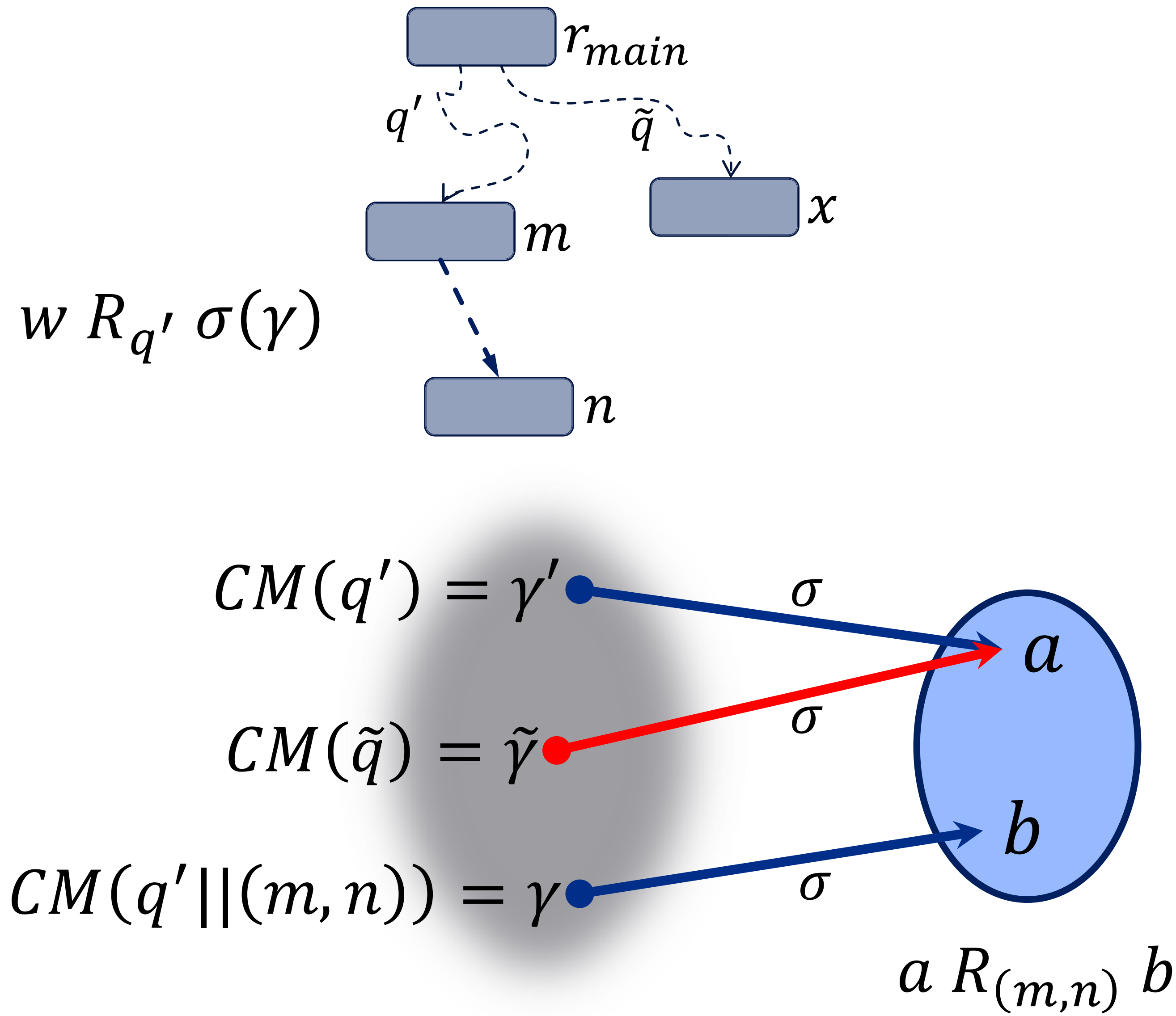
Invalid paths in IAP



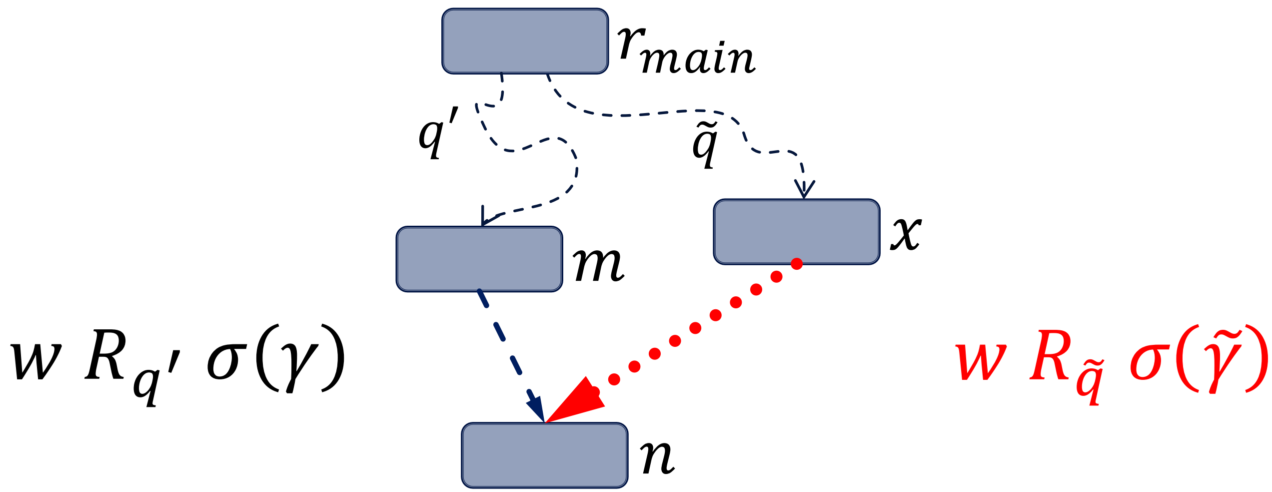
Invalid paths in IAP

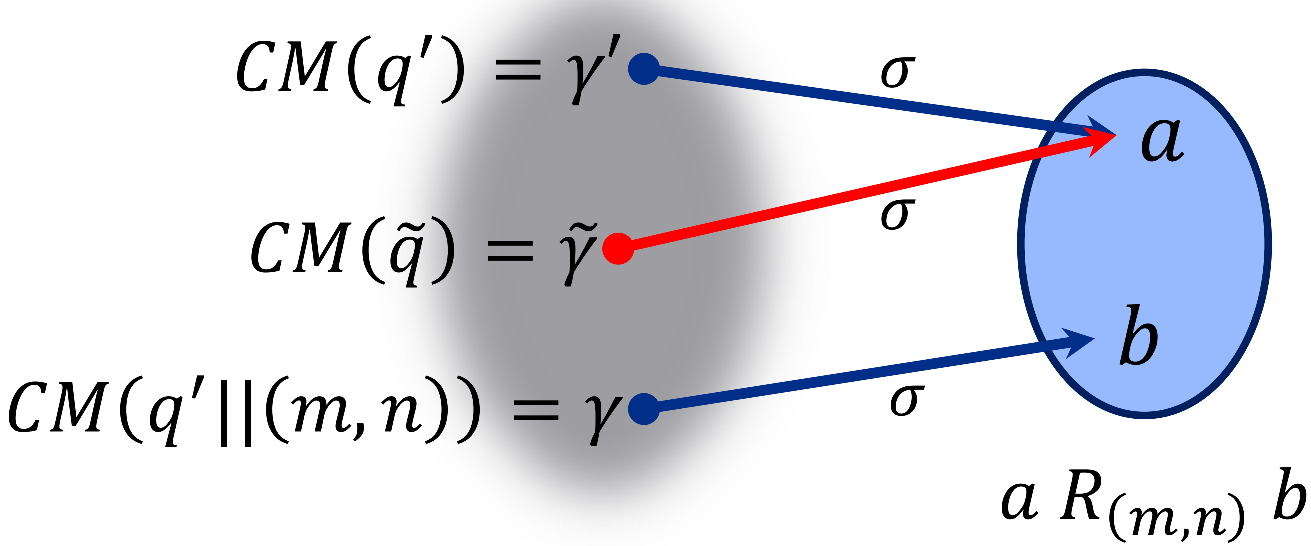


Invalid paths in IAP



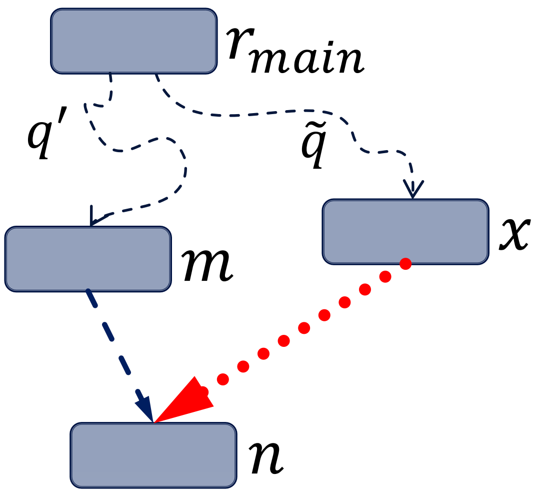
Invalid paths in *IAP*





Invalid paths in IAP

$w\ R_{q'}\ \sigma(\gamma)$



$w\ R_{\tilde{q}}\ \sigma(\tilde{\gamma})$

$CM(q') = \gamma'$

$$\Rightarrow IAP(r_{main}, n) \supseteq IVP(r_{main}, n)$$

$CM(q' \,||\, (m,n)) = \gamma$

$a\ R_{(m,n)}\ b$

Dataflow analysis using $\hat{\Gamma}$

Now define framework $(\hat{L}, \hat{F})$

- $\hat{L} := \hat{\Gamma} \to L$
- For edge (m, n), data in ξ_m is propagated to ξ_n from all $a \in \hat{\Gamma}$ to $b \in \hat{\Gamma}$ for which $a\, R_{(m,n)}\, b$:

$$f^*_{(m,n)}(\xi_m)(b) := \bigwedge\left\{f_{(m,n)}(\xi_m(a)) \,|\, a\, R_{(m,n)}\, b\right\}$$

- As before, define DF equations, the MFP solution $\{\hat{x}^*_n\}_{n \in N^*}$ (which exists as $\hat{\Gamma}$ finite), and the MOP solution $\hat{y}_n$ using IAP.
  $\Rightarrow$ Obviously $\hat{y}_n \leq y_n$, so a safe approximation.

Dataflow analysis using $\hat{\Gamma}$ (contd.)

- Without proof:
  - L distributive: $\hat{x}_n = \hat{y}_n \leq y_n$
  - L only monotone: $\hat{x}_n \leq \hat{y}_n \leq y_n$

- Examples:
  - $\hat{\Gamma} = \{w\}$
    $\sigma: \gamma \mapsto w$
    $*: \{w, w\} \mapsto w$
  - $\hat{\Gamma} = \mathbb{Z}/k\mathbb{Z},\ k \in \mathbb{N}$
    $\forall$ call $c_i \in N^*: \sigma(c_i) = i$
    $a * b \ := (a + b)\ mod\ k$

$\Rightarrow$ intraproc. analysis

$\Rightarrow$ only k call strings

Summary

Summary: CS Approach structure

- CS approach makes interprocedural flow explicit
  - Avoids functional composition of propagation functions
  - Keeps track of origin of data by tagging with CS
  - Complexity of approach “controlled” by Γ

Summary: Results of CS approach (1/2)

Assume (L, F) distributive.

- L and Γ finite:
  MFP_{CS} iteratively computable and yields MOP solution.
- Only L finite:
  Iterative calculation of MFP_{CS} will converge on MOP if Γ replaced by Γ_M, else likely to diverge
  - Depending on problem, better bounds exist.

Summary: Results of CS approach (2/2)

All other cases combinations of:

- (L, F) not distributive
- L infinite
- F not bounded

- Convergence then only guaranteed if Γ embedded into finite subset $\hat{\Gamma}$
  - Also likely to be the only feasible approach to the other situations

- Solution then is safe approximation of MOP

Thanks!