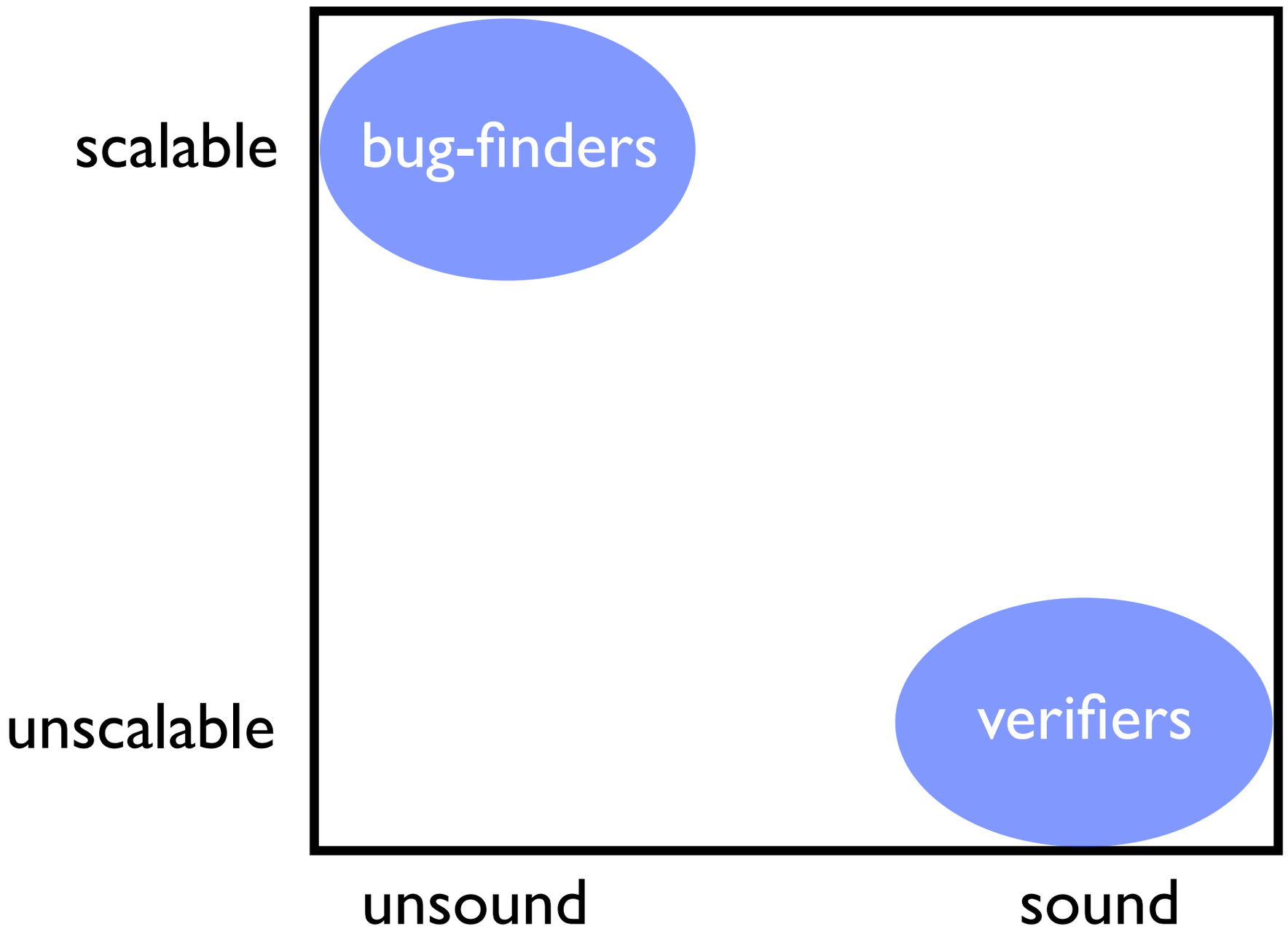
Design and Implementation of Sparse Global Analyses for C-like Languages

Hakjoo Oh, Kihong Heo, Wonchan Lee,
Woosuk Lee, and Kwangkeun Yi

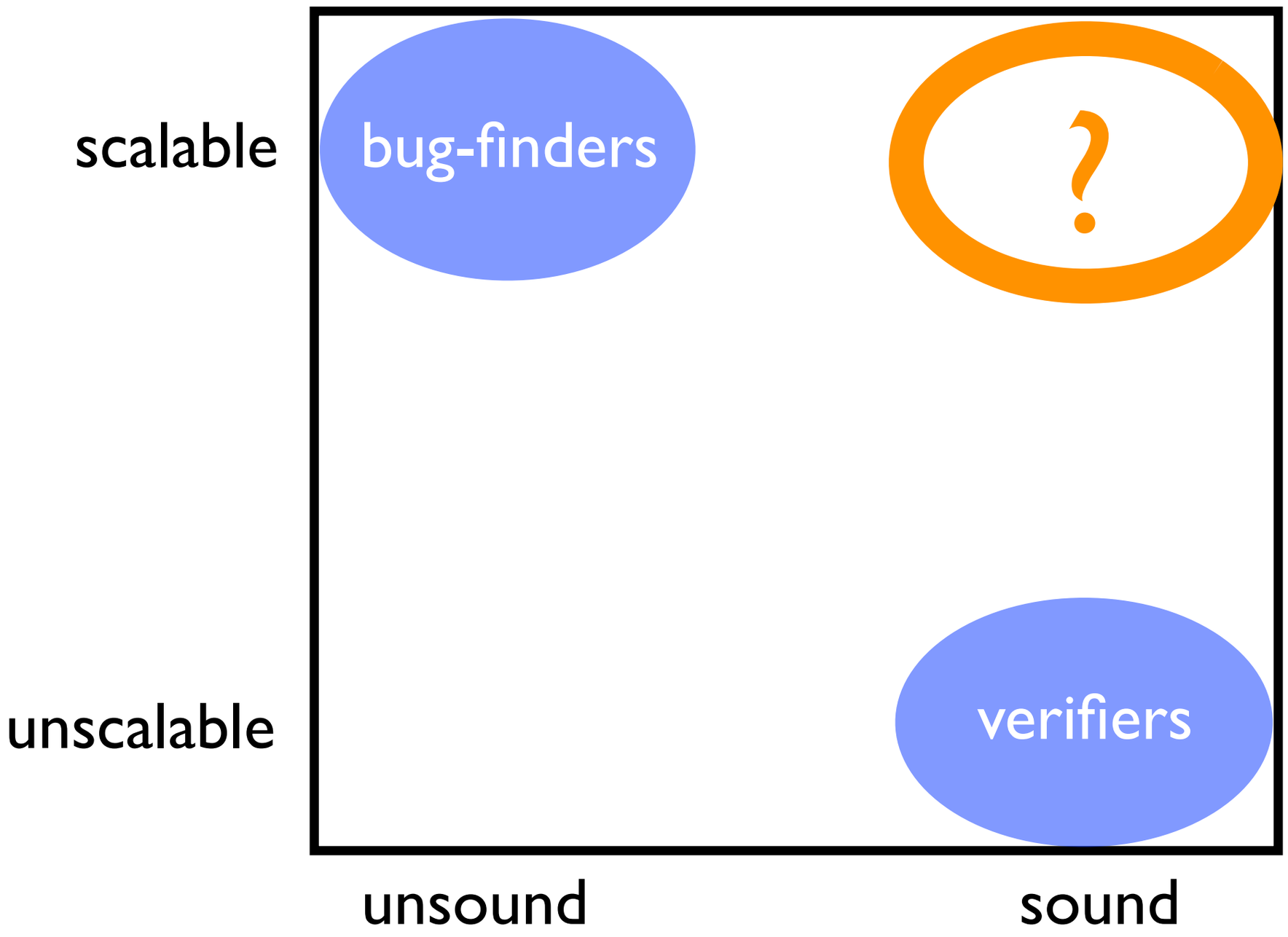
Programming Research Laboratory
Seoul National University

PLDI 2012 @ Beijing, China

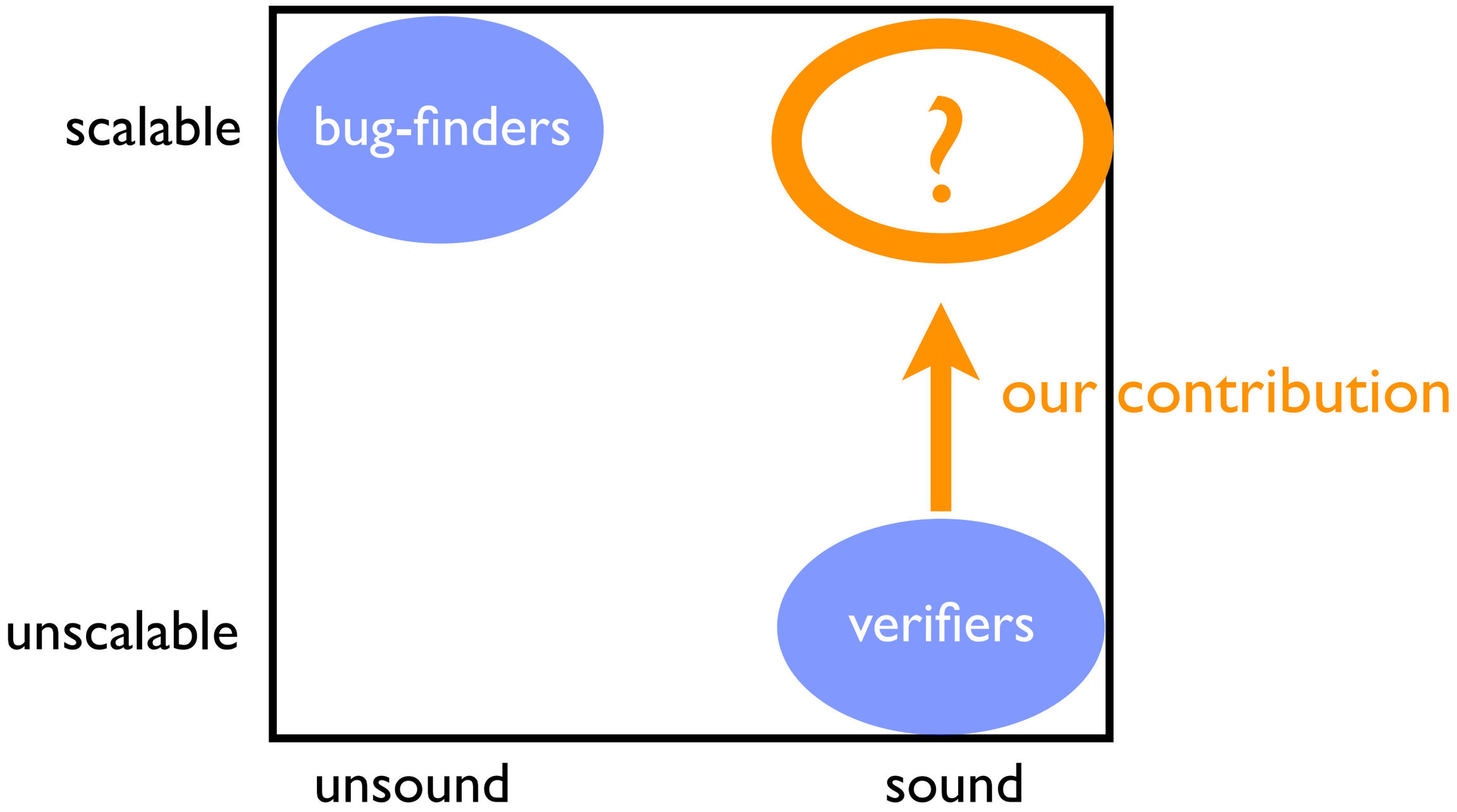
Dichotomy in Static Analysis



Dichotomy in Static Analysis



Dichotomy in Static Analysis



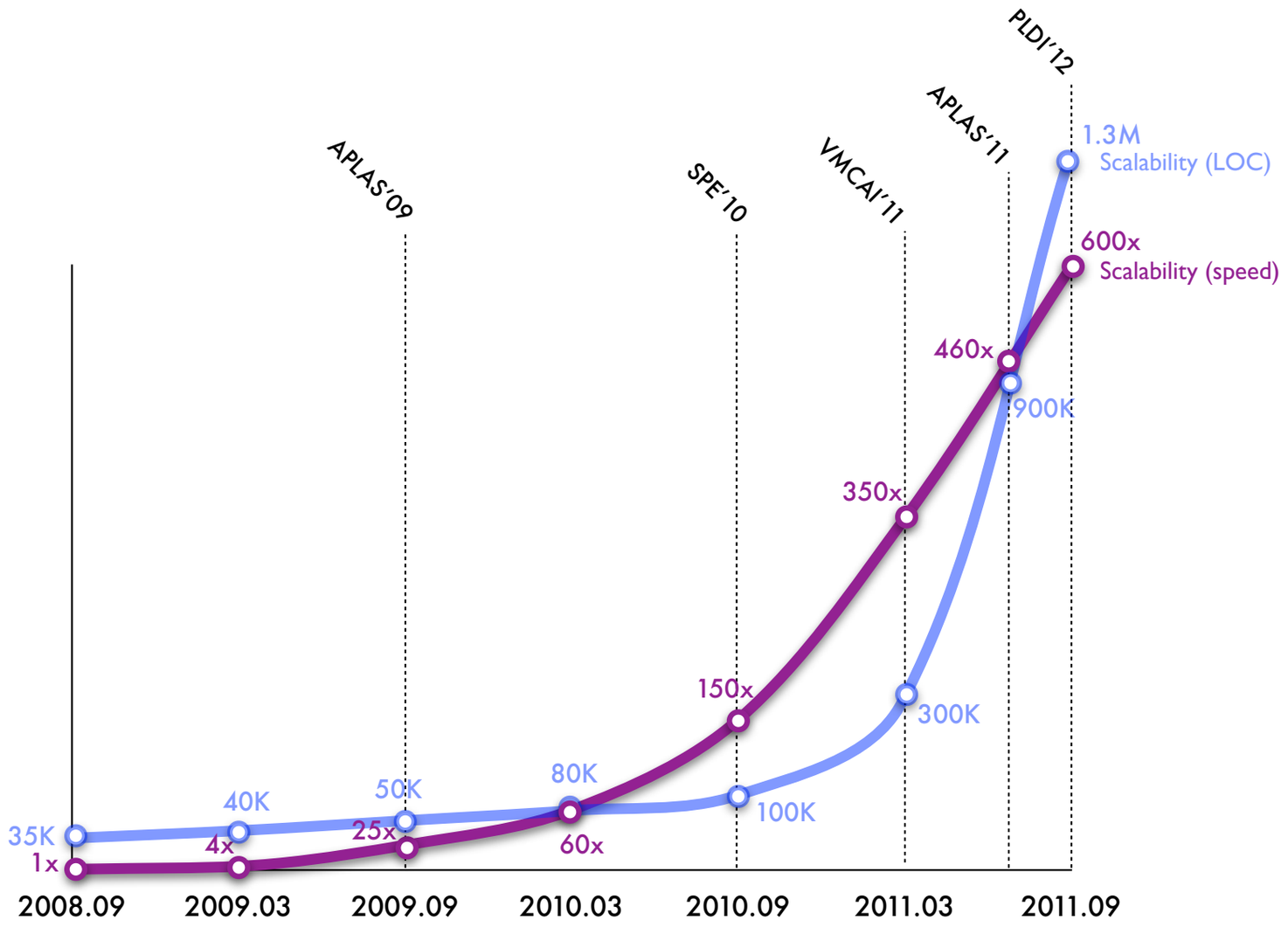
Our Story

- In 2007, we commercialized Sparrow The Early Bird
  - memory-bug-finding tool for full C, non domain-specific
  - designed in abstract interpretation framework
  - sound in design, unsound yet scalable in reality
- Realistic workbench available
  - “let’s try to scale-up its sound & global analysis version”

Scalability Improvement

Sparrow
The Early Bird

sound & global analysis version



- < 1.4M in 10hr with intervals
- < 0.14M in 20hrs with octagons

Precision-Preserving Sparse Analysis Framework

baseline analysis

$$\hat{F} : \hat{D} \to \hat{D}$$

sparsify $\Longrightarrow$

"sparse" version

$$\hat{F}_s : \hat{D} \to \hat{D}$$

$$fix\,\hat{F} \quad \overset{\text{still}}{=} \quad fix\,\hat{F}_s$$

General for AI-based analyzers for C-like languages

Sparse Analysis Framework

- "Right Part at Right Moment"
- "Full Exploitation"
- enabled by Abstract Interpretation theory

Program

$\langle \mathbb{C}, \hookrightarrow \rangle$

- $\mathbb{C}$: set of program points
- $\hookrightarrow \subseteq \mathbb{C} \times \mathbb{C}$: control flow relation

$c' \hookrightarrow c$ (c is the next program point to c')

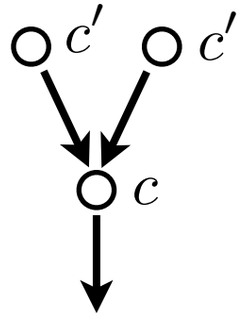
Baseline Analysis

- One abstract state $\in \hat{\mathbb{S}}$ that subsumes all reachable states at each program point

$$
\begin{array}{rcl}
[\![\hat{P}]\!] \in \mathbb{C} \to \hat{\mathbb{S}} & = & \mathit{fix}\,\hat{F} \\
\mathbb{S} & = & \mathbb{L} \to \mathbb{V}
\end{array}
$$

- Abstract semantic function

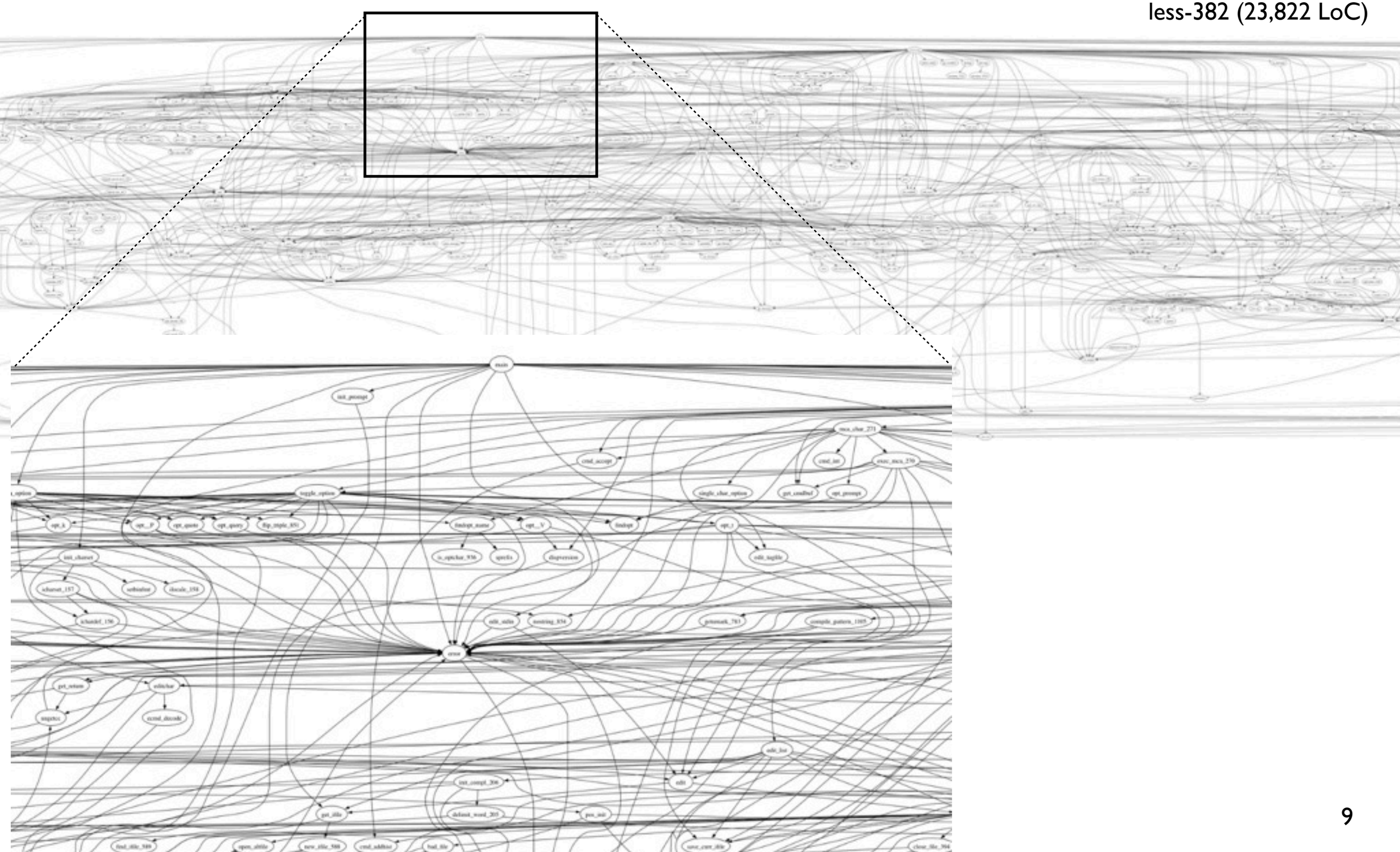
$$
\begin{array}{rcl}
\hat{F} & \in & (\mathbb{C} \to \hat{\mathbb{S}}) \to (\mathbb{C} \to \hat{\mathbb{S}}) \\
\hat{F}(\hat{X}) & = & \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \hookrightarrow c} \hat{X}(c'))
\end{array}
$$



$\hat{f}_c \in \hat{\mathbb{S}} \to \hat{\mathbb{S}}$: abstract semantics at point c

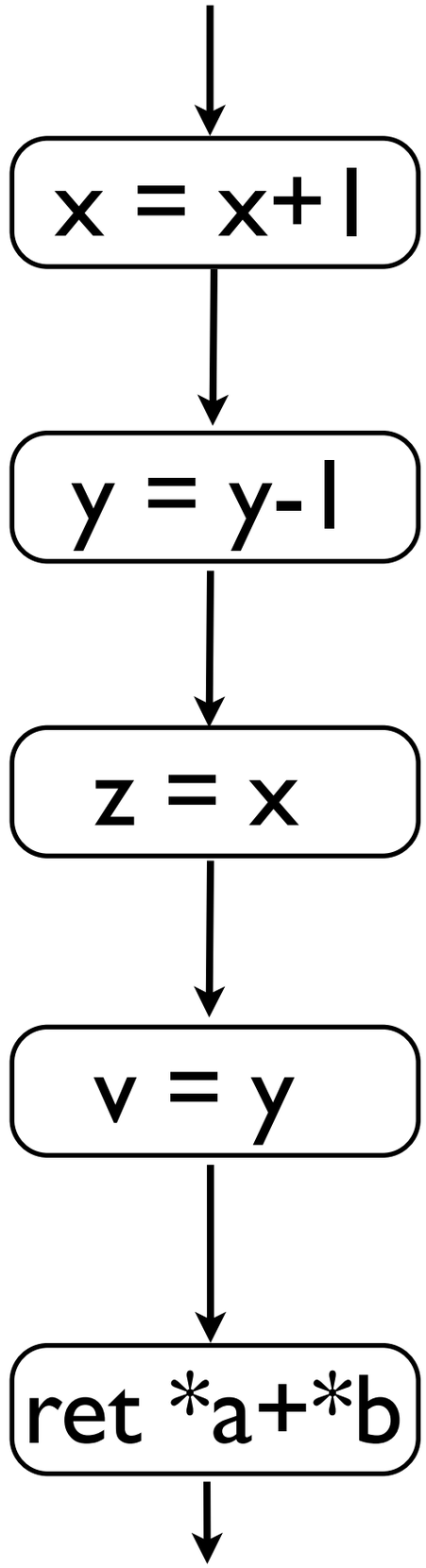
Direct Implementation (convention) Too Weak To Scale

less-382 (23,822 LoC)



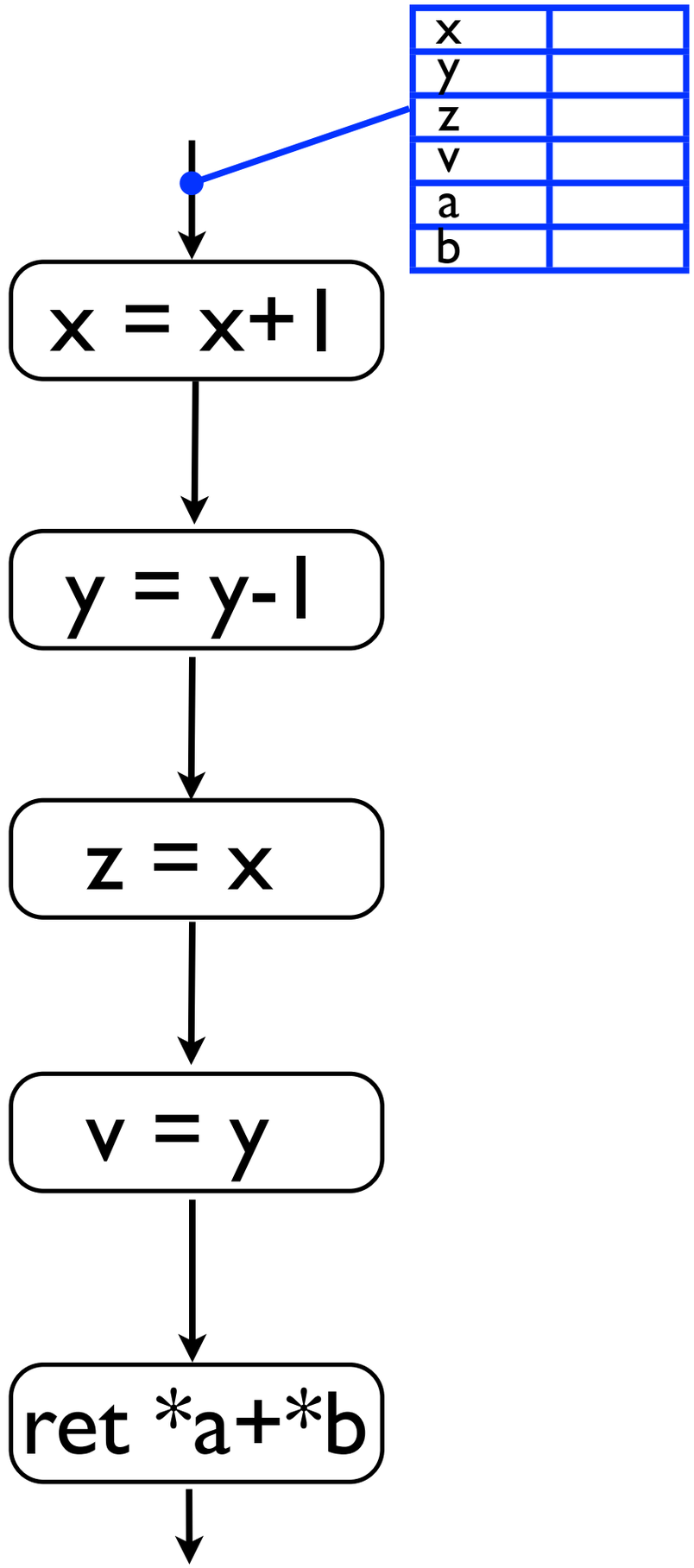
"Sparsifying" the Analysis

"Right Part at Right Moment"



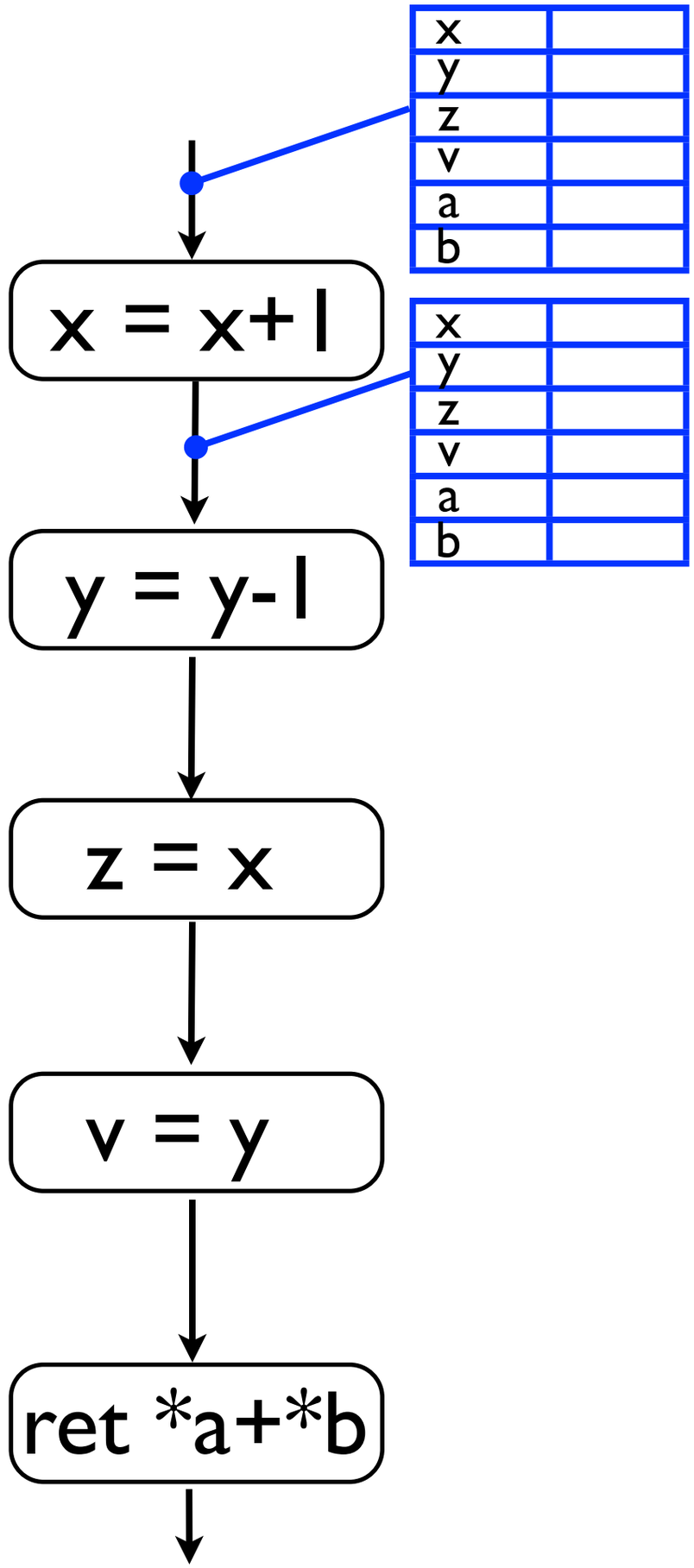
"Sparsifying" the Analysis

"Right Part at Right Moment"



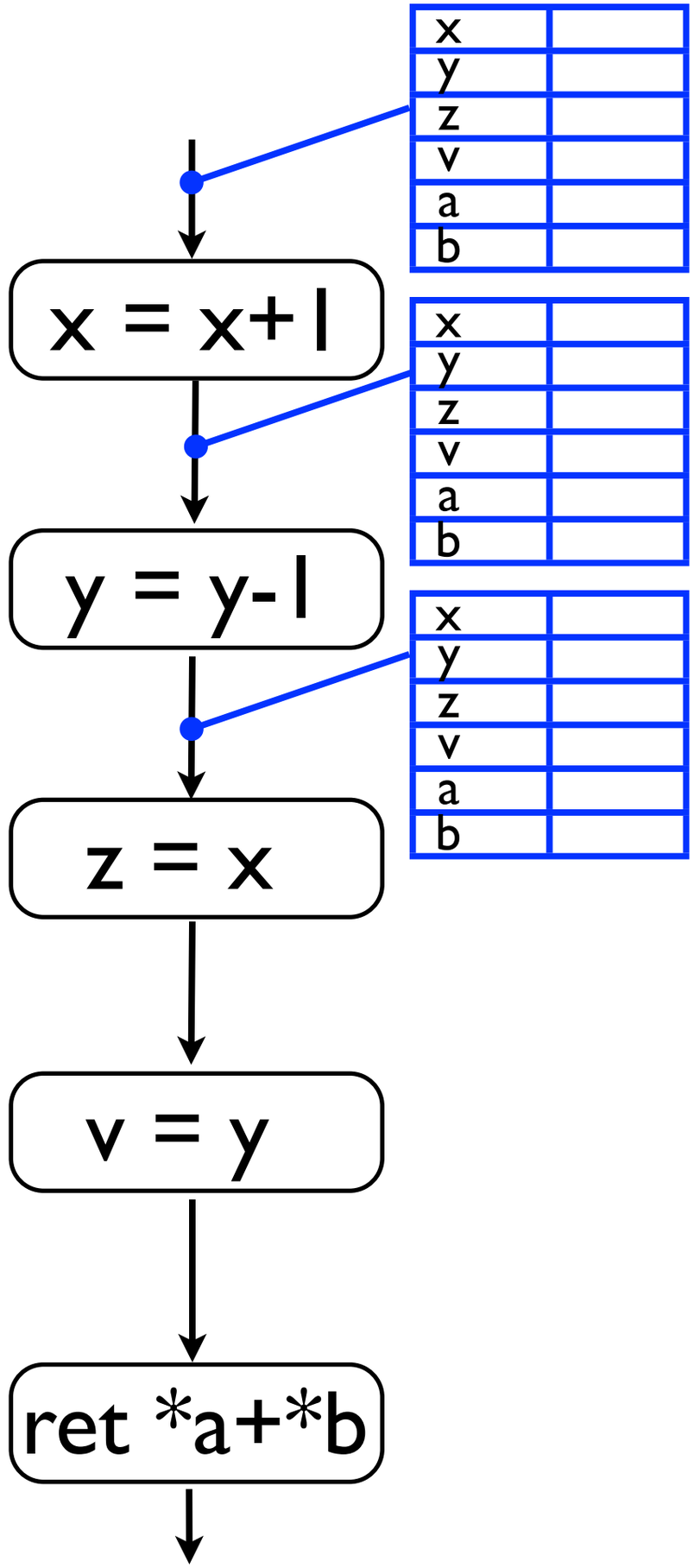
"Sparsifying" the Analysis

"Right Part at Right Moment"



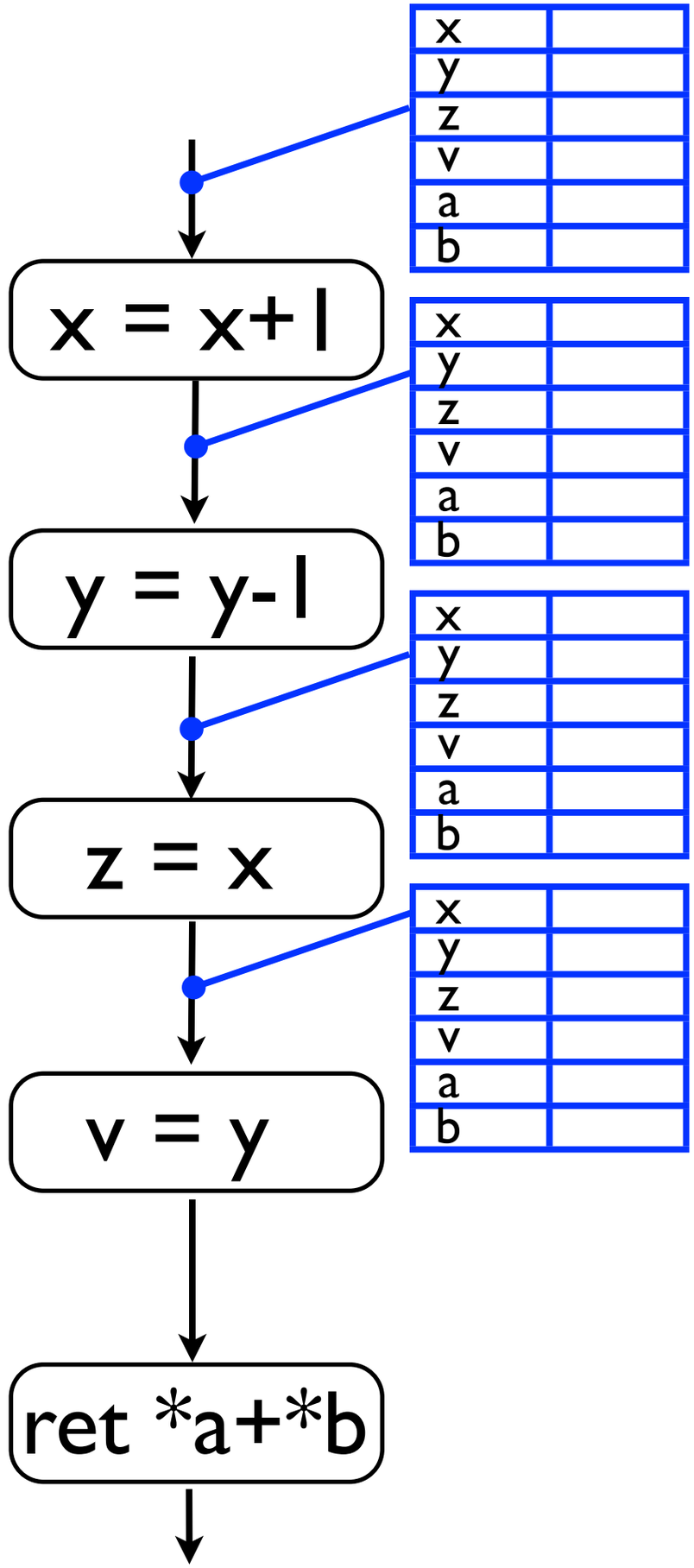
"Sparsifying" the Analysis

"Right Part at Right Moment"



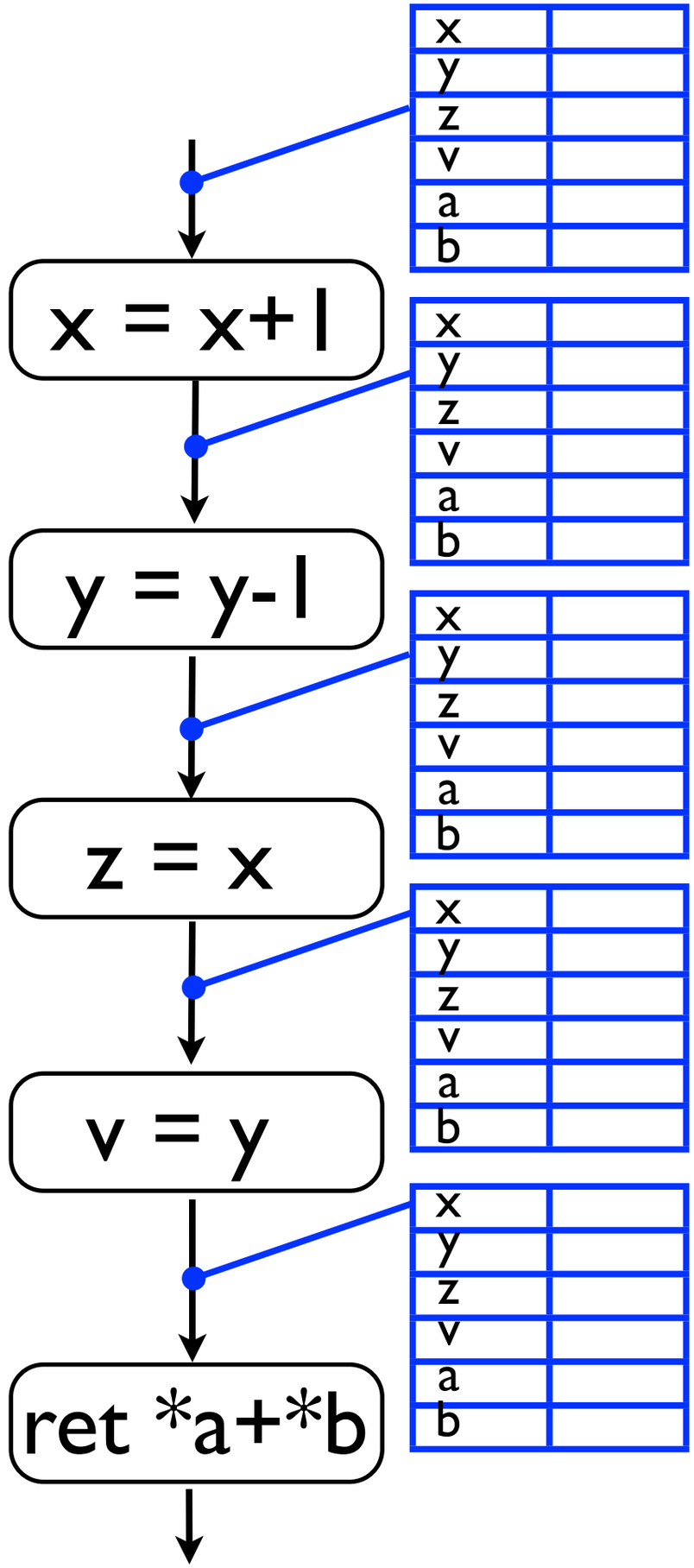
"Sparsifying" the Analysis

"Right Part at Right Moment"



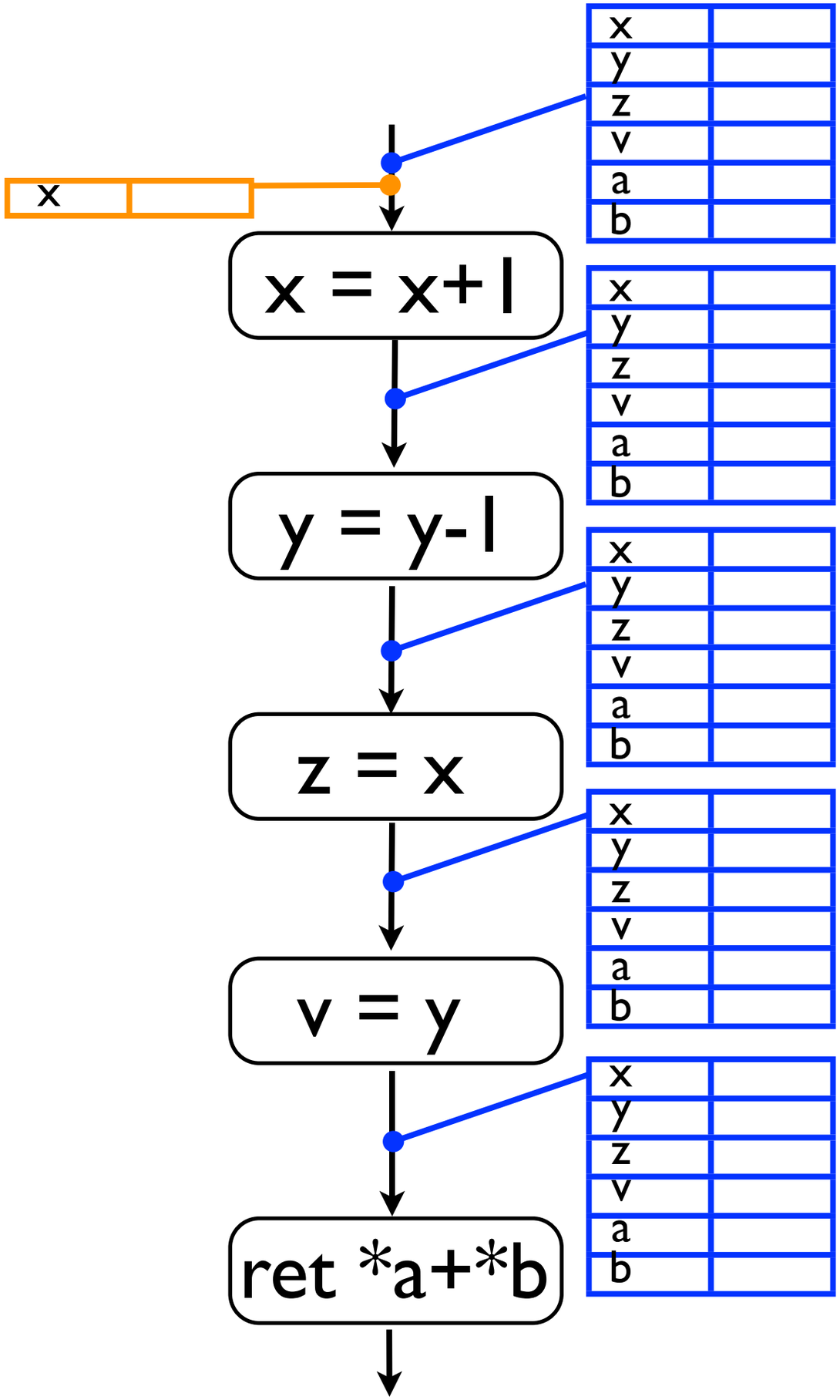
"Sparsifying" the Analysis

"Right Part at Right Moment"



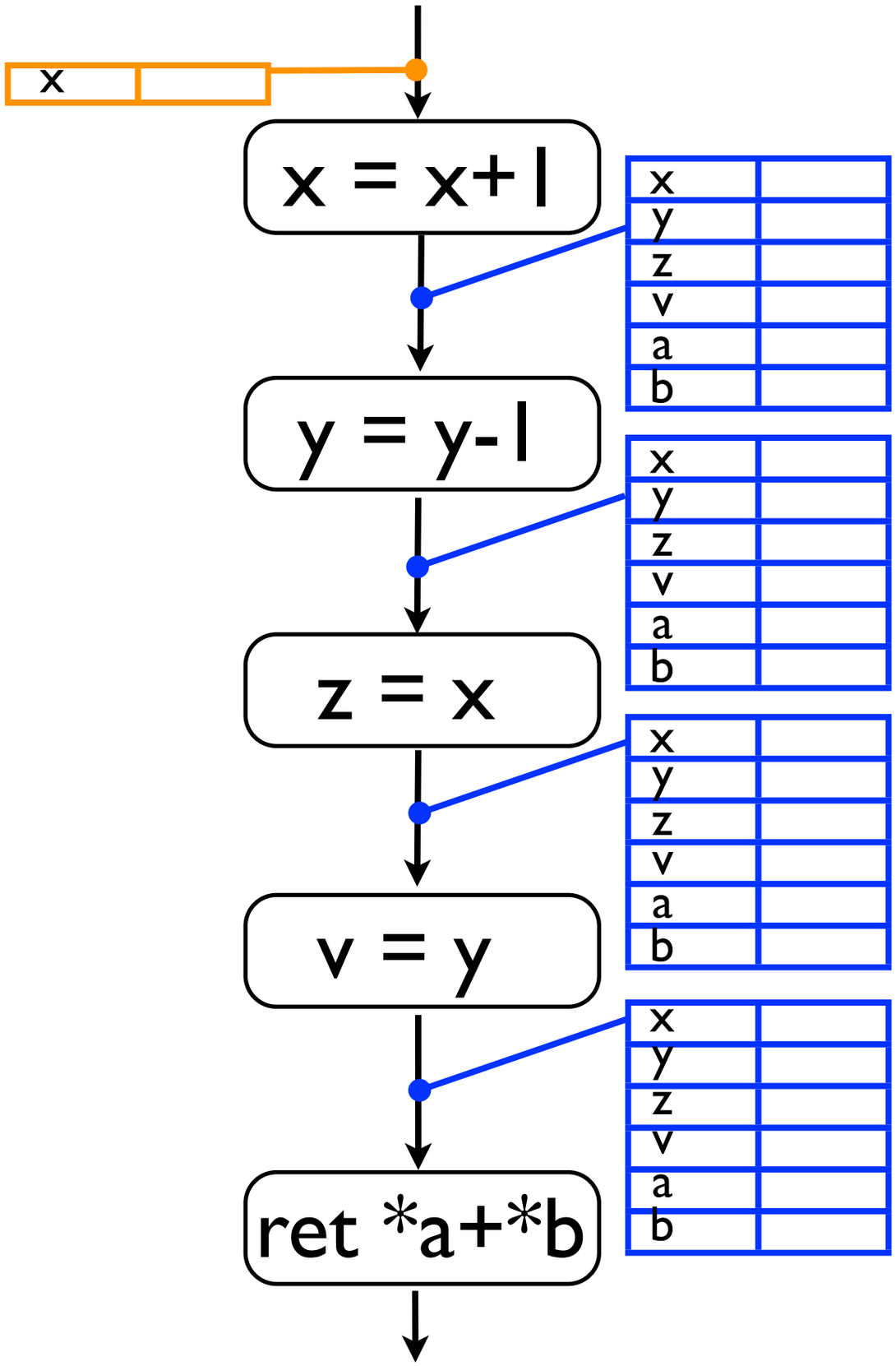
"Sparsifying" the Analysis

"Right Part at Right Moment"



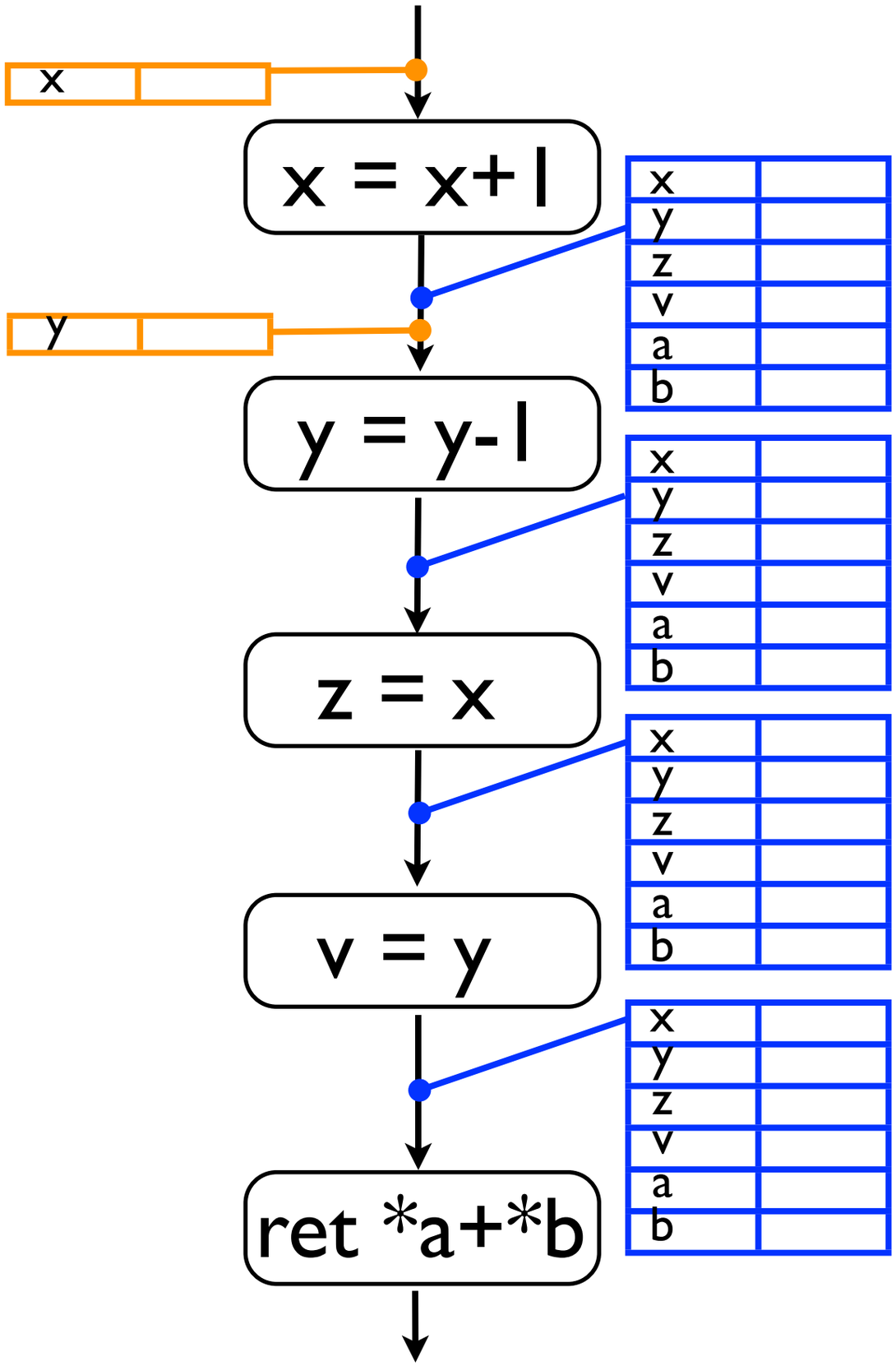
"Sparsifying" the Analysis

"Right Part at Right Moment"



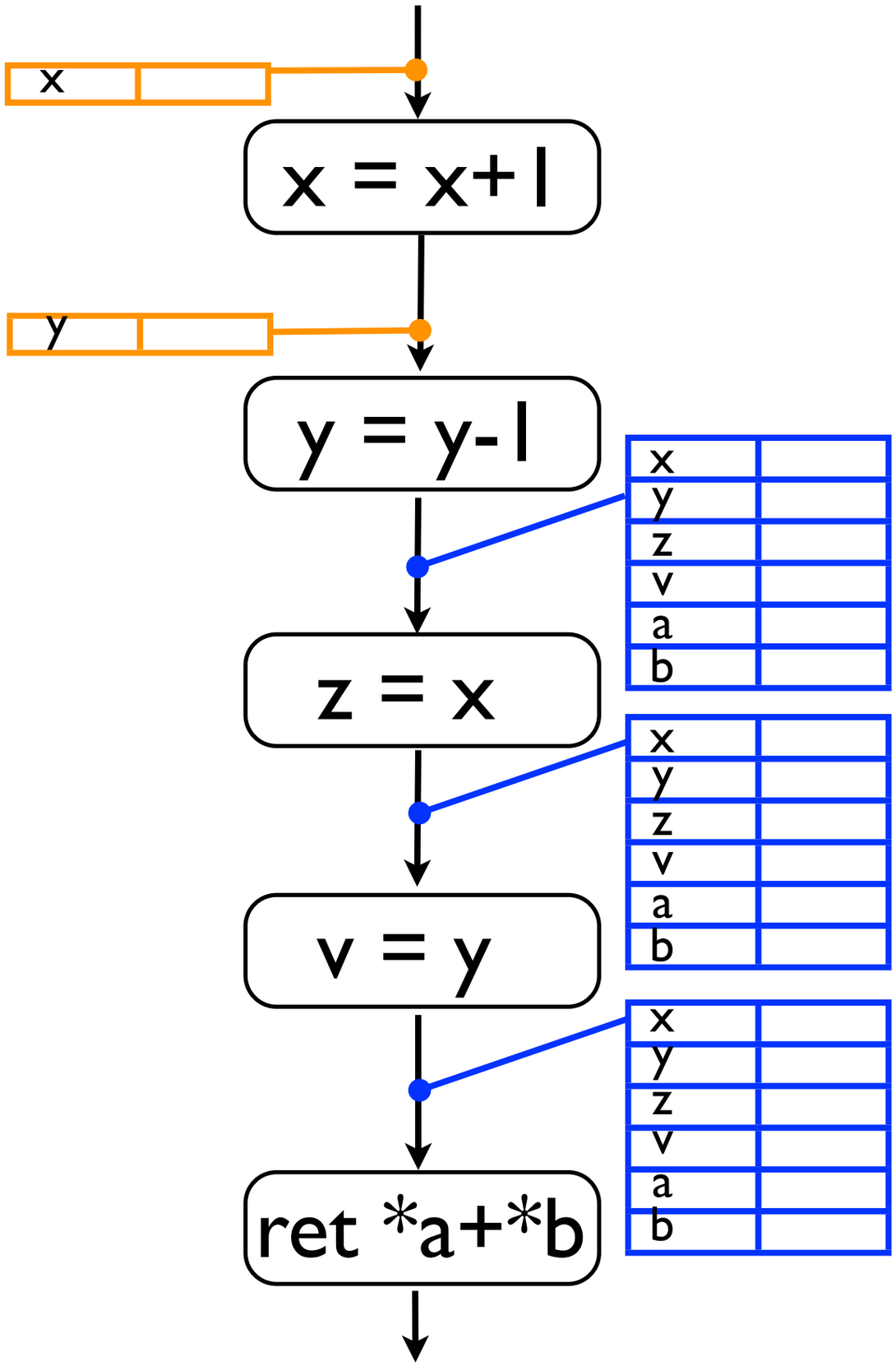
"Sparsifying" the Analysis

"Right Part at Right Moment"



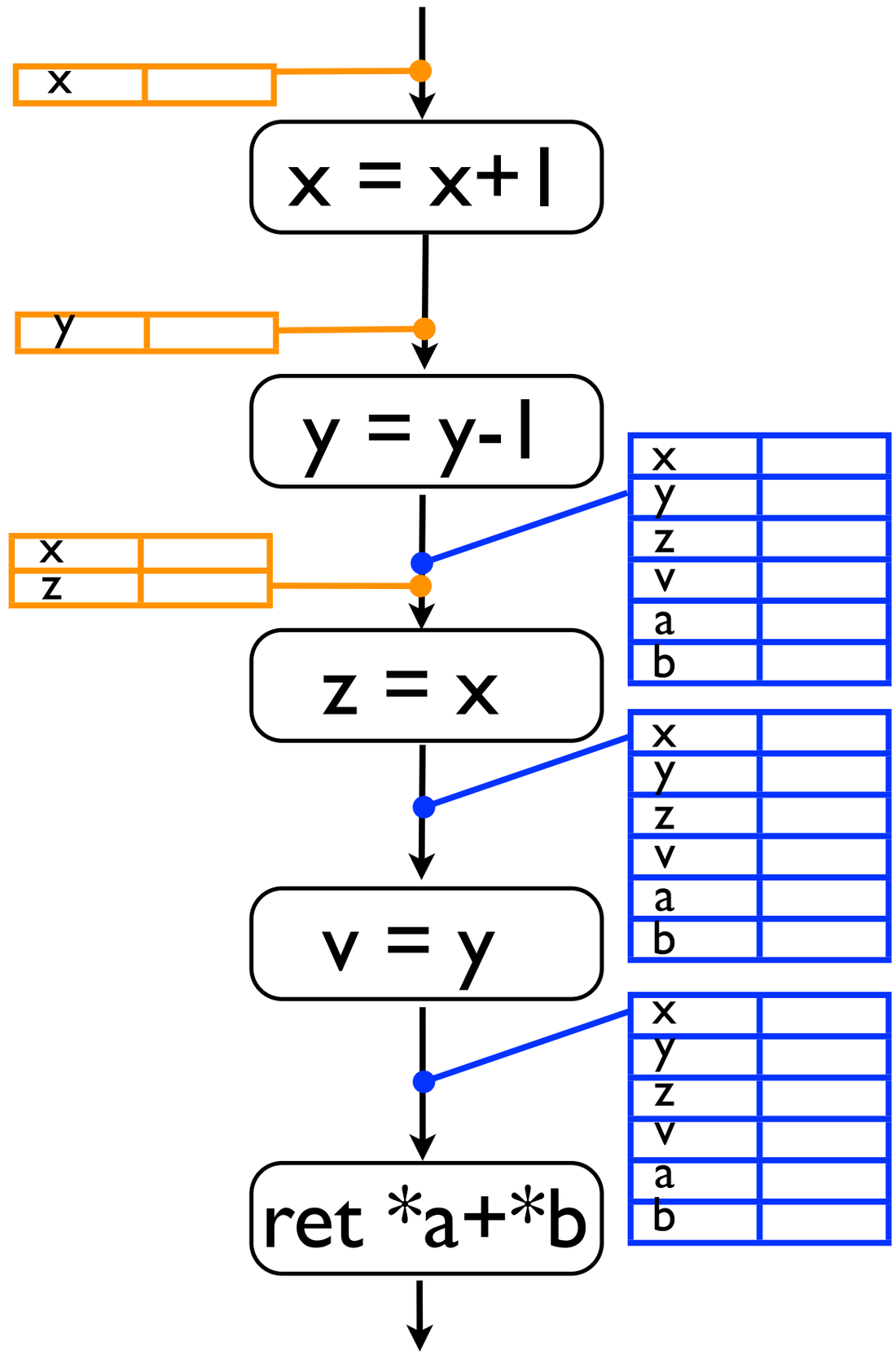
"Sparsifying" the Analysis

"Right Part at Right Moment"



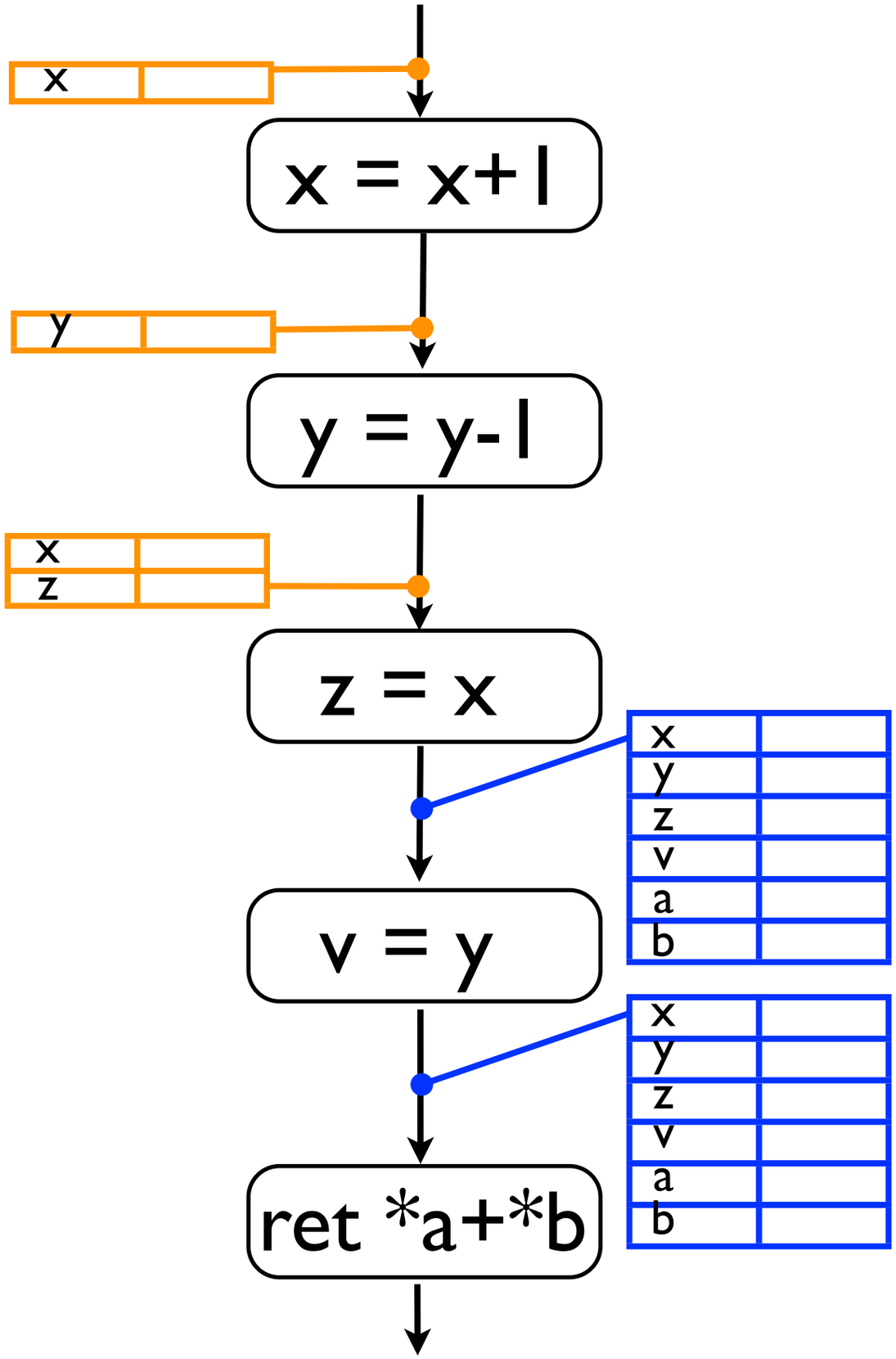
"Sparsifying" the Analysis

"Right Part at Right Moment"



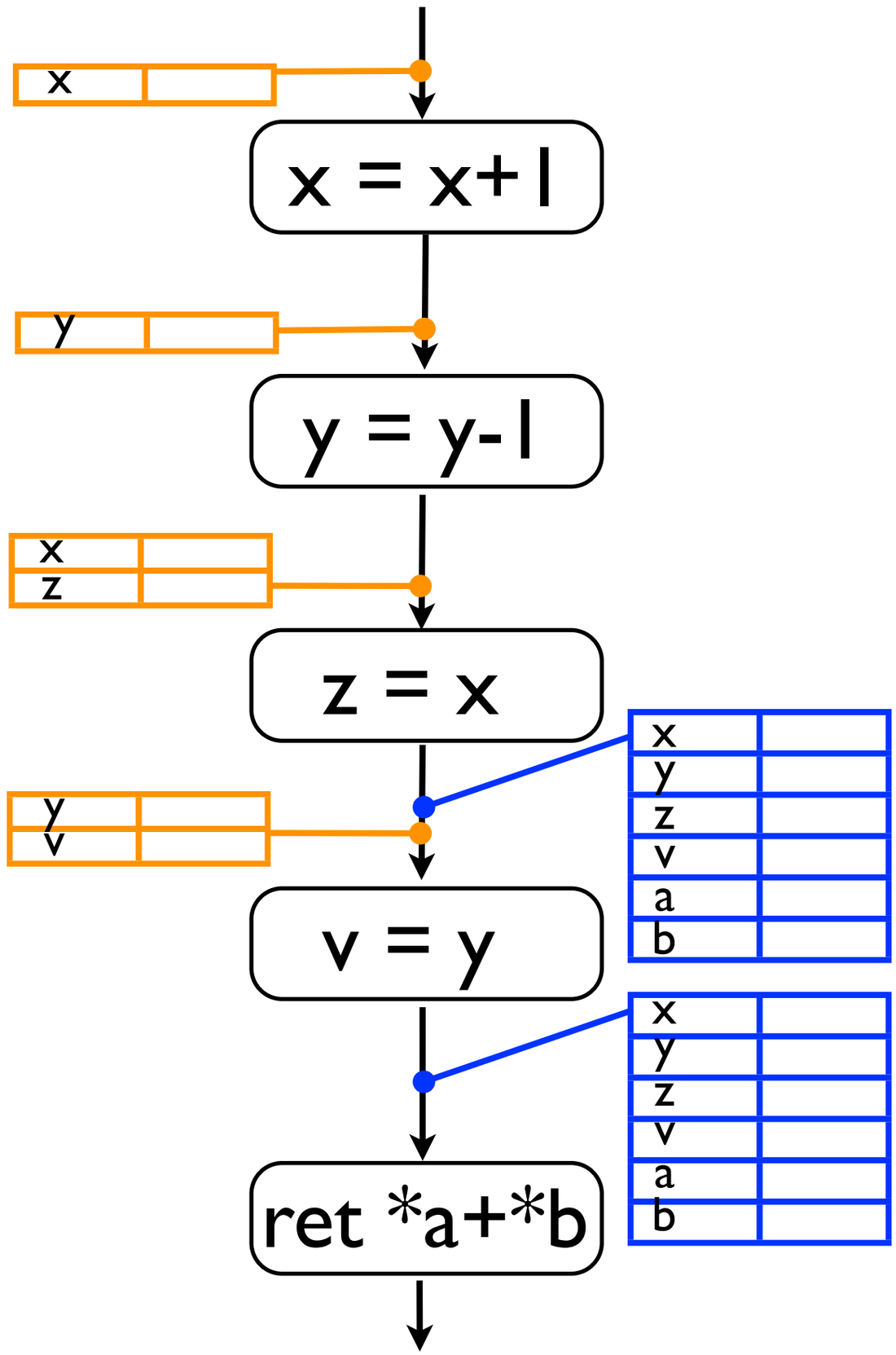
"Sparsifying" the Analysis

"Right Part at Right Moment"



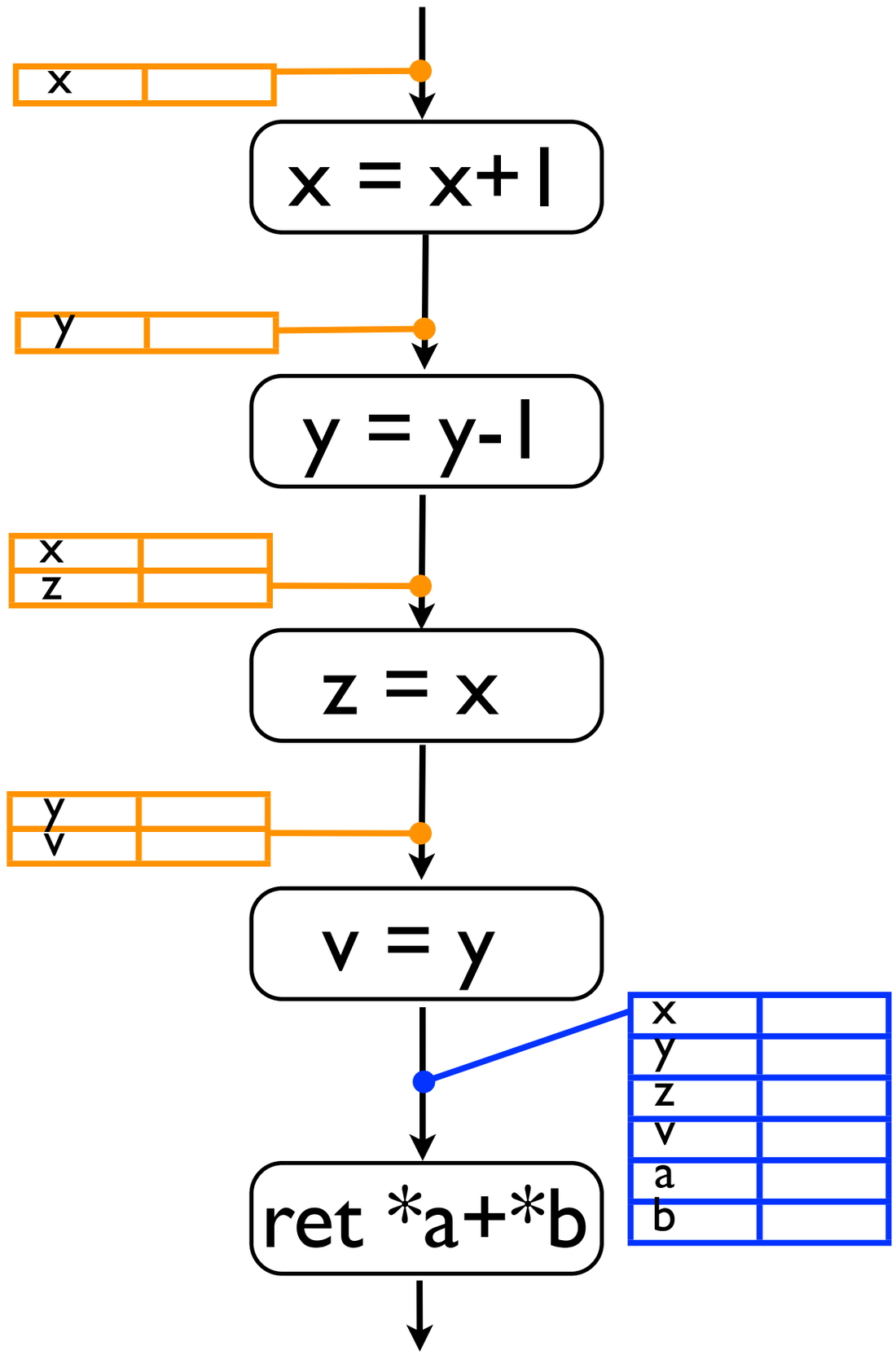
"Sparsifying" the Analysis

"Right Part at Right Moment"



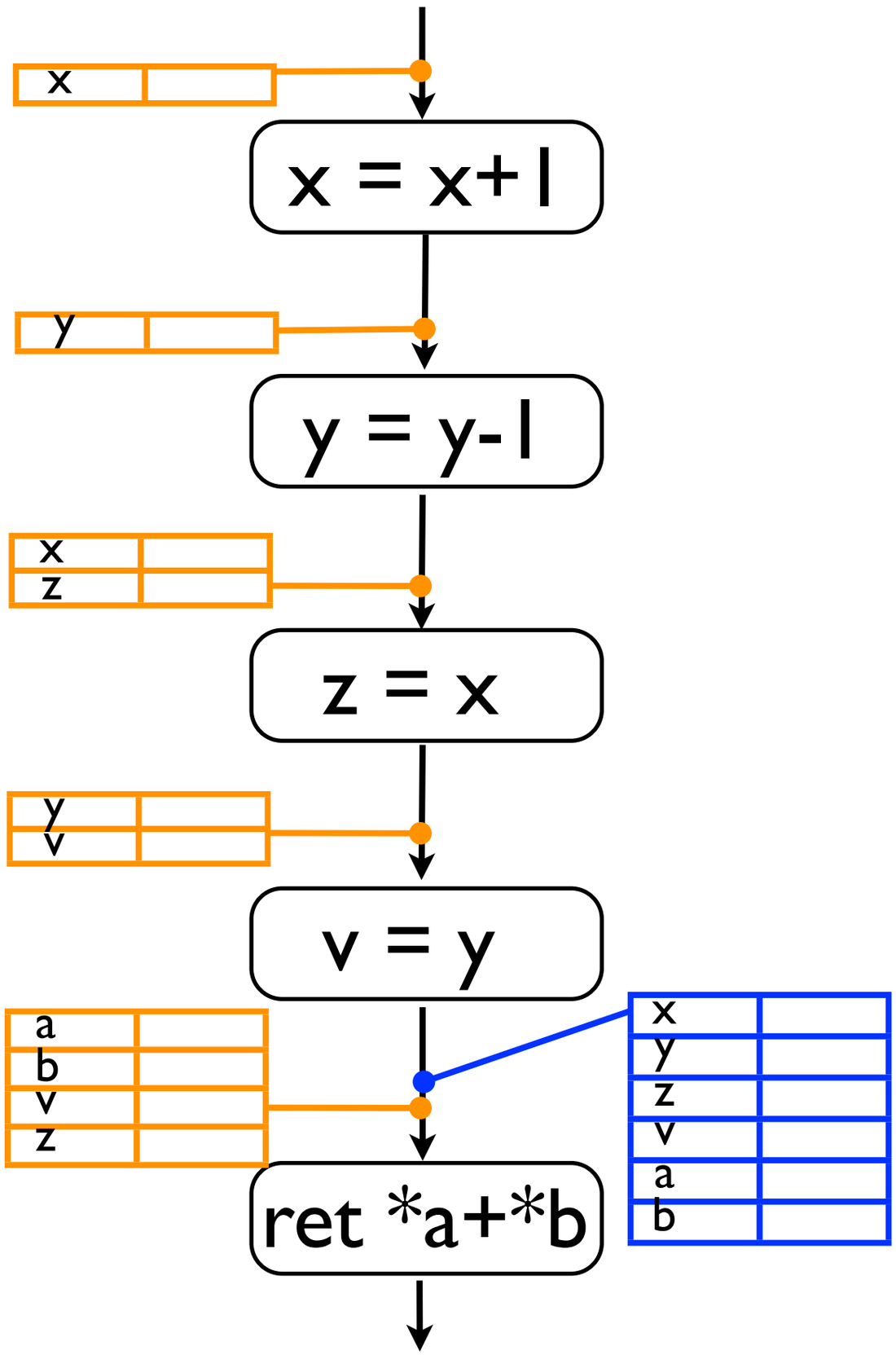
"Sparsifying" the Analysis

"Right Part at Right Moment"



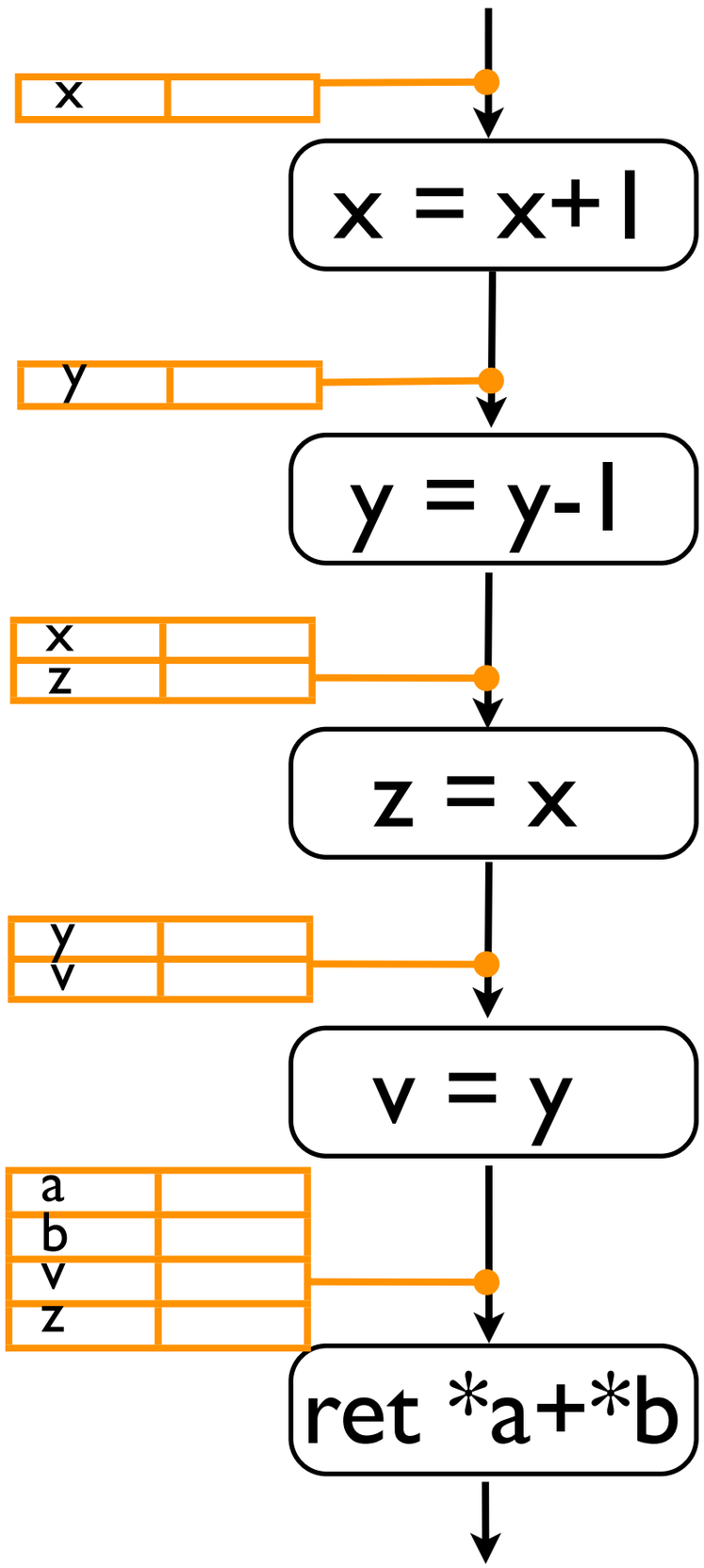
"Sparsifying" the Analysis

"Right Part at Right Moment"



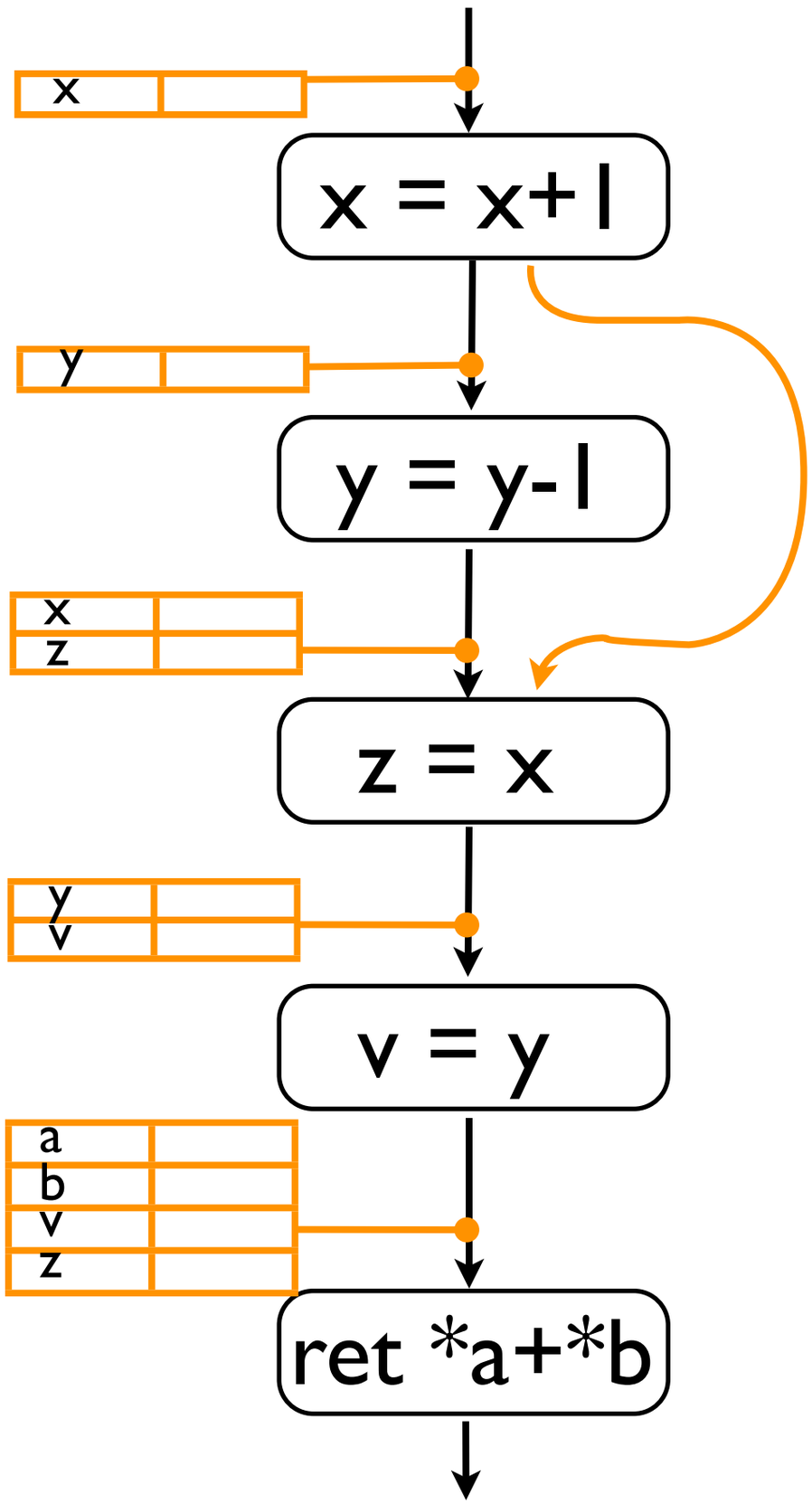
"Sparsifying" the Analysis

"Right Part at Right Moment"



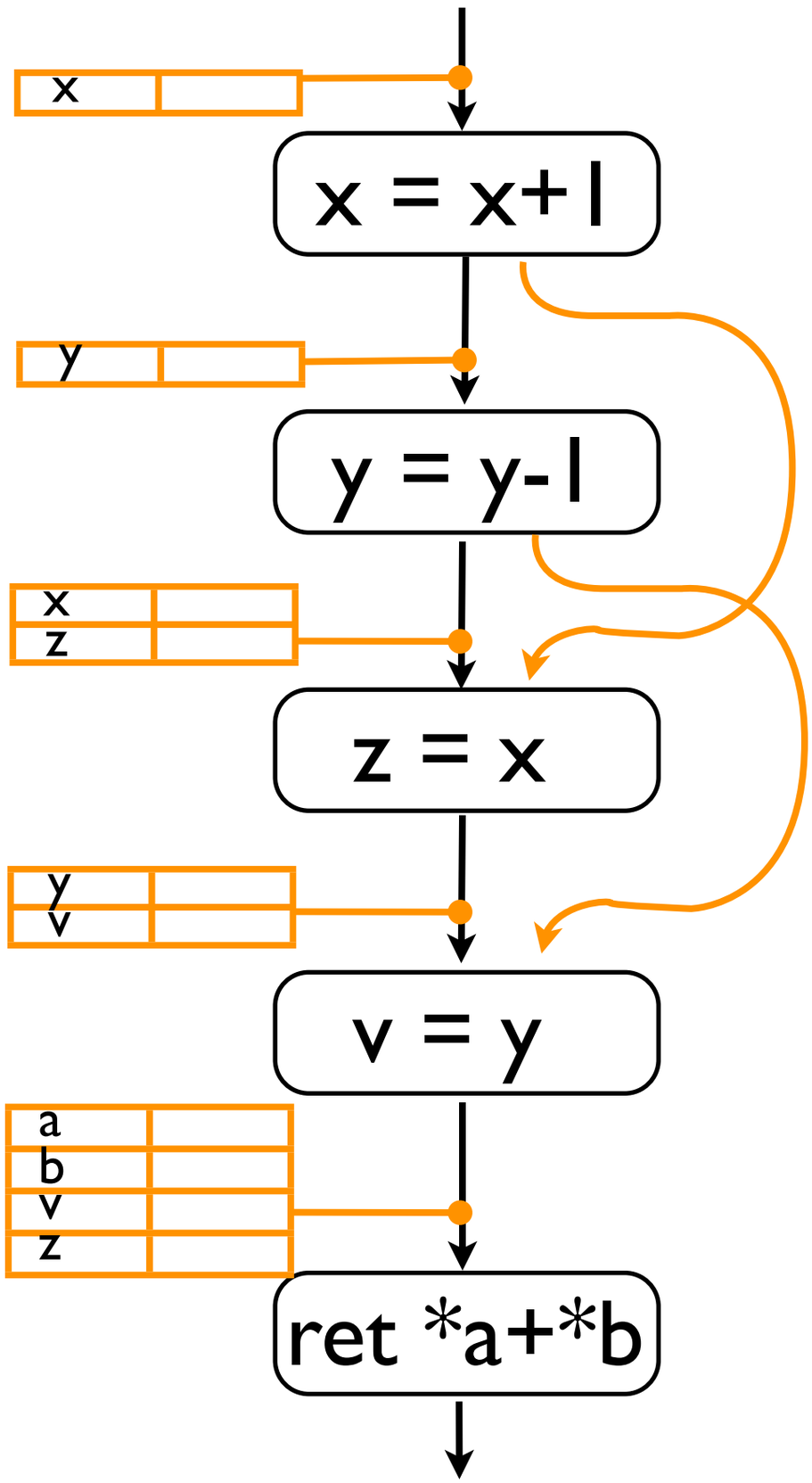
"Sparsifying" the Analysis

"Right Part at Right Moment"



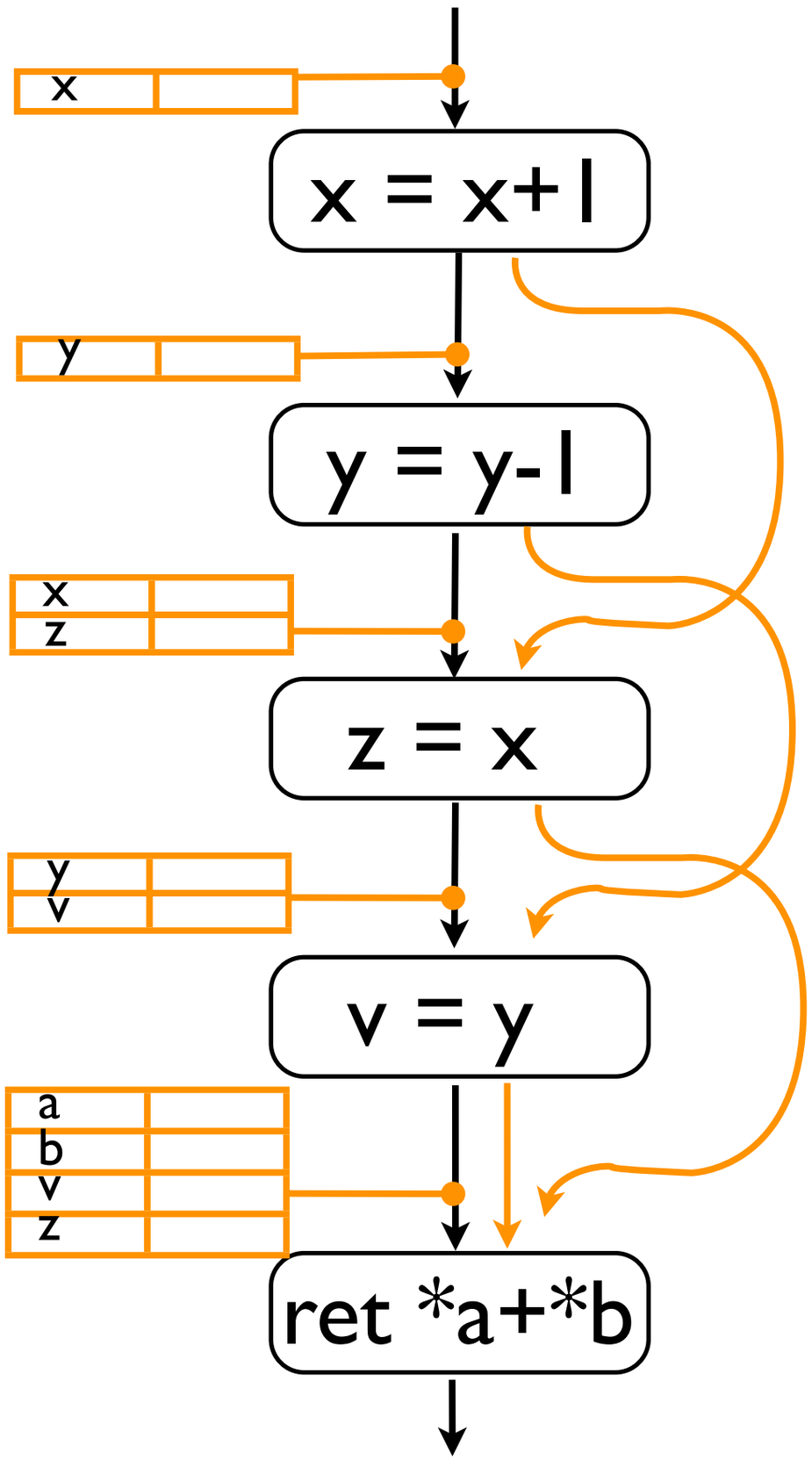
"Sparsifying" the Analysis

"Right Part at Right Moment"



"Sparsifying" the Analysis

"Right Part at Right Moment"

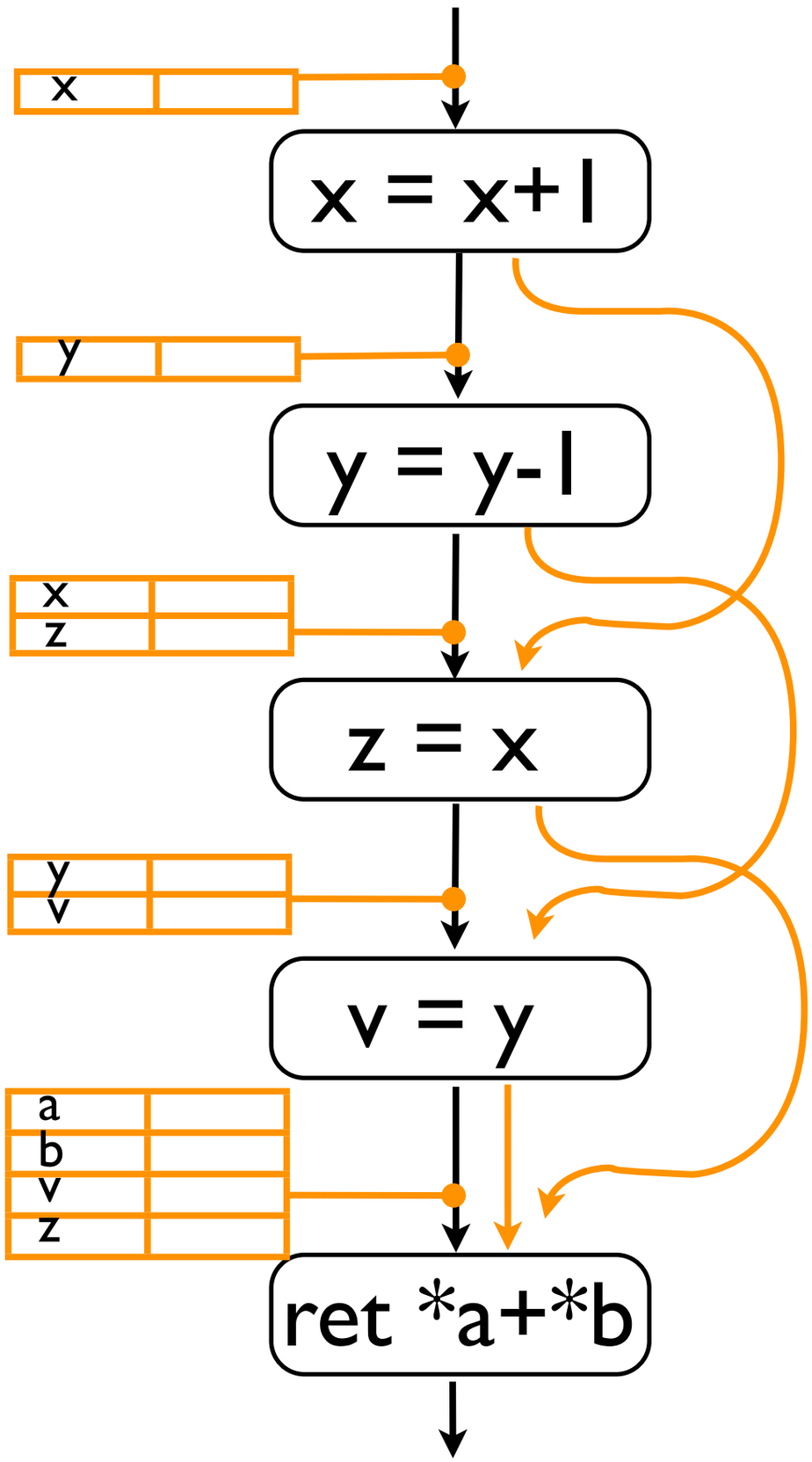


"Sparsifying" the Analysis

"Right Part at Right Moment"

$$\hat{F}(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \hookrightarrow c} \hat{X}(c')).$$

replace syntactic dependency by semantic dependency (data dependency)



Towards Sparse Version

Analyzer computes the fixpoint of $\hat{F}_{\_} \in (\mathbb{C} \to \hat{\mathbb{S}}) \to (\mathbb{C} \to \hat{\mathbb{S}})$

- baseline non-sparse one

$$\hat{F}(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \hookrightarrow c} \hat{X}(c')).$$

- unrealizable sparse version

$$\hat{F}_s(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \overset{l}{\rightsquigarrow} c} \hat{X}(c')|_l).$$

- realizable sparse version

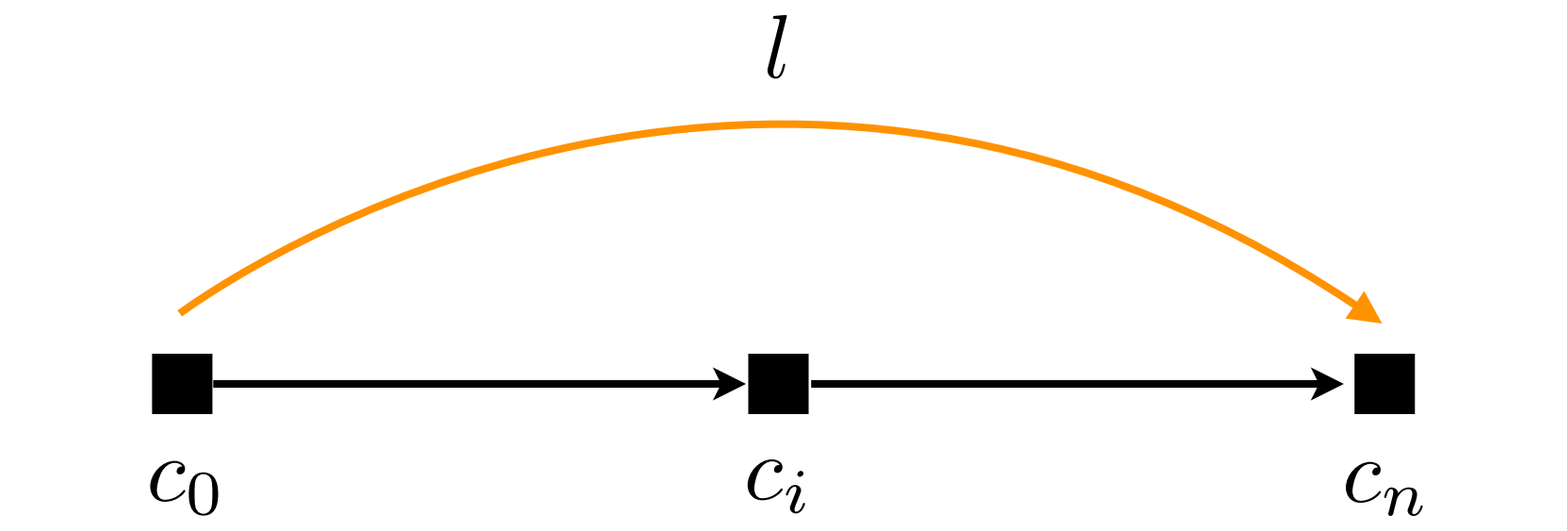
$$\hat{F}_a(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \overset{l}{\rightsquigarrow}_a c} \hat{X}(c')|_l).$$

Unrealizable Sparse One

$$\hat{F}_s(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \overset{l}{\rightsquigarrow} c} \hat{X}(c')|_l).$$

Data Dependency

$$c_0 \overset{l}{\rightsquigarrow} c_n \triangleq \exists c_0 \ldots c_n \in \mathsf{Paths}, l \in \hat{\mathbb{L}}.$$
$$l \in \mathsf{D}(c_0) \cap \mathsf{U}(c_n) \wedge \forall i \in (0,n).l \notin \mathsf{D}(c_i)$$



$l \in \mathsf{D}(c_0)$ $l \notin \mathsf{D}(c_i)$ $l \in \mathsf{U}(c_n)$

Unrealizable Sparse One

$$\hat{F}_s(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \overset{l}{\rightsquigarrow} c} \hat{X}(c')|_l).$$

Data Dependency

$$c_0 \overset{l}{\rightsquigarrow} c_n \triangleq \exists c_0 \ldots c_n \in \mathsf{Paths}, l \in \hat{\mathbb{L}}. \\ l \in \mathsf{D}(c_0) \cap \mathsf{U}(c_n) \wedge \forall i \in (0,n).l \notin \mathsf{D}(c_i)$$

Def-Use Sets

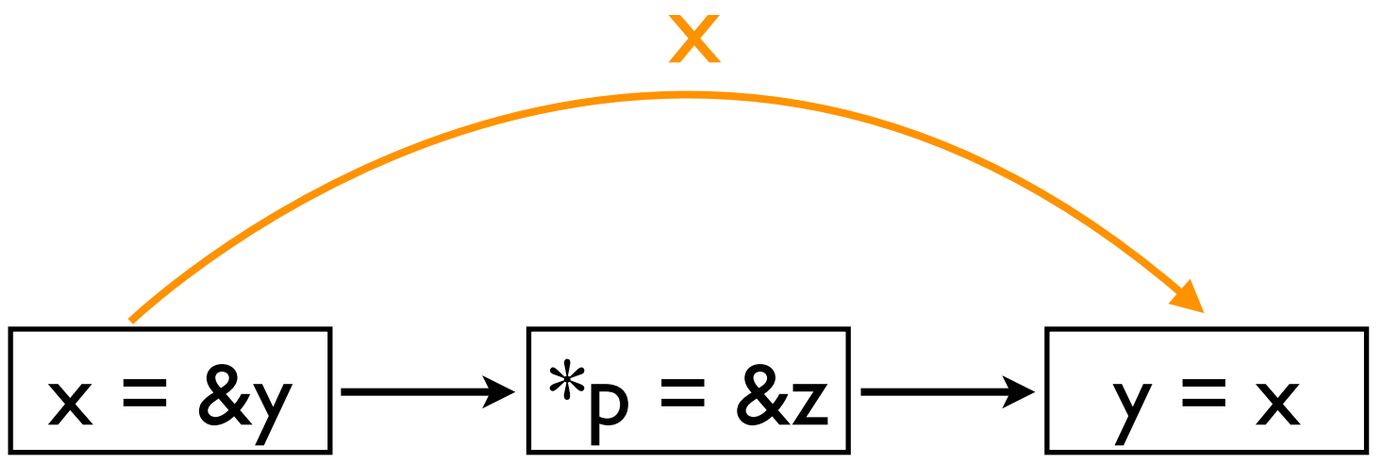
$$\mathsf{D}(c) \triangleq \{l \in \hat{\mathbb{L}} \mid \exists \hat{s} \sqsubseteq \bigsqcup_{c' \hookrightarrow c} (\mathit{fix}\hat{F})(c').\hat{f}_c(\hat{s})(l) \neq \hat{s}(l)\}.$$

$$\mathsf{U}(c) \triangleq \{l \in \hat{\mathbb{L}} \mid \exists \hat{s} \sqsubseteq \bigsqcup_{c' \hookrightarrow c} (\mathit{fix}\hat{F})(c').\hat{f}_c(\hat{s})|_{\mathsf{D}(c)} \neq \hat{f}_c(\hat{s}\backslash_l)|_{\mathsf{D}(c)}\}.$$

Preserving

$$\mathit{fix}\hat{F} = \mathit{fix}\hat{F}_s \quad \text{modulo D}$$

Data Dependency Example



Def	{x}	{a, b}	{y}
Use	ϕ	{p, a, b}	{x}

Realizable Sparse One

$$\hat{F}_a(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \overset{l}{\rightsquigarrow}_a c} \hat{X}(c')|_l).$$

Realizable Data Dependency

$$c_0 \overset{l}{\rightsquigarrow}_a c_n \triangleq \exists c_0 \ldots c_n \in \mathsf{Paths}, l \in \hat{\mathbb{L}}.$$
$$l \in \mathsf{D}(c_0) \cap \mathsf{U}(c_n) \wedge \forall i \in (0,n).l \notin \mathsf{D}(c_i)$$

Realizable Sparse One

$$\hat{F}_a(\hat{X}) = \lambda c \in \mathbb{C}.\hat{f}_c(\bigsqcup_{c' \overset{l}{\rightsquigarrow}_a c} \hat{X}(c')|_l).$$

Realizable Data Dependency

$$c_0 \overset{l}{\rightsquigarrow}_a c_n \triangleq \exists c_0 \ldots c_n \in \mathsf{Paths}, l \in \hat{\mathbb{L}}. \; l \in \mathsf{D}(c_0) \cap \mathsf{U}(c_n) \wedge \forall i \in (0,n).l \notin \mathsf{D}(c_i)$$

Preserving

$$fix\hat{F} \overset{\text{still}}{=} fix\hat{F}_a \quad \text{modulo } \hat{\mathsf{D}}$$

If the following two conditions hold

Conditions of $\hat{\mathsf{D}}$ & $\hat{\mathsf{U}}$

- over-approximation

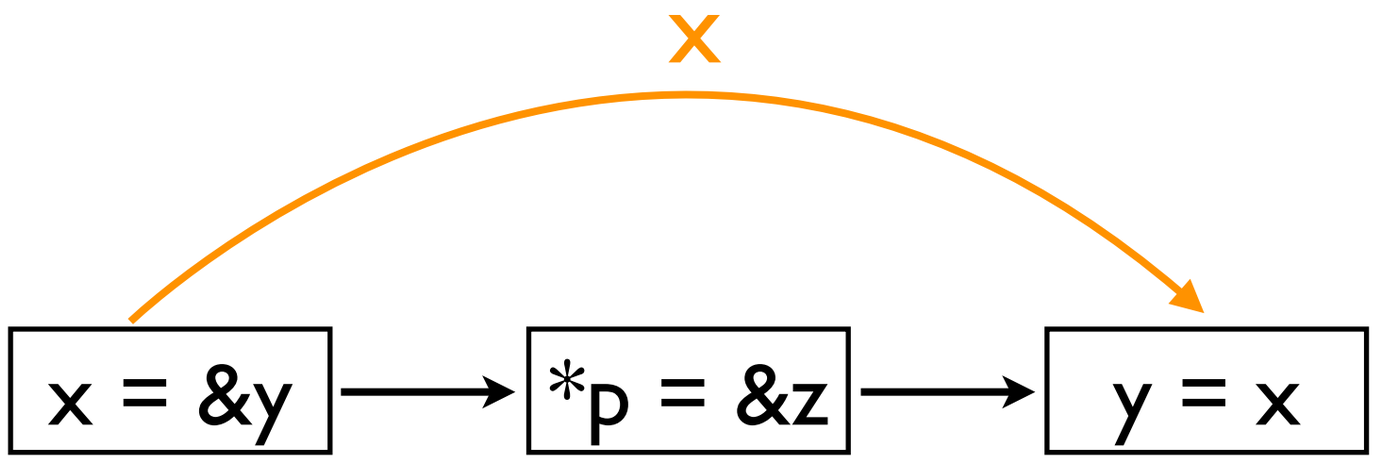
$$\mathsf{D}(c) \supseteq \mathsf{D}(c) \wedge \mathsf{U}(c) \supseteq \mathsf{U}(c)$$

- spurious definitions should be also included in uses

$$\underline{\hat{\mathsf{D}}(c) - \mathsf{D}(c)} \subseteq \hat{\mathsf{U}}(c)$$

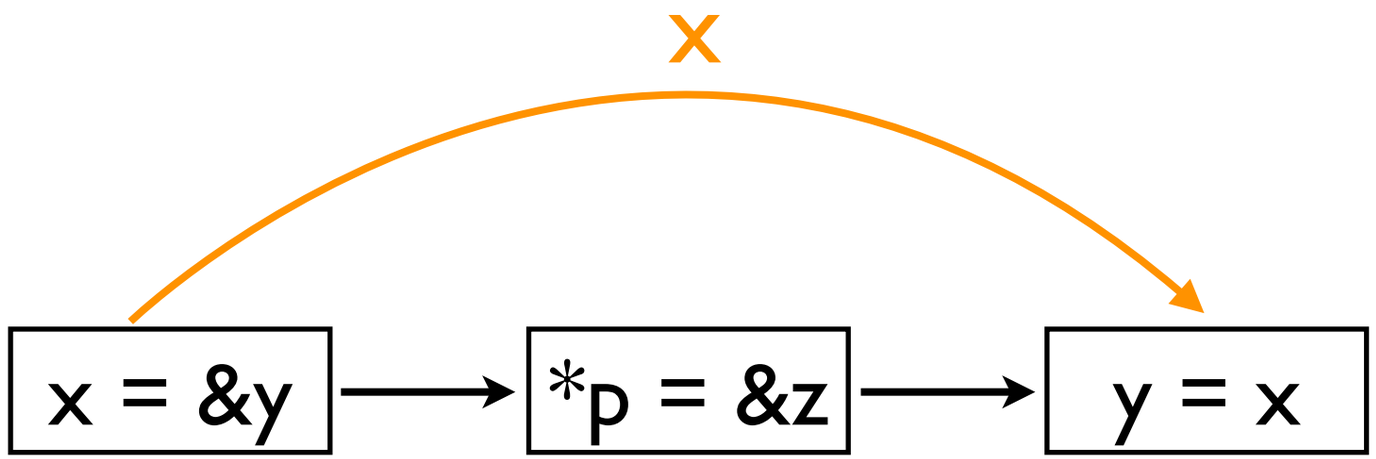
spurious definitions

Why the Conditions of $\hat{D}$ & $\hat{U}$



Def	{x}	{a, b}	{y}
Use	ϕ	{p, a, b}	{x}

Why the Conditions of $\hat{D}$ & $\hat{U}$

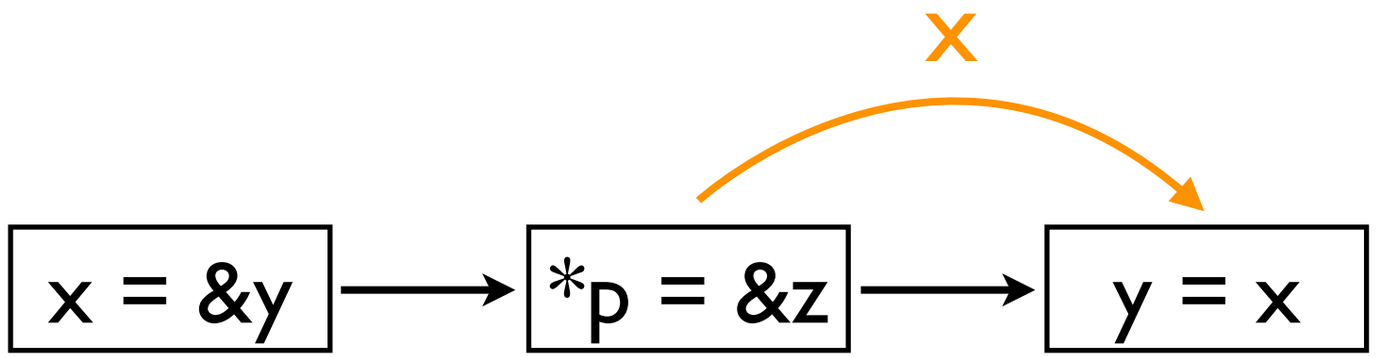


Approx. Def	{x}	{a, b, x}	{y}
Approx. Use	ϕ	{p, a, b}	{x}

$$\underline{\hat{D}(c) - D(c)} \not\subseteq \hat{U}(c)$$

{x}

Why the Conditions of $\hat{D}$ & $\hat{U}$

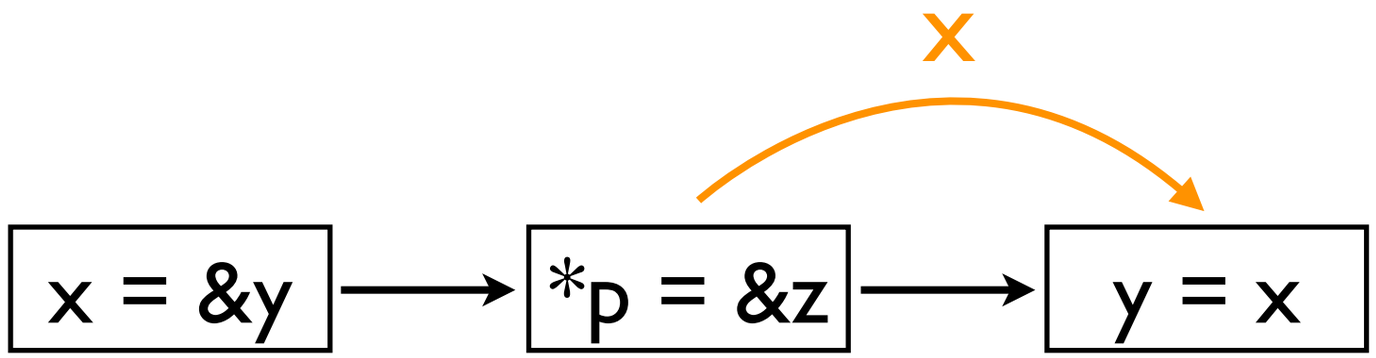


Approx. Def	{x}	{a, b, x}	{y}
Approx. Use	ϕ	{p, a, b}	{x}

$$\underline{\hat{D}(c) - D(c)} \not\subseteq \hat{U}(c)$$

{x}

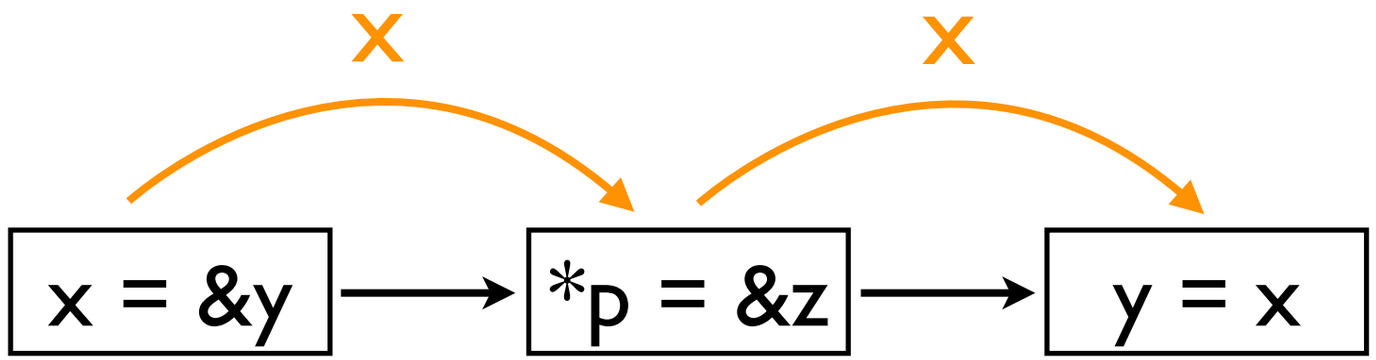
Why the Conditions of $\hat{D}$ & $\hat{U}$



Approx. Def	{x}	{a, b, x}	{y}
Approx. Use	ϕ	{p, a, b, x}	{x}

$$\frac{\hat{D}(c) - D(c) \subseteq \hat{U}(c)}{\{x\}}$$

Why the Conditions of $\hat{D}$ & $\hat{U}$



Approx. Def	{x}	{a, b, x}	{y}
Approx. Use	ϕ	{p, a, b, x}	{x}

$$\frac{\hat{D}(c) - D(c) \subseteq \hat{U}(c)}{\{x\}}$$

Hurdle: $\hat{D}$ & $\hat{U}$ Before Analysis?

- Yes, by yet another analysis with further abstraction
  - e.g., flow-insensitive abstraction

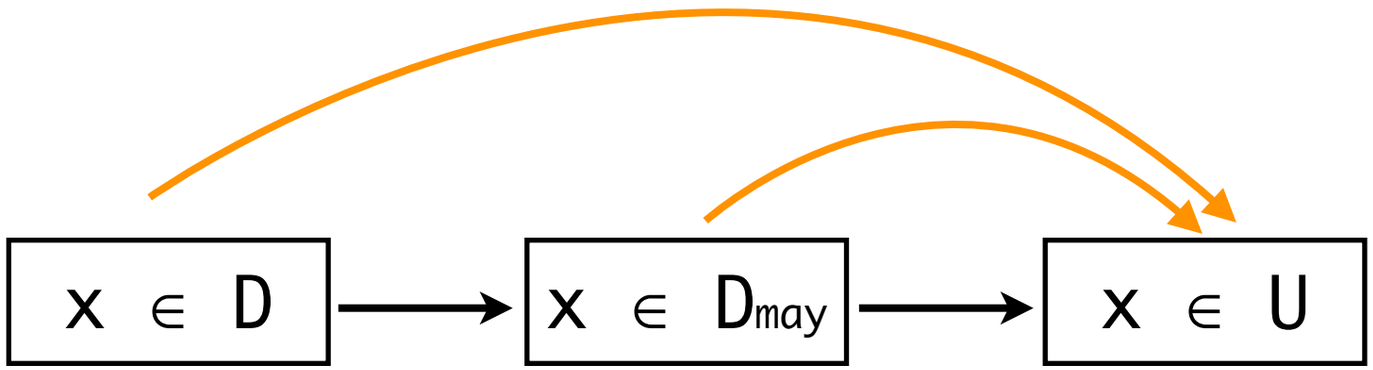
$$\mathbb{C} \to \hat{\mathbb{S}} \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \hat{\mathbb{S}} \qquad \hat{F}_p = \lambda \hat{s}.(\bigsqcup_{c \in \mathbb{C}} \hat{f}_c(\hat{s}))$$

- In implementation, $\hat{U}$ includes $\hat{D}$

$$\hat{\mathsf{D}}(c) - \mathsf{D}(c) \subseteq \hat{\mathsf{U}}(c)$$

Existing Sparse Techniques
(developed mostly in dfa community)

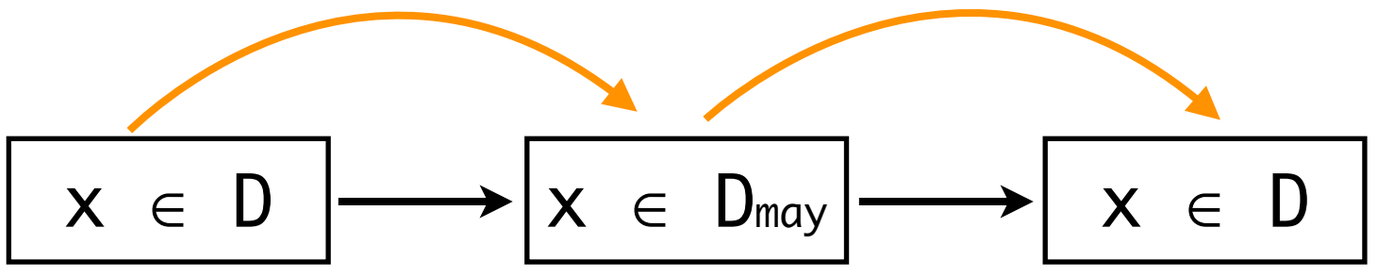
- Different notion of data dependency



def-use chains fail to preserve original precision

Existing Sparse Techniques
(developed mostly in dfa community)

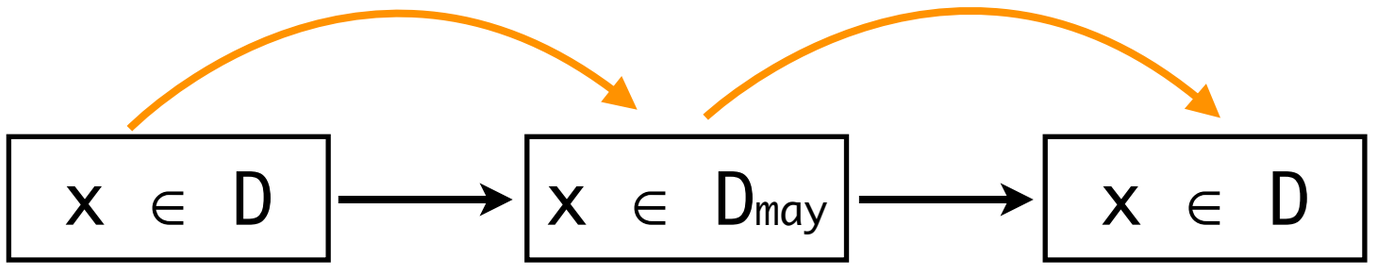
- Different notion of data dependency



our data dependency preserves original precision

Existing Sparse Techniques
(developed mostly in dfa community)

- Different notion of data dependency



- Existing sparse analyses are not general
  - tightly coupled with particular analysis, or
  - limited to a particular target language

Performance

Experiments

- On top of Sparrow The Early Bird
  - Sparse non-relational analysis with interval domain

$$\hat{\mathbb{S}} = AbsLoc \rightarrow Interval$$

- Sparse relational analysis with octagon domain

$$\hat{\mathbb{S}} = Packs \rightarrow Octagon$$

Performance

Sparse Interval Analysis

Program	**LOC**	**Non-sparse**		**Sparse**		**Spd↑**	**Mem↓**
		Time	**Mem**	**Time**	**Mem**		
gzip-1.2.4a	7 K	772	240	3	63	257 x	74 %
bc-1.06	13 K	1,270	276	7	75	181 x	73 %
less-382	23 K	9,561	1,113	33	127	289 x	86 %
make-3.76.1	27 K	24,240	1,391	21	114	1,154 x	92 %
wget-1.9	35 K	44,092	2,546	11	85	4,008 x	97 %
a2ps-4.14	64 K	∞	N/A	40	353	N/A	N/A
sendmail-8.13.6	130 K	∞	N/A	744	678	N/A	N/A
nethack-3.3.0	211 K	∞	N/A	16,373	5,298	N/A	N/A
emacs-22.1	399 K	∞	N/A	37,830	7,795	N/A	N/A
python-2.5.1	435 K	∞	N/A	11,039	5,535	N/A	N/A
linux-3.0	710 K	∞	N/A	33,618	20,529	N/A	N/A
gimp-2.6	959 K	∞	N/A	3,874	3,602	N/A	N/A
ghostscript-9.00	1,363 K	∞	N/A	14,814	6,384	N/A	N/A

Performance

Sparse Interval Analysis

Program	LOC	Non-sparse		Sparse		Spd↑	Mem↓
		Time	Mem	Time	Mem		
gzip-1.2.4a	7 K	772	240	3	63	257 x	74 %
bc-1.06	13 K	1,270	276	7	75	181 x	73 %
less-382	23 K	9,561	1,113	33	127	289 x	86 %
make-3.76.1	27 K	24,240	1,391	21	114	1,154 x	92 %
wget-1.9	35 K	44,092	2,546	11	85	4,008 x	97 %
a2ps-4.14	64 K	∞	N/A	40	353	N/A	N/A
sendmail-8.13.6	130 K	∞	N/A	744	678	N/A	N/A
nethack-3.3.0	211 K	∞	N/A	16,373	5,298	N/A	N/A
emacs-22.1	399 K	∞	N/A	37,830	7,795	N/A	N/A
python-2.5.1	435 K	∞	N/A	11,039	5,535	N/A	N/A
linux-3.0	710 K	∞	N/A	33,618	20,529	N/A	N/A
gimp-2.6	959 K	∞	N/A	3,874	3,602	N/A	N/A
ghostscript-9.00	1,363 K	∞	N/A	14,814	6,384	N/A	N/A

Performance

Sparse Octagon Analysis

Program	**LOC**	**Non-sparse**		**Sparse**		**Spd↑**	**Mem↓**
		Time	**Mem**	**Time**	**Mem**		
gzip-1.2.4a	7 K	2,078	2,832	21	269	98 x	91 %
bc-1.06	13 K	9,536	6,987	55	358	173 x	95 %
tar-1.13	20 K	∞	N/A	188	526	N/A	N/A
less-382	23 K	∞	N/A	432	458	N/A	N/A
make-3.76.1	27 K	∞	N/A	331	666	N/A	N/A
wget-1.9	35 K	∞	N/A	288	646	N/A	N/A
screen-4.0.2	45 K	∞	N/A	16,433	9,199	N/A	N/A
a2ps-4.14	64 K	∞	N/A	8,546	1,996	N/A	N/A
sendmail-8.13.6	130 K	∞	N/A	64,808	29,658	N/A	N/A

Summary

Our Sparse Framework

For precise, sound, and scalable static analysis

- Define a global safe abstract interpreter
- Make it sparse with our data dependencies
- Resulting sparse one scales with the same precision

Thank you